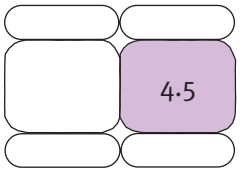
# E-BOOK: QUALCHE RIFLESSIONE SULLA (FUTURA?) EDITORIA DIGITALE

**Virginio B. Sala**

Il mondo dell'editoria, come altri settori, ha subito e fatto proprio il fenomeno della digitalizzazione. L'introduzione di dispositivi portatili dedicati alla lettura, software specifici per personal computer e pocket PC, hanno fatto pensare alla morte del libro di carta e nel contempo hanno portato alla coniazione del neologismo e-book. Che cos'è dunque l'e-book? Può essere definito ancora un libro oppure è qualcosa di diverso? L'articolo oltre a illustrare il nuovo panorama dell'editoria digitale offre anche qualche spunto di riflessione.




## 1. INTRODUZIONE

I mestieri dell'editoria sono cambiati molto, da quando è iniziata la diffusione del *personal computer* (PC): sono ben pochi ormai gli autori che scrivono i loro testi su una macchina per scrivere e, anche se con la matita o una penna rimane ancora molto comodo prendere qualche appunto e preparare schedine di lavoro, pochi sottovalutano la capacità organizzativa di un *database* o delle "scalette" redatte con la funzione "struttura" di un elaboratore di testi. Grazie allo schermo di un computer si svolgono poi sostanzialmente tutte le fasi successive, l'*editing* dei testi, l'eventuale realizzazione di disegni o grafici, l'elaborazione delle immagini (che a loro volta nascono in gran parte già digitali, o comunque vengono digitalizzate in una fase abbastanza precoce della loro esistenza), l'impaginazione. Il prodotto finale che esce da una redazione per avviarsi al ciclo industriale della stampa e della confezione è un oggetto digitale: ossia, uno o più *file*, nei possibili standard che permettono lo svolgimento delle fasi successive, a loro volta in buona misura ancora digitali, finché i *bit* lasciano il posto alla fisicità della carta stampata.

Queste trasformazioni hanno inciso profondamente sulla struttura delle case editrici, provocando la quasi totale scomparsa di talune figure professionali, la nascita di altre, l'ibridazione di altre ancora e, altrettanto significativamente, si è modificato il panorama di contorno: quasi estinti i compositori, si sono moltiplicati gli "studi" redazionali e i grafici-impaginatori indipendenti.

Per il lettore, l'utente finale del prodotto dell'impresa editoriale, che cos'è cambiato? Ben poco, verrebbe da dire: alla fine c'è ancora un oggetto fatto di fogli di carta, legati fra loro in qualche modo, su cui sono impressi segni fatti con l'inchiostro. È cambiato anche il modo in cui si producono le carte e quello con cui si fabbricano gli inchiostri, ma si hanno ancora libri, fascicoli, opuscoli che si leggono o si usano come trenta, cinquanta, cento o trecento anni fa. Certo, il libro dal punto di vista delle sue caratteristiche fisiche è il frutto raffinato di una evoluzione che ha superato il mezzo millennio e, grazie alla quale, ha acquisito do-

ti di comodità e flessibilità d'uso assai elevate, una "barriera" che qualunque concorrente deve superare per essere, se non proprio accettato, almeno preso in considerazione.
Non è proprio vero, però: esiste, infatti, una grande quantità di scritti che da tempo non raggiunge più (se non facoltativamente) il formato cartaceo, perché nasce e rimane digitale. Basta pensare agli "aiuti" in linea di molti prodotti *software*, che equivalgono a manuali per l'utente, e alla grande quantità di materiale presente in Internet. Certo, quella che si può fare con questi scritti non è esattamente la stessa "esperienza di lettura" che si vive con un libro, ma la *funzione* che quegli oggetti svolgono è del tutto analoga a quella che fino a ieri hanno svolto, e in molti campi svolgono ancora, i prodotti editoriali cartacei. Con alcune differenze di notevole importanza che consistono soprattutto nell'efficacia degli strumenti di ricerca, nell'estrema facilità di duplicazione, nella velocità e nell'economia di distribuzione, e con alcune possibilità, di portata altrettanto notevole, come l'ipertestualità e la sostanziale parità, nel mondo digitale, di testi, immagini, immagini in movimento, suoni e programmi.

## 2. LA RIVOLUZIONE TECNOLOGICA NELL'EDITORIA: L'E-BOOK

La prospettiva di estendere questi vantaggi a tutti i prodotti editoriali ha motivato numerosi tentativi da parte dei produttori di *hardware* e software, che hanno avuto un punto di massima visibilità, a partire dalla fine del 1999, con una serie ravvicinata di annunci. Dispositivi portatili dedicati alla lettura, software specifici per personal computer e *pocket PC*, hanno fatto gridare alla (ennesima) rivoluzione dell'editoria, con qualche inno alla definitiva morte del libro di carta e la coniazione del neologismo *e-book* (Figura 1). Il rumore mediatico attorno alla pretesa rivoluzione si è affievolito rapidamente con la rovinosa esplosione della bolla speculativa sui titoli tecnologici, che ha trascinato con sé tutto l'eccesso d'entusiasmo per la *new economy*. È stata una rivoluzione incompiuta (una delle tante), ma ha lasciato parecchi segni e ha gettato un seme per il quale si può pronosticare una crescita più lenta e ponderata ma, probabilmente molto più difficile da arrestare.
Che cos'è un e-book, in realtà? La domanda ammette più d'una risposta. Già la parola "libro" porta con sé una buona dose d'ambiguità: libro è l'oggetto fisico – di cui si può dire che pesa, per esempio, 200 grammi, che è fabbricato con carta "usomano" di una certa grammatura, rilegato in brossura e via dicendo; ma libro è, anche, in qualche modo la matrice dei singoli oggetti fisici – per cui si può dire che di un certo libro sono state stampate 2000 copie, o che ne sono state tirate tre edizioni; e libro, infine, è il "contenuto" di quegli oggetti, indipendente dalla singola realizzazione fisica – per cui si può raccontare a un amico che il *Don Chisciotte* o *Guerra e pace* è il libro più bello che si è letto.
La versione "elettronica" aggiunge qualche ulteriore complessità ai significati della parola "libro": si è parlato di e-book per indicare un dispositivo dedicato alla lettura, un programma per la lettura che possa essere eseguito su un dispositivo *general-purpose*, il file che archivia i contenuti da leggere o, ancora, il sistema specifico che presiede all'erogazione e alla distribuzione di quei file per gli strumenti di lettura e, infine, anche i contenuti digitali di ogni singola pubblicazione. Ce n'è di che ingenerare abbastanza confusione!
La prima cosa da mettere in chiaro è che l'elemento cruciale nelle soluzioni prospettate negli ultimi anni sotto il nome di e-book è una delle tante soluzioni proposte per risolvere il problema del controllo delle copie e

**FIGURA 1**
*Esempio di e-book per la lettura di fumetti*

del *copyright*. Per i materiali cartacei, infatti, nella maggior parte dei Paesi, un editore acquisisce dall'autore il diritto esclusivo di produrre e vendere copie cartacee della sua opera ed è tutelato in quella sua esclusività. La possibilità di rendere efficace l'esclusiva dipende dalla fisicità degli oggetti libro e dalla relativa difficoltà di produrre copie equivalenti di valore commerciale. Per qualche secolo l'istituto del *copyright* (che è, letteralmente, il "diritto di copia") ha funzionato abbastanza bene: per i libri la sua tutela è stata messa un po' in crisi dalle fotocopie, così come, nell'editoria musicale, per i dischi è stata messa in crisi dall'economicità dei registratori domestici. Nel mondo digitale, la possibilità di produzione di copie diventa addirittura quasi banale: copiare un file (perché alla fine si tratta di questo) è un'operazione elementare, e per di più quello che si ottiene come risultato è un file esattamente identico a quello di partenza. Una fotocopia è sempre meno bella della pagina di un libro originale e la fotocopia di una fotocopia è anche peggio. Nel passaggio da un disco a una cassetta qualcosa si perde e la qualità musicale peggiora in eventuali passaggi successivi da una cassetta a un'altra, ma la copia di un file digitale è indistinguibile dall'originale, quali che siano i "contenuti" del file – indipendentemente dal fatto che rappresenti un testo, un'immagine, video, audio o programma.

Ogni volta che le tecnologie di copia fanno un passo avanti, le regole del gioco inevitabilmente cambiano. Per un editore che già ha un percorso produttivo fondamentalmente digitale, l'idea di fermarsi a quello stadio e non dover accedere al percorso industriale della stampa è attraente: oltre a un risparmio di costi industriali, si eliminano in un colpo solo i problemi di magazzino e di distribuzione, con tutte le inefficienze relative. Sfruttando Internet per la distribuzione, basta conservare su un *server* in rete l'originale di un libro digitale, poi se ne genera una copia, sempre digitale, quando qualcuno la richiede e la si invia a destinazione tramite la rete stessa. Non ci sono più problemi di copie invendute ferme nel magazzino, né di resi dalle librerie. E non ci sono più neanche i problemi dei libri che escono da catalogo perché le possibilità stimate di vendita non giustificano una tiratura con i processi convenzionali di stampa e tutti i costi di magazzino e distribuzione. Una copia digitale di un libro, invece, può rimanere, anche, indefinitamente su un server con un costo marginale.

**DRM** (*Digital Rights Management*) è il nome generico dei sistemi di gestione dei diritti per dispositivi digitali, dei sistemi cioè che gestiscono le licenze e le chiavi per la fruizione di contenuti digitali. Un sistema DRM può essere specifico per gli e-book, ma può anche avere un campo d'azione più ampio e gestire prodotti digitali diversi, dall'audio al *video* – quindi anche brani musicali o *film*. La portata dei singoli sistemi DRM può, quindi, variare molto, ma i più ambiziosi (come quello della Microsoft) mirano alla gestione dei diritti per *qualunque* tipo di oggetto digitale. In termini molto generali, questi sistemi governano la concessione di licenze d'uso e delle chiavi crittografiche per la decifrazione e quindi l'uso dei contenuti. All'interno del file che rappresenta i contenuti digitali, per esempio un libro elettronico, si possono inserire dei *metadati*, indicazioni standardizzate che specificano gli "aventi diritto" (autore, agente, editore, per esempio) e come e a quali condizioni quel file possa essere ceduto. Un certo libro elettronico, per esempio, potrebbe essere ceduto a un determinato "prezzo di copertina" a utenti generici, con una particolare suddivisione degli utili fra autore ed editore, e a un prezzo diverso a una biblioteca, magari con una diversa ripartizione degli utili fra gli aventi diritto; potrebbe essere ceduto gratuitamente a particolari organizzazioni senza fini di lucro oppure potrebbe essere ceduto con diritti di stampa o meno; ecc.. Tutte queste indicazioni possono essere codificate come metadati; il sistema DRM deve leggerle, interpretarle e applicarle. Tra i suoi compiti ci sono anche quelli di provvedere alla liquidazione delle competenze agli aventi diritto: quando un libro elettronico viene venduto, accredita all'autore, all'editore, eventualmente all'agente le rispettive percentuali. Le responsabilità del sistema possono diventare molto gravose quando in un'opera sono presenti contributi di diversa natura: disegni, fotografie, citazioni da altre opere... Se il fotografo X ha consentito l'utilizzo di una sua fotografia Y solo a determinate condizioni (economiche e giuridiche), il sistema deve verificare che siano rispettate... e via di questo passo. Per svolgere le sue funzioni, un sistema DRM deve collegarsi ai sistemi dove sono conservati i contenuti, a quelli che svolgono le funzioni crittografiche, a quelli degli operatori finanziari (banche, gestori di carte di credito, per esempio), a quelli degli eventuali intermediari come biblioteche o librerie.

## 2.1.Come si crea un e-book

La creazione di un libro elettronico segue le vie convenzionali della realizzazione di un libro: un *word processor* per il testo, programmi di grafica per le illustrazioni e così via. Per arrivare a creare un e-book in stile Adobe, il prodotto finale deve essere convertito in formato PDF (*Portable Document Format*), cosa che gran parte degli editori già fa, come ultimo passo verso la stampa. Un libro in formato PDF è già un libro elettronico: può essere letto con l'*Acrobat e-book Reader* e tutte le funzioni del programma di lettura sono automaticamente attive. Gli manca però tutta la parte di gestione dei diritti, per la quale bisogna che il file "passi" attraverso un sistema **DRM** Adobe (attualmente denominato *Content*

*Server*), che provvede alla protezione attraverso cifratura e ne gestisce, quindi, la distribuzione con la cessione delle chiavi di decifrazione, gli eventuali pagamenti e così via. Per il sistema Microsoft si può partire, per i casi più semplici, da un file Microsoft Word, in cui siano integrati gli eventuali contenuti grafici, audio, video; una funzione specifica permette, quindi, di generare un file in formato .lit (lo standard dei libri elettronici per Microsoft), che a quel punto sarà leggibile con il Microsoft Reader. Come per il caso Adobe, però, per la gestione dei diritti bisogna passare ancora attraverso un sistema DRM Microsoft. Per la realizzazione di prodotti più complessi, si parte da contenuti creati in HTML (*Hyper Text Markup Language*), in un programma di *word processing* o in un programma di impaginazione; poi li si elabora con un software di conversione in formato .lit (un esempio: *ReaderWorks* della Overdrive); infine, si sottopone il file .lit a un sistema DRM.

## 3. GLI SCENARI FUTURI DELL'E-BOOK

La prospettiva sembra rosea, ma con un ostacolo fondamentale: l'editore di libri non ha, normalmente, fonti di sostentamento al di là della vendita dei propri prodotti. E non è facile convincere il resto del mondo che ciascuno deve pagare qualcosa per scaricare dalla rete un file che può copiare a costi praticamente nulli da un amico, senza fatica e senza perdita di contenuti o di qualità. Il semplice imperativo morale certo non basta.

La soluzione proposta sotto il nome di e-book è una variante degli schemi di protezione del software contro la copia: il file che l'editore mette in rete non è "in chiaro", ossia liberamente leggibile da chiunque abbia un adeguato programma di lettura o un dispositivo dedicato, bensì è cifrato, e per leggerlo bisogna possedere la chiave di decifrazione. Anche le **tecniche crittografiche**, nel mondo digitale, sono diventate relativamente semplici da applicare e si possono stabilire con altrettanta facilità schemi piuttosto complessi. Ovviamente, se il file fosse cifrato una volta sola, basterebbe scambiarsi insieme al file la chiave di decifrazione per aver aggirato ancora una volta il sistema di protezione del diritto alla copia. Il "trucco" sta nel cifrare il file in modo diverso ogni volta, e assegnare a ogni utente una chiave di decifrazione diversa, per di più legata, nella sua costruzione, a qualche elemento distintivo che non possa essere trasferito ad altri, o che ciascuno abbia interesse a non divulgare: il numero di serie univoco del dispositivo fisico che si userà per leggere il file, per esempio, o magari il numero della carta di credito con cui si paga il libro elettronico.

Se l'utente possiede un dispositivo dedicato, quello che si può chiamare un *e-book reader*, tutto è reso più facile: grazie ai numeri di serie dei suoi componenti hardware quell'apparecchio è identificato in modo

Nella **crittografia** "classica" dall'algoritmo (chiave) utilizzato per cifrare un testo in chiaro si può ottenere l'algoritmo (chiave) per la decifrazione, che è esattamente il procedimento inverso. Un esempio elementare: in un cifrario a sostituzione, alla A del testo in chiaro si può sostituire la F, alla B la R e via di questo passo; per poter decifrare il testo cifrato, basta sostituire alla F la A, alla R la B e così via. Le due chiavi, di cifratura e decifrazione, sono "simmetriche", l'inversa una dell'altra. Molte fra le tecniche crittografiche moderne si basano sull'esistenza di funzioni matematiche *non invertibili*, per le quali cioè non esiste la funzione inversa: tali funzioni permettono di costruire sistemi crittografici molto robusti, per i quali l'algoritmo di cifratura può essere addirittura reso pubblico perché dalla sua conoscenza non è possibile derivare un algoritmo di decifrazione. Nel sistema della *crittografia a chiavi pubbliche*, ciascuno è dotato di due chiavi, una pubblica e una privata: quella pubblica viene divulgata, quella privata no. Se X vuole inviare a Y un'informazione riservata, la cifra utilizzando la sua chiave privata e la chiave pubblica di Y. Y potrà decifrarla usando la chiave pubblica di X e la propria chiave privata: la chiave pubblica di X gli permette di verificare che l'informazione è stata cifrata proprio da X (quindi, di *autenticare* il mittente del messaggio), la sua chiave privata garantisce la segretezza. In un sistema basato su un principio di questo genere, se X è chi vende un libro elettronico, poniamo l'editore, Y un utente e l'informazione da trasferire è un e-book, l'editore cifra il libro elettronico con la propria chiave privata e con la chiave pubblica dell'utente; l'utente potrà fruire la sua copia dell'e-book decifrandolo con la chiave pubblica dell'editore e con la propria chiave privata. Le chiavi possono essere assegnate all'utente caso per caso; in un sistema generalizzato potrebbero essere assegnate una volta per tutte (potrebbero essere, per esempio, quelle della firma digitale).

univoco e gli si possono trasmettere, via Internet, file cifrati in modo che solo quell'apparecchio possa decifrarli e leggerli. Il software incorporato nel dispositivo renderà tutte le operazioni trasparenti all'utente, che non si accorgerà neanche della complessità della tecnologia crittografica all'opera – scoprirà, effettivamente, la sua esistenza solo nel momento in cui cercherà di passare a qualcun altro una copia dei suoi libri elettronici, perché su un altro dispositivo non risulteranno leggibili. Su un personal computer o un pocket PC, macchine di uso generale e, dunque, sostanzialmente più aperte, il procedimento diventa un po' più delicato, ma identico nella sostanza.

**FIGURA 2**
*Il Rocket book*

**FIGURA 3**
*Dimensioni contenute e schermo di alta qualità*

Al di là della curiosità suscitata inizialmente, però, gli e-book reader (Figura 2) (come il Rocket, il primo prodotto di questo tipo) non hanno conquistato il mercato: una delle cause è sicuramente il prezzo elevato, nell'ordine delle centinaia di Euro, dovuto fra l'altro alla necessità di disporre, in dimensioni contenute, di uno schermo di alta qualità, in grado di visualizzare abbastanza bene i caratteri (Figura 3). Schermi portatili, anche se relativamente pesanti, con possibilità di collegamento a un PC attraverso porte seriali o ottiche, permettono un'esperienza di lettura abbastanza gradevole e comoda. Si possono anche portare a letto come un libro di carta, e addirittura permettono di leggere senza accendere la luce, grazie alla retroilluminazione – il che per qualcuno può essere un modo interessante di risolvere problemi di convivenza con il *partner*. I software di cui sono dotati rendono semplicissimi gli equivalenti del voltare pagina avanti e indietro, del saltare all'indice, del mettere segnalibri; in più hanno funzioni di ricerca e i più avanzati permettono di aggiungere brevi annotazioni.

I due programmi di lettura, **Microsoft Reader e Acrobat e-book Reader della Adobe** che si contendono le preferenze degli utenti di personal computer hanno tutte quelle caratteristiche e qualcuna in più – in costante aumento con il succedersi delle versioni. Poiché entrambi vengono distribuiti gratuitamente, l'investimento iniziale non è un ostacolo, e la diffusione di personal e pocket computer dovrebbe garantire una base di potenziali utilizzatori molto ampia. Una premessa, questa, su cui hanno basato le loro iniziative parecchi editori, Apogeo, Fazi, Mondadori fra i primi in Italia, con risultati però deludenti che non sembrano, però, riguardare la lettura sullo schermo in sé lo dimostrerebbe il numero di libri elettronici gratuiti scaricati quanto, piuttosto, una permanente diffidenza verso gli schemi di protezione e verso i pagamenti *on-line* e certamente un po' di disagio nei confronti dei sistemi di generazione delle chiavi e dell'acquisto di oggetti "virtuali".

Basta tutto questo a spiegare quello che fin

I due programmi più diffusi per la lettura di e-book sono **Acrobat e-book Reader della Adobe e il Microsoft Reader**. La Adobe ha puntato su una evoluzione del diffusissimo Acrobat Reader, la Microsoft ha creato un prodotto *ad hoc*. Le differenze, al di là dell'aspetto esteriore, per l'utente sono poche e variabili da versione a versione. La differenza sostanziale sta nel tipo di file che i due programmi accettano in ingresso, e cioè nel formato dei file: Adobe usa il suo formato PDF, Microsoft ha optato per un formato basato su HTML. Il PDF è orientato alla pagina, cioè rappresenta i documenti con formati di pagina fissi; il formato Microsoft invece è pensato per un'impaginazione dipendente dal dispositivo di visualizzazione. Il primo ha quindi una certa rigidità, che tradisce le sue origini nel mondo della stampa, ma assicura il mantenimento dell'impaginazione prevista all'origine; il secondo permette più facilmente di far passare "contenuti" da un dispositivo all'altro (per esempio, dallo schermo di un personal computer a quello di un pocket PC, che hanno dimensioni del tutto diverse) adattando la visualizzazione. Ciascuno dei due formati ha, quindi, vantaggi e svantaggi; come stanno le cose in questo momento, si potrebbe dire che PDF è più vicino alla mentalità dello stampato tradizionale e meglio si presta per la riproduzione di libri o riviste in cui l'impaginazione ha un'importanza non trascurabile; il formato Microsoft è più libero, si presta meglio a contenuti informativi per i quali l'impaginazione è un fattore secondario. Il formato Microsoft, per esempio, si adatterebbe senza fatica agli schermi dei cellulari di nuova generazione, ma a costo di rinunciare, per così dire, ad assegnare a ogni documento una sua fisionomia grafica stabile.
Certo, anche se la natura di fondo di un formato limita in qualche misura le sue potenzialità, è possibile che nelle versioni future dei loro programmi Adobe e Microsoft trovino il modo di aggirare, più o meno elegantemente, le rispettive limitazioni. Resta il fatto che la presenza di due formati in concorrenza non aiuta l'editore, che non ha di per sé motivi per "sposare" l'uno a scapito dell'altro, e attualmente si vede costretto a prevedere la resa dei suoi "contenuti" in entrambe le modalità. Per risolvere il problema, la tendenza prevalente – ma non esclusiva – è quella di lavorare a un livello di maggiore astrazione, strutturando i contenuti con un linguaggio di marcatura come XML (*eXtensible Markup Language*), da cui poter derivare con le opportune trasformazioni automatiche entrambi i formati (e, in futuro, qualunque altra soluzione si presenti).
La presenza dei due formati non aiuta neanche il lettore, naturalmente: il quale può, sì, procurarsi entrambi i programmi senza particolare fatica e senza costi, ma rimane inevitabilmente perplesso quando scopre che di alcuni libri esiste solo uno dei due formati, mentre di certi altri esiste solamente il formato concorrente.

qui si può definire l'insuccesso degli e-book? Probabilmente no. Più di vent'anni fa, un programma come Visicalc ha fatto fare rapidamente il salto al personal computer, da *hobby* per appassionati di elettronica a strumento di lavoro e di utilità: era una versione elettronica dei fogli dei contabili, ma con un paio di aggiunte, ovvero la possibilità di inserire nelle celle non solo parole e numeri, ma anche formule. I calcoli automatici hanno reso subito evidenti a tutti i vantaggi. Anche i programmi per la lettura degli e-book estendono il concetto di "libro", ma evidentemente non in misura altrettanto efficace. I vantaggi delle funzioni digitali di ricerca a tutto testo, dell'evidenziazione, dei segnalibri, delle annotazioni sono tali sono in situazioni particolari, nello studio e nel lavoro professionale. Per chi si limita a leggere un romanzo o un saggio, questi vantaggi sono senza effetto.
I libri elettronici disponibili, peraltro, sono nella quasi totalità semplici riversamenti dei "vecchi" libri di carta e non sfruttano tutte le potenzialità della tecnologia digitale (Figura 4): nulla impedisce, in linea di principio, di creare nuovi "libri" in cui accanto al testo trovino posto audio, video, immagini in movimento, programmi che consentono l'interazione. In realtà, però, qualcuno l'ha già fatto: basta pensare alle enciclopedie multimediali, che generalizzano in modo efficace l'enciclopedia cartacea. Curiosamente, però, mentre le enciclopedie di carta sono considerate libri, le enciclopedie multimediali sono "titoli multimediali" e non libri elettronici. Sarà solo una questione terminologica, ma l'uso linguistico è una spia di un atteggiamento.
Produrre libri elettronici che sfruttino tutte le potenzialità del digitale comporta un'ulteriore, drastica trasformazione delle attività editoriali, e porta con sé un cambiamento del ruolo dell'autore. Si potrebbe, per esempio, pensare di scrivere un libro su Georg Friedrich Haendel, il grande musicista tedesco del Settecento, prevedendo la riproduzione di immagini (quadri, lettere ecc.), rimandando alla lettura delle partiture che sono rimaste e facendo riferimento, infine, a incisioni disponibili in CD, ad allestimenti testimoniati su videocassetta o

**FIGURA 4**
*Un esempio di e-book attualmente disponibile sul mercato*

Dvd. Ma se la destinazione di questo lavoro non è la carta, ma si tratta, invece, di un libro elettronico, si può fare di più: raccontando di come il suo *Rinaldo* abbia avuto varie versioni negli anni, si può prevedere, per esempio, l'inclusione di spezzoni audio che mettano a confronto passi particolari dell'opera nelle diverse redazioni e, infine, è possibile esplorare, anche, la storia recente dell'interpretazione prevedendo l'inserimento di spezzoni di video che illustrino i diversi allestimenti, le diverse scenografie, le diverse interpretazioni drammatiche. È un po' come parlare di un quadro potendolo effettivamente mostrare – cosa che la tecnologia tradizionale della stampa permette di fare. Lavori di questo genere difficilmente possono essere opera di un solo autore, e richiedono all'editore competenze in settori che tradizionalmente erano separati.

Alla fine il prodotto sarà "un libro scritto" su Haendel, o sarà un'altra cosa? Forse sta proprio in questo il bello del libro elettronico, che può essere *qualcosa*, o magari *molto* più di un libro di carta. È sempre difficile azzardare previsioni, ma è certamente possibile che i "lettori" si conquistino realizzando opere nuove, pensate sin dall'inizio come opere da fruire con mezzi digitali, lavori che non potrebbero avere mai un equivalente perfetto su libri cartacei.

Se questa è la strada, il meccanismo degli e-book, con le sue tecniche crittografiche per la protezione dei diritti d'autore e di copia risulterà forse più accettabile e comprensibile. Si deve chiedere, però, alla tecnologia informatica strumenti a supporto degli autori più adatti di quelli odierni: meno legati al testo dei word processor e più facili da usare dei programmi di regia multimediale.

La situazione attuale è molto fluida: tutti gli operatori legati al mondo editoriale, non più sotto i riflettori, sperimentano e fanno circolare nuove idee. Difficilissimo, ovviamente, dire quali sperimentazioni e quali idee abbiano più probabilità di dare i frutti più duraturi, ma alcune prospettive sono davvero intriganti. Tanto per citarne solo un paio, si può riflettere su dove può portare la diffusione della larga banda e dell'Internet *always on*. Nel momento in cui essere sempre collegati alla Rete fosse la norma, che senso avrebbe "scaricare" libri elettronici, brani musicali o altro, per averne una copia privata? Tranne che per i collezionisti, quel che conta, come è stato suggerito, non è il possesso ma l'accesso: se il possesso di un libro di carta si può giustificare anche con il piacere di ammirare l'oggetto, toccarlo, apprezzarne la fattura, tutti questi fattori si annullano inevitabilmente per un oggetto virtuale. Potrebbero avere un senso grandi "biblioteche" elettroniche a cui si accederebbe a piacere, con sezioni libere e altre protette per le opere coperte da copyright, e con schemi di pagamento per abbonamenti o a "consumo". Non è neanche un'idea tanto nuova: è una variante aggiornata del progetto Xanadu di cui si era fatto alfiere Ted Nelson negli anni Ottanta. Vale ancora la pena andare a rileggere il suo utopistico *Literary Machines*.

C'è poi un'innovazione che, se uscisse finalmente dall'ombra e si diffondesse, potrebbe convincere anche molti fra i più refrattari: l'*e-paper*, ovvero la "carta elettronica". Si tratta di fogli di plastica flessibile dotati di *microchip*, con piccole sferette metà bianche e metà nere magnetizzabili racchiuse all'interno: le configurazioni di magnetizzazione delle sferette permettono di comporre caratteri e grafica e il risultato sono fogli che consentono la lettura in modo molto simile ai fogli di carta stampati. A quel punto sì l'analogia con il libro tradizionale sarebbe quasi perfetta. Si scarica dalla rete un libro elettronico, lo si legge sulla

carta elettronica e, quando lo si è letto, lo si cancella e si scarica qualcos'altro. E, a differenza di quel che succede con gli e-book reader della prima generazione, che hanno uno schermo di dimensione fissa, e relativamente piccola, si possono pensare fogli di dimensioni diverse per libri di natura diversa, riproducendo anche i diversi formati dei libri attuali.

## Bibliografia


[1] Barbier F, Barbier B, Lavenir C: *La storia dei media. La comunicazione da Diderot a Internet*. Christian Marinotti Edizioni, Milano, 2002.

[2] ContentLink: www.contentlinkinc.com

[3] D'Anna R: *e-Book. Il libro a una dimensione*. Adnkronos libri, Roma, 2001.

[4] E-book: www.ebooks.com

[5] Evolutionbook: www.evolutionbooks.com

[67 Open eBook Forum: www.openebook.org

[7] Rosenblatt B, Trippe B, Mooney S: *Digital Rights Management*. M&T Books, New York, 2002.

[8] Sala VB: *e-book. Dal libro di carta al libro elettronico*. Apogeo, Milano, 2001.

[9] Toschi L (a cura): *Il linguaggio dei nuovi media*. Apogeo, Milano, 2001.



VIRGINIO B. SALA è direttore editoriale della Apogeo di Milano e della Bononia University Press. Laureato in filosofia, lavora nell'editoria dal 1972. I suoi interessi vanno, oltre che alle nuove tecnologie della comunicazione, alla musica.
virginio@apogeonline.com

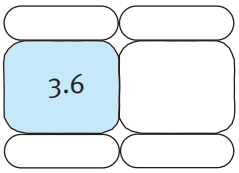

**Digital Object Identification**

Lo schema oggi diffuso per l'identificazione degli oggetti digitali in Internet è ancora legato alla loro collocazione fisica: http://www.tizio.com/caio/sempronio/documento.htm, per esempio, indica un documento (una pagina web in HTML) su un server che si chiama tizio.com. Questo nome è solo una denominazione, comoda per gli esseri umani, di un indirizzo fisico preciso e che identifica un sistema specifico. Se quel documento (che potrebbe essere un testo, una fotografia, un campione audio o altro ancora) viene fisicamente spostato, non lo si può più raggiungere se non si conosce il suo nuovo indirizzo, e quindi la sua collocazione fisica effettiva. Da tempo si sta cercando un modo diverso di denominare gli oggetti digitali, che sia indipendente dalla loro collocazione effettiva: uno dei più interessanti fra i progetti internazionali in corso è quello che va sotto il nome di *Digital Object Identification* (DOI, www.doi.org), iniziativa avviata nel 1994 dall'AAP (*Association of American Publishers*). Un identificatore DOI è costituito da due parti, un prefisso e un suffisso separati da una barra o *slash* (/); il prefisso è costituito a sua volta da due parti, il numero della *directory* DOI (per il momento ne esiste una sola, identificata dal numero 10 – in decimale) e il numero che identifica l'editore; il suffisso è scelto liberamente dall'editore e può essere qualsiasi stringa di lettere, numeri e caratteri di interpunzione, purché univoca (cioè diversa da ogni altra usata da quell'editore). L'identificatore non fa alcun riferimento a una collocazione fisica dell'oggetto: di questa tiene traccia, invece, la directory DOI, dove a ogni identificatore DOI viene associata la sua posizione nella Rete. Nel momento in cui il DOI si generalizzasse, si potrà accedere a qualsiasi oggetto digitale semplicemente indicandone il nome; la directory DOI svolgerà il compito di "risolvere" l'identificatore inviando alla posizione fisica dell'oggetto (a quello che oggi è il suo URL). Il vantaggio di questo schema è che, se la collocazione di un oggetto muta, l'indicazione della sua nuova sede deve essere inviata solo alla directory; tutti i possibili utenti continuano a riferirsi al nome, che accompagna l'oggetto per tutta la sua vita. Lo sviluppo di un sistema di identificazione generale (non importa che sia il DOI o qualche altro metodo) è indispensabile per sistemi di DRM generalizzati.

# SISTEMI FUZZY

**Andrea Bonarini**

I sistemi fuzzy sfruttano la possibilità offerta dagli insiemi fuzzy e dalla logica fuzzy di rappresentare modelli in termini simbolici che abbiano uno stretto legame con le realtà misurate. Sono, quindi, modelli di facile comprensione e di progettazione relativamente immediata. Vengono sempre più spesso adottati per applicazioni che spaziano dal controllo di impianti industriali all'interpretazione delle immagini, dalla classificazione dei dati all'analisi del segnale, dal controllo di elettrodomestici alle applicazioni spaziali.




## 1. FUZZY?

Quando introdusse il concetto di *fuzzy set* 33 anni fa [4], Lofti Zadeh, iraniano emigrato a Berkeley, poté cogliere solo in parte l'impatto che questo avrebbe avuto sui settori più disparati, dal controllo, alle basi di dati, dalla modellistica, alla programmazione dei calcolatori, alla realizzazione di oggetti di consumo appetibili e affidabili. Le applicazioni fuzzy nel mondo sono ormai milioni e ne vengono realizzate di nuove in numero sempre crescente. Controllori fuzzy fanno parte dei sistemi ABS (*Antilock Braking System*) delle automobili, nelle lavatrici, aspirapolvere e videocamere, nei *robot* che giocano a calcio e in quelli industriali, nei treni della metropolitana e nelle sonde spaziali. Sistemi fuzzy supportano decisioni economiche e sistemi di controllo di qualità, permettono di trovare informazioni in rete e in basi documentali, supportano nella selezione del personale e nell'interpretazione di immagini.
Da dove viene questo successo?
Zadeh, uno dei ricercatori che negli anni '60 aveva contributo allo sviluppo delle teorie dell'automazione, a un certo punto si accorse che i modelli matematici che venivano utilizzati in quell'ambito non erano adeguati a rappresentare semplici concetti di uso comune, come "caldo" o "alto". Avrebbe voluto includere della conoscenza comune nei suoi modelli, ma i concetti qualitativi che venivano usati dagli esperti mal si adattavano alla forma matematica fino ad allora disponibile per rappresentare modelli. Erano anche gli anni in cui venivano sviluppati i primi sistemi esperti e in cui si cominciava a formalizzare la conoscenza. Era chiaro da secoli come i concetti in questione potessero essere rappresentati con classi; ad esempio, il concetto di "caldo" può essere definito elencando degli oggetti caldi, i quali costituiscono un insieme definito proprio dal fatto che i suoi membri hanno quella proprietà. La teoria degli insiemi è alla base della matematica e della rappresentazione della conoscenza e Zadeh propose un'estensione del concetto di insieme, l'insieme fuzzy, dalla quale si è poi giunti a estendere l'aritmetica, la logica, le misure, costruendo gli strumenti per infinite

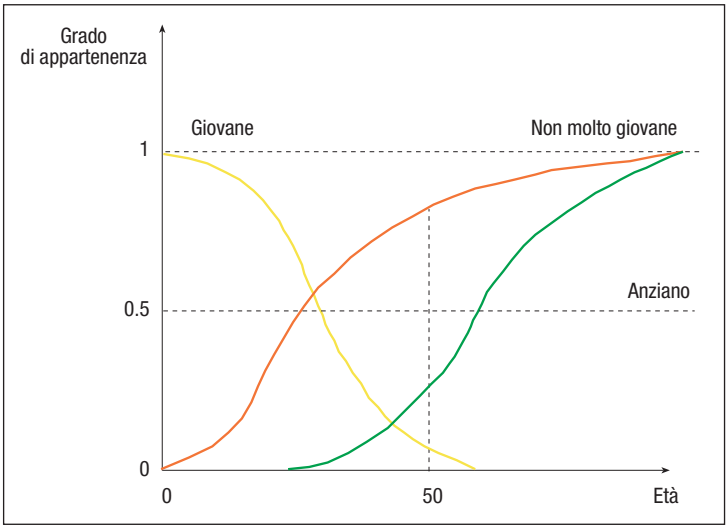



**FIGURA 1**
*Alcune funzioni di appartenenza*

applicazioni, tutte in grado di rappresentare concetti in modo analogo a quanto fanno gli esseri umani.

## 2. COS'È UN INSIEME FUZZY?

Per capire cos'è un insieme fuzzy si consideri dapprima il concetto di insieme tradizionale, che nel seguito verrà chiamato insieme *crisp*, per contraddistinguerlo dagli insiemi fuzzy. Un insieme è composto da tutti gli elementi dell'universo del discorso (l'ambito che interessa al modellista) che soddisfano una data funzione di appartenenza. Per un insieme crisp, la funzione di appartenenza è booleana, cioè associa ad ogni elemento x dell'universo del discorso un valore "vero" o "falso" a seconda che x appartenga o non appartenga all'insieme. La funzione di appartenenza definisce, quindi, l'insieme e può avere diverse forme. Per esempio, l'insieme dei laureati è definito da una funzione di appartenenza che ritorna "vero" se e solo se la persona x ha conseguito una laurea. Esistono però concetti più qualitativi. Per esempio, a che età una persona è considerabile come "giovane"? Fino a 20 anni? A 30? A 50? Certamente, una persona di 20 anni è considerabile come "giovane" più di una persona di 40... Diviene, allora, importante poter definire quanto un elemento dell'universo del discorso possegga una certa proprietà, o, in altri termini, quanto possa appartenere all'insieme degli elementi che posseggono quella proprietà. È possibile, dunque, definire una funzione di appartenenza per un insieme, che ritorni un valore nell'intervallo da 0 "falso" a 1 "vero". Questo permette di definire "quanto" si ritiene che un elemento dell'universo del discorso appartenga all'insieme, cioè permette di dare un grado di appartenenza all'insieme non necessariamente booleano.

Nella figura 1, è riportato un esempio di alcune funzioni di appartenenza definite sulla variabile età che caratterizza gli elementi dell'universo del discorso. Si noti la funzione che definisce l'insieme delle persone giovani, quella relativa alle persone anziane, e quella relativa all'insieme delle persone non molto giovani che, come si vedrà, si può ricavare dalla prima applicando opportuni operatori.

È possibile vedere che una persona di 50 anni, secondo le definizioni date da chi ha definito queste funzioni di appartenenza, appartiene sia all'insieme delle persone non molto giovani, con un grado di appartenenza relativamente elevato, sia all'insieme delle persone giovani con grado basso, sia all'insieme delle persone anziane, pure con grado basso. L'appartenenza a entrambi questi ultimi due insiemi potrebbe sembrare una contraddizione (come può una persona essere allo stesso tempo giovane e vecchia?), ma non lo è per la definizione che è stata data dei due insiemi, che hanno le funzioni di appartenenza parzialmente sovrapposte. Questa possibilità permette di utilizzare insiemi fuzzy per rappresentare concetti in ambienti in cui la rilevazione degli stessi può essere approssimata o incerta, e dà un grande potere descrittivo.

La definizione di insieme fuzzy non è che un'estensione della definizione classica di insieme. L'insieme fuzzy ha una frontiera che non è più una linea netta di demarcazione tra gli elementi che appartengono all'insieme e quelli che non vi appartengono, ma un'area in cui si trovano elementi classificabili come appartenenti all'insieme con un certo grado. Per questo l'insieme è detto "fuzzy", cioè "sfumato". In figura 2, è possibile notare alcuni esempi di insiemi crisp e di insiemi fuzzy. Si noti che in linea di principio insiemi crisp e insiemi fuzzy si possono intersecare, in quanto entrambi operano nello stesso universo del discorso e possono esistere elementi che appartengono a entrambi.

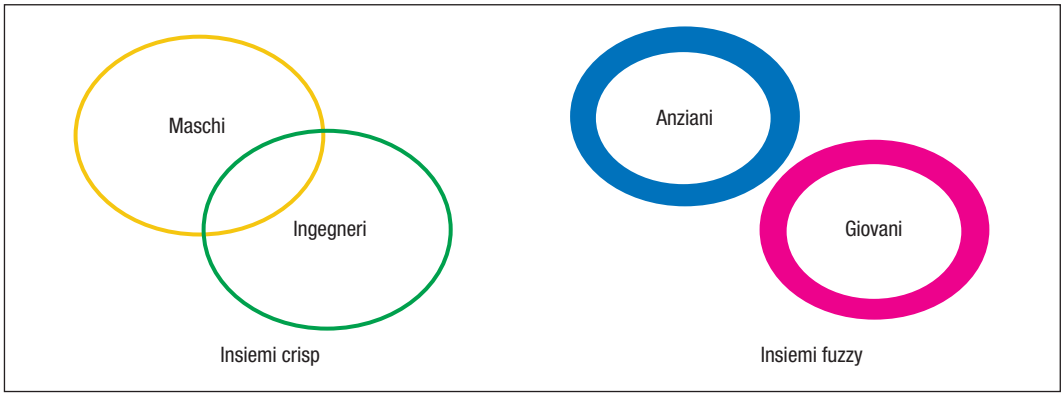



**FIGURA 2**
*Due insiemi crisp e due insiemi fuzzy. In linea di principio insiemi crisp e insiemi fuzzy si possono intersecare, in quanto entrambi operano nello stesso universo del discorso*

Il termine scelto da Zadeh per denominare i nuovi insiemi da lui definiti, rendeva conto della loro peculiare caratteristica, ma costituì uno dei primi ostacoli alla diffusione di queste idee: chi avrebbe voluto usare delle cose "sfuocate" quando c'erano tanti modelli matematici a disposizione, precisi e ben definiti? Nonostante questo, il messaggio fu colto, dapprima, in Europa, dove nel 1974 Ebrahim Mamdani definì delle regole basate sugli insiemi fuzzy per poter realizzare il controllore del reattore di un cementificio [3]. Questo è un impianto di difficile modellizzazione matematica, che veniva controllato manualmente con successo da operatori che ragionavano in termini di "temperatura alta" e "poca acqua", reagendo a una situazione descritta in termini qualitativi con azioni del tipo "aggiungere poco cemento" o "aumentare la portata dell'acqua di molto". Mamdani codificò queste regole in termini fuzzy, che permettevano di interpretare i numeri forniti dai sensori (o forniti agli attuatori) con i termini linguistici usati dagli operatori: era nato il controllo in linguaggio naturale.

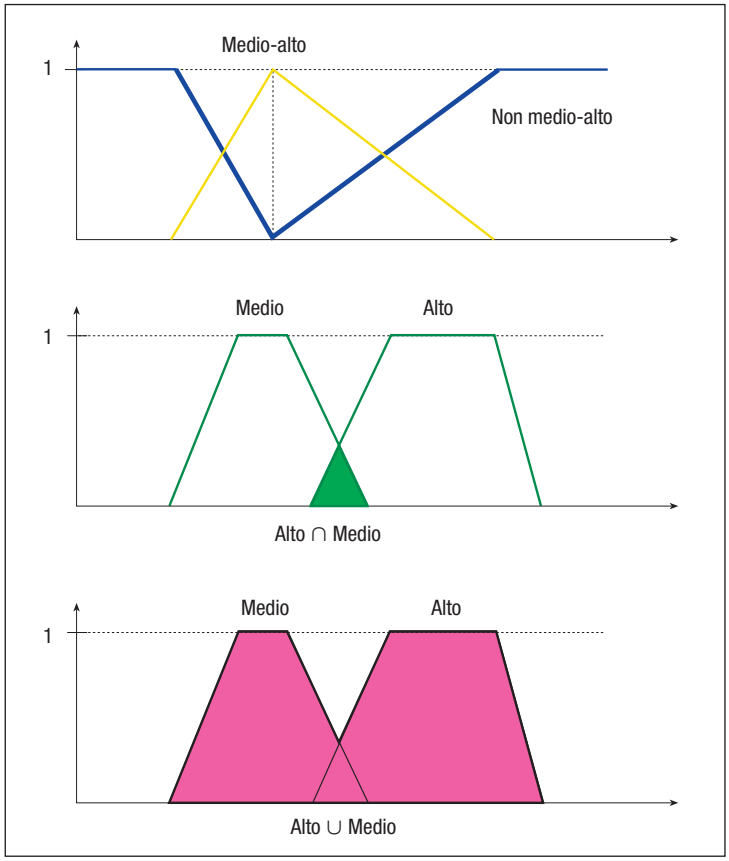



**FIGURA 3**
*Composizione di funzioni di appartenenza per gli operatori insiemistici*

## 3. COSA SONO LE REGOLE FUZZY?

Le regole proposte da Mamdani, e da allora largamente usate, non sono altro che un modo per mettere in relazione una descrizione di una situazione in termini linguistici con un'azione da svolgere, espressa anch'essa a parole. Sono basate su un'altra estensione di un concetto matematico che segue immediatamente dall'estensione degli insiemi: la logica fuzzy. I valori di verità di una proposizione non sono più solamente "vero" o "falso", ma possono variare in un insieme ordinale, normalmente nell'insieme dei numeri reali da 0 a 1. Analogamente, si estendono tutti i concetti caratteristici della logica. Gli operatori logici, negazione (not), disgiunzione (or) e congiunzione (and) vengono ridefiniti in modo da poter fornire valori coerenti con il loro significato anche per *operandi* con valori non booleani, ma reali. L'estensione segue, immediatamente, la definizione dei loro corrispondenti nella teoria degli insiemi (complemento, unione, intersezione (Figura 3)).

Per esempio, la negazione viene definita semplicemente come il complemento a 1 della funzione di appartenenza all'insieme complementare. Indicando con $\mu_A(x)$ la funzione di appartenenza di un elemento x all'insieme fuzzy A e con $\bar{A}$ il complementare di A, si ha che $\mu_{\bar{A}}(x) = 1 - \Omega\mu_A(x)$.
In logica fuzzy sono poi stati definiti tutti gli altri elementi della logica, quali quantificatori, modificatori, meccanismi inferenziali, sui quali non è il caso di dilungarsi in questo contesto, rimandando i lettori interessati alla letteratura specifica [1]. Per dare un'idea della potenzialità di queste estensioni si può citare il fatto che i modificatori, che hanno il ruolo di modificare il valore di verità di una proposizione, in logica fuzzy sono in linea di principio infiniti, mentre in logica booleana si ha la sola negazione, dato che la modifica di un valore booleano (per esempio, "vero") può solo essere l'altro "falso". Per esempio, "molto" è un modificatore che deforma la funzione di appartenenza, e di cui si è vista un'applicazione nella definizione dell'insieme "non molto giovane" in figura 1. I quantificatori, che in logica classica sono il quantificatore esistenziale (Esiste un x tale che...) e il quantificatore universale (Per ogni x ...) in logica fuzzy possono essere infiniti, come ad esempio: "Esistono molti x tali che...", "Per quasi tutti gli x...", Per qualche x...". In queste frasi, tutti i termini ("molti", "quasi tutti", "qualche") possono essere definiti da altrettanti insiemi fuzzy che ne rappresentano il significato in maniera ben definita. Anche i meccanismi inferenziali sono stati fuzzyficati, introducendone le versioni valide per una logica con valori di verità continui. Per esempio, il fuzzy *modus ponens* è una versione del classico meccanismo usato per dedurre informazione da informazione nota, in cui gli elementi coinvolti possono avere valori di verità compresi tra 1 e 0. Se ne vedranno degli esempi qui di seguito.
A questo punto, è possibile rispondere alla domanda: "Cosa sono le regole fuzzy?"
Sono rappresentazioni di inferenze logiche fatte su composizioni di proposizioni fuzzy, nella forma (X è LX), dove X è una variabile detta "variabile linguistica" in quanto assume come valori i nomi di insiemi fuzzy, e LX è l'etichetta che denota un insieme fuzzy definito per quella variabile. Sono costituite da un antecedente, da cui viene ricavato, per inferenza fuzzy, il valore di verità di un conseguente. Anche se antecedenti e conseguenti, in linea di principio, possono essere composti da una qualunque combinazione di proposizioni fuzzy, nella stragrande maggioranza delle applicazioni ci si limita ad avere regole in cui gli antecedenti sono in congiunzione tra di loro e i conseguenti pure. Si avranno, quindi, delle regole nella forma:

**if** (X is LX) **and** (Y is LY) **and** ... **and** (Z is LZ)

**then** (U is LU) **and** ... **and** (V is LV)

Per esempio, si può avere una regola che dice che "se la temperatura è alta e la pressione è alta, allora la quantità di gasolio da fornire al bruciatore dell'impianto deve essere ridotta di molto". Questo permetterà di ricavare il valore dell'uscita a partire dai valori degli ingressi rilevati, cioè di effettuare un'inferenza fuzzy, come si vedrà nell'esempio presentato nella prossima sezione.

## 4. UN ESEMPIO DI FUNZIONAMENTO DELLE REGOLE FUZZY

Nella figura 4, sono rappresentate due variabili linguistiche A e B, definite su un insieme di sette valori ciascuna. I valori sono insiemi fuzzy, definiti da funzioni di appartenenza le cui etichette indicano rispettivamente Negativo grande, Negativo medio, Negativo piccolo, Zero, Positivo piccolo, Positivo medio e Positivo grande. Si noti che questi insiemi sono definiti su un asse di valori reali corrispondenti a due variabili reali associate alle variabili linguistiche: grazie all'interpretazione dei valori reali in termini di insiemi fuzzy si potrà usare l'insieme di regole fuzzy per mappare valori reali su valori reali, attraverso un modello espresso in termini linguistici. Si noti, inoltre, che le funzioni di appartenenza hanno tutte la stessa forma e che si distribuiscono uniformemente sull'intervallo di definizione delle variabili reali associate. Questa è una scelta di progetto comune, che tra l'altro ottimizza la resistenza del modello al rumore, ma non è affatto obbligato-

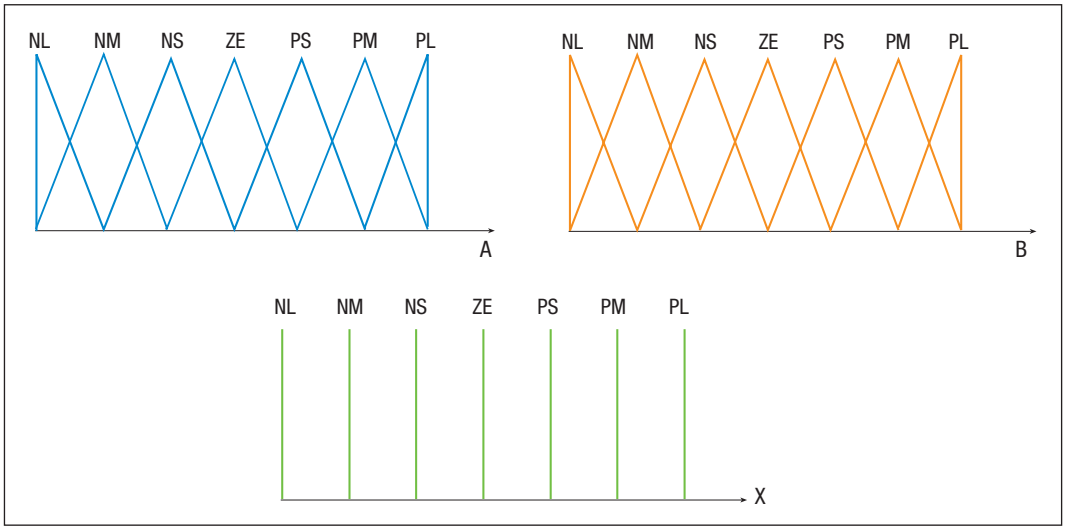



**FIGURA 4**
*Le variabili nelle regole fuzzy dell'esempio*

ria: il progettista può definire le forme che più sono adatte a rappresentare i concetti coinvolti e distribuirle a piacere sull'intervallo di definizione. Esistono anche molti strumenti di ottimizzazione o di apprendimento automatico del numero, della forma e della distribuzione delle funzioni di appartenenza, che si occupano di ottenere prestazioni ottimali lavorando sui dati a disposizione e sollevando il progettista da un compito altrimenti oneroso.
Nella figura 4, si ha anche un'altra variabile linguistica, usata per l'uscita, che assume valori su un insieme di insiemi fuzzy con una forma particolare. Si tratta di insiemi fuzzy contenenti un solo elemento ciascuno, senza ulteriori frontiere ("singleton" o "singoletti"). L'uso di questo tipo di valori fuzzy per i conseguenti di regole fuzzy è pratica comune nel controllo fuzzy e non inficia particolarmente la qualità del modello, semplificandone, invece, la computazione che, spesso, nel caso del controllo, è fatta su *microchip* di bassissimo costo e complessità.
Le regole che sono considerate nell'esempio che segue sono:

R1: **if** (A is PL) **and** (B is PS) **then** (X is PM)

R2: **if** (A is PM) **and** (B is PS) **then** (X is PS)

R3: **if** (A is PL) **and** (B is PM) **then** (X is PM)

Si noti come vengono usate queste regole in una tipica applicazione di controllo in cui gli ingressi numerici vengono interpretati in termini fuzzy, attivano le regole fuzzy che generano uscite fuzzy che vengono alla fine riportate in termini numerici per essere trasmesse agli attuatori. In altri termini, il controllore fuzzy mappa ingressi reali in uscite reali attraverso un modello linguistico.

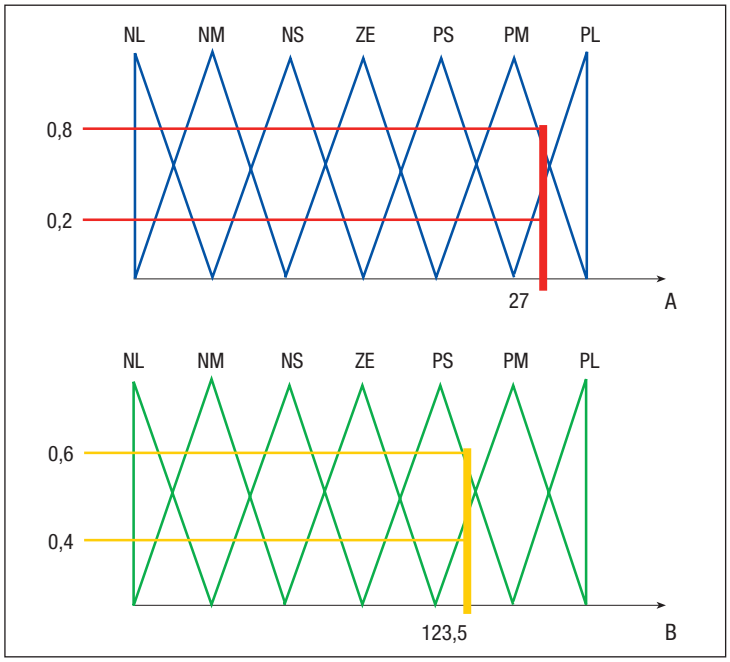



**FIGURA 5**
*Passo 1: fuzzyficazione*

### PASSO 1 FUZZYFICAZIONE (MATCHING)

I valori reali in ingresso vengono interpretati in termini di fuzzy set (Figura 5). Per la variabile A, il valore in ingresso è appartenente all'insieme PL con grado 0,2 e all'insieme PM con grado 0,8. Per la variabile B, il valore in ingresso è PM con grado 0,4 e PS con grado 0,6.

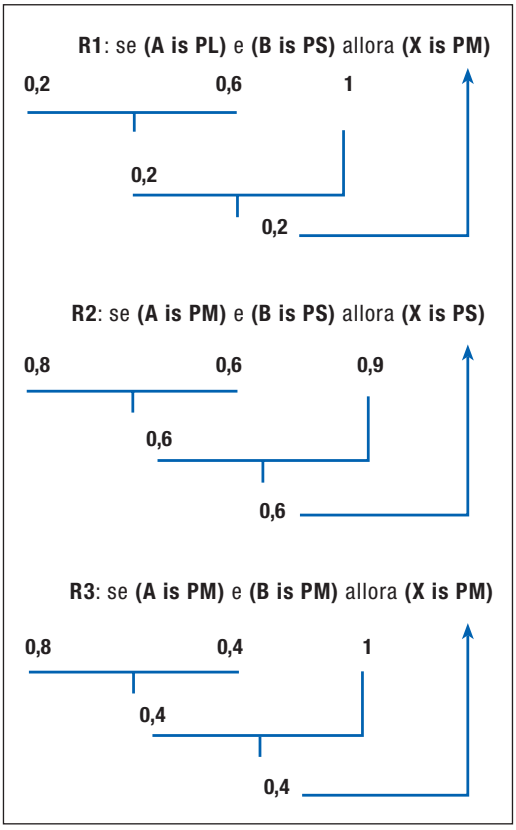



**FIGURA 6**
*Passi 2 e 3: combinazione dei valori nell'antecedente e trasmissione del valore al conseguente*

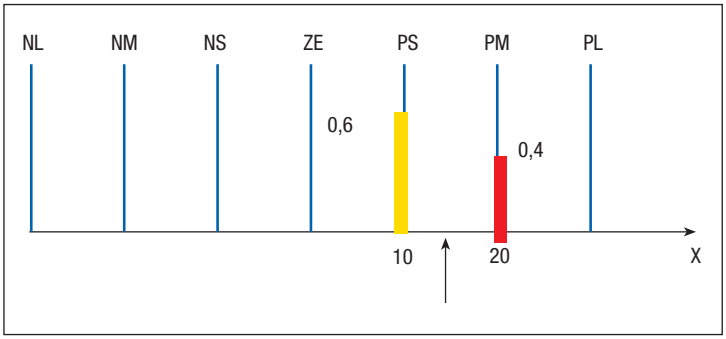



**FIGURA 7**
*Defuzzyficazione*

### Passo 2 Combinazione dei valori nell'antecedente

I valori delle diverse clausole nell'antecedente vengono combinati con l'operatore "and" (congiunzione) per ottenere il valore complessivo dell'antecedente, che indica in sostanza quanto le condizioni di attivazione della regola sono ben rappresentative dei valori di ingresso al sistema. L'operatore utilizzato è qui il minimo e i risultati sono riportati in figura 6.

### Passo 3 Trasmissione del valore al conseguente

Ottenuto il valore dell'antecedente, occorre trasmetterlo al conseguente. In generale, è possibile associare un valore alle regole che ne indichi l'importanza. È il numero che in figura 6 è riportato sotto al termine "then". Questo viene combinato con il valore dell'antecedente, ancora una volta con un operatore che, anche qui, sarà il minimo. Il valore così ottenuto passa al conseguente. Il significato dell'operazione è il seguente: il conseguente ha un peso che dipende da quanto la regola è adeguata per rappresentare una certa situazione definita dagli ingressi e da quanto è importante in generale. Nell'esempio in questione, pur essendo presente una regola con valore di importanza diverso da 1, questo non ha avuto alcun effetto in quanto la combinazione degli antecedenti era già più piccola, e l'operatore utilizzato (il minimo) seleziona il valore più piccolo.

### Passo 4 Combinazione dei valori dei conseguenti

Come si può notare anche in questo semplice esempio, può darsi che ci siano diverse regole che hanno lo stesso conseguente ma antecedenti diversi. Questo può provocare il fatto che i conseguenti siano associati a pesi diversi. Occorre aggregare i diversi pesi calcolati ottenuti per ogni conseguente e questo viene fatto con un operatore, nell'esempio, il massimo. Il risultato finale dell'inferenza fuzzy risulta così essere:

(X is PM) con grado 0,4

(X is PS) con grado 0,6

### Passo 5 Defuzzyficazione

Il processo inferenziale è terminato, ma ci si può riportare a un valore numerico per la variabile di uscita associata alla variabile linguistica presente nel conseguente. Questo è quanto fatto, ad esempio, nel controllo fuzzy. L'operazione si chiama *defuzzyficazione* ed è effettuata con un operatore scelto in una vasta classe. In questo esempio, viene scelta la semplice media pesata, o baricentro, ottenendo il valore riportato nella figura 7 e calcolato come:

$$(10 \cdot 0{,}6 + 20 \cdot 0{,}4)/(0{,}6 + 0{,}4) = 14$$

Per riassumere e focalizzare quanto presentato nell'esempio, si può supporre che A rappresenti un errore di temperatura e B un errore di pressione in un processo industriale e che X sia una variabile di controllo, per esem-

pio la quantità di acqua di raffreddamento per il processo, rispetto a un riferimento. Sono state utilizzate delle regole che stabilivano relazioni tra queste grandezze, ad esempio R1: "Se l'errore di temperatura è positivo ed elevato, e l'errore di pressione è positivo e piccolo, allora la quantità di acqua di raffreddamento è positiva e media rispetto al valore di riferimento". Applicando queste regole agli errori di temperatura e pressione introdotti nel sistema come numeri reali, si è ottenuto un valore finale della portata che è a sua volta scelto nell'insieme dei reali. È stato, cioè, utilizzato un modello simbolico, costituito da termini linguistici e regole per calcolare una relazione tra ingressi e uscite reali.

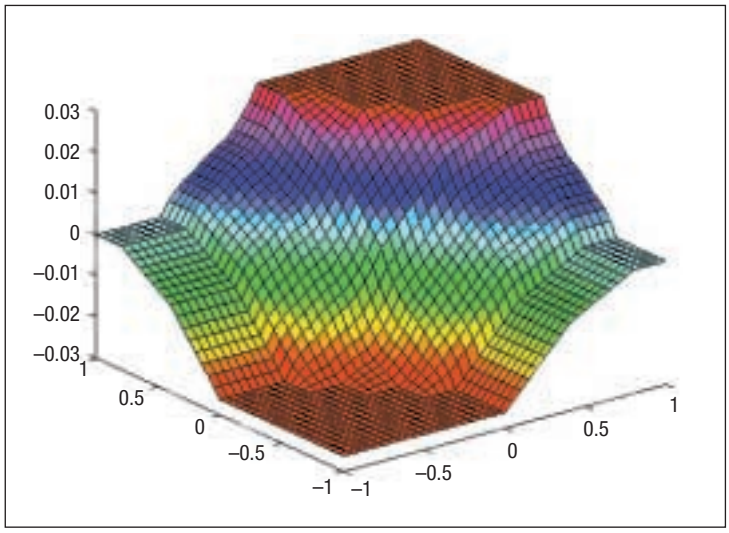



**FIGURA 8**
*Una superficie di controllo fuzzy per un braccio robotizzato*

## 5. I MODELLI FUZZY

Un insieme di regole fuzzy costituisce un modello, espresso in termini linguistici, che mette in relazione ingressi e uscite. Si è appena visto che ingressi e uscite possono essere numeri reali e che, quindi, il modello è, in linea di principio, applicabile anche laddove vengono applicati modelli matematici. Inoltre, come si è visto, il modello esprime in maniera precisa dei concetti qualitativi. Non si tratta, quindi, come a volte capita di sentire da chi non conosce questi sistemi, di un modello "raffazzonato" o "definito malamente"; si tratta, invece, di un modello definito in maniera formale, precisa, in termini di relazioni matematiche tra funzioni matematiche, le funzioni di appartenenza. In generale, è un modello non lineare, che ha numerose proprietà interessanti, tra cui una buona robustezza al rumore, e la possibilità di definire l'uscita desiderata in maniera ottimale per ogni valore degli ingressi. Si veda, per esempio, una tipica superficie di controllo in figura 8: in ascisse e ordinate rispettivamente errore e derivata dell'errore di uno degli assi del braccio e sull'asse z l'azione di controllo. Si vedrà più avanti come queste proprietà ne facciano un ottimo strumento per controllare sistemi complessi, difficilmente identificabili in modo sufficientemente preciso in termini matematici.
Proprio queste proprietà hanno portato al successo il controllo fuzzy, che ha conosciuto negli anni '80 un vero *boom* in Giappone, si dice per la maggior similitudine del modello con il modo di pensare dei giapponesi, rispetto ai modelli matematici. In quegli anni, furono realizzati controlli fuzzy di treni, di elicotteri, di impianti industriali e anche di oggetti di uso comune come aspirapolvere, macchine fotografiche, lavatrici e perfino mini-forni a vapore per cuocere il riso (*rice-cookies*).
Negli anni '80 un altro pioniere dei sistemi fuzzy, il giapponese Michio Sugeno, aveva definito un nuovo tipo di regole fuzzy aprendo ulteriori possibilità di modellizzazione [4]. Queste regole hanno antecedenti del tutto analoghi a quelli delle regole di Mamdani, permettendo di definire con termini linguistici le situazioni in cui vengono attivate. Il conseguente è, invece, un modello, espresso in termini dei valori (crisp) degli ingressi. Per esempio:

**if** (A is LA) **and** (B is LB) **and**... **then** U is f (*A*, *B*)

con f (*A*, *B*) = a + b*A* + c*B*

dove a, b e c sono costanti, mentre *A* e *B* sono i valori numerici delle variabili associate ad A e B all'ingresso. In questo esempio, il modello che compare nel conseguente della regola è un semplice modello lineare, ma negli anni successivi sono stati introdotti come conseguenti modelli di ogni tipo, da polinomi a reti neurali. Questo tipo di regole, in altri termini, permette di definire in termini linguistici le condizioni in cui appli-

care un modello. Inoltre, dato che le condizioni sono fuzzy, è possibile combinare le uscite dei modelli presenti nelle diverse regole con gli operatori di combinazione visti sopra, ottenendo così una variazione graduale (*smooth*) nell'intervallo tra l'applicazione di un modello e l'applicazione di un altro. L'utilizzo di queste regole non solo ha introdotto nuove possibilità di modellizzazione, ma ha anche permesso di provare delle proprietà importanti per sistemi fuzzy, quali, ad esempio, la stabilità del controllore.
Il boom del controllo fuzzy in Giappone ha portato questa tecnologia all'attenzione di tutto il mondo e le applicazioni sviluppate sono aumentate enormemente negli anni facendo entrare di forza la tecnologia fuzzy nel bagaglio di base dell'ingegnere e del modellista.

## 6. NON SOLO CONTROLLO

Modelli fuzzy sono usati non solo per il controllo, ma anche in tutti quei settori dove è necessario definire modelli robusti sia a partire da informazioni qualitative, sia a partire da dati, utilizzando magari sistemi di sintesi automatica del modello.
Tra i molteplici settori applicativi se ne citano, nel seguito, solo alcuni.
**Rappresentazione dell'incertezza e modelli di conoscenza.** Le teorie fuzzy sono state usate per definire una pletora di modelli e misure per la rappresentazione dell'incertezza [1] (possibilità e necessità, plausibilità e credenza e molte altre), usati nei settori più disparati, anche laddove le teorie probabilistiche non possono essere applicate. Basandosi su misure di incertezza si possono, inoltre, definire modelli di conoscenza in cui si tenga conto dell'incertezza nei dati e nel modello stesso. Questo permette di presentare come uscita del modello risultati con pesi diversi, invece di dover per forza giungere a un unico risultato nel tentativo di eliminare ogni contraddizione. Si pensi, ad esempio, a un sistema di diagnosi, in cui i dati siano affetti da incertezza o anche solo approssimati, come potrebbe essere nella diagnosi medica, ma anche nella diagnosi industriale di sistemi in condizione critica (come un impianto in fase di avviamento). È certamente preferibile ottenere una serie di possibili diagnosi pesate da un opportuno indicatore di plausibilità, piuttosto che una sola diagnosi che potrebbe non essere quella corretta a causa di errori nelle rilevazioni.
**Sistemi di supporto alle decisioni.** Spesso un decisore deve prendere decisioni basandosi su informazioni qualitative derivate da osservazioni personali. In altri casi, si basa su valutazioni sintetiche ricavate da masse di dati, magari con un processo automatico di data *mining*. Gli insiemi fuzzy ben si prestano a caratterizzare entrambi i tipi di informazione, permettendo di esplicitare modelli altrimenti taciti. Si pensi, per esempio, a una decisione di marketing strategico in cui il decisore pensa in termini di alta o bassa ricettività del mercato, alto o basso rapporto prezzo/costo di produzione ecc. In questi casi, i dati numerici possono essere interpretati e aggregati in maniera fuzzy e un sistema di supporto alle decisioni può presentare risultati emulando il ragionamento del decisore, opportunamente codificato in un sistema fuzzy.
**Sistemi di information retrieval.** Nella ricerca di informazioni in documenti e in rete è interessante poter esprimere richieste "*soft*", magari associate a indicatori linguistici o numerici relativi all'importanza data ai vari termini. I motori di ricerca combinano con operatori fuzzy questi indicatori e le caratterizzazioni fuzzy delle richieste, con altri indicatori risultanti dall'operazione di ricerca e possono così fornire delle risposte che soddisfano l'utente. Si possono, così, esprimere richieste del tipo: "Vorrei indicazioni sui prezzi delle auto veloci e con bassi consumi apparse sul mercato negli ultimi tempi". Un'analoga richiesta trattata con sistemi basati sul concetto di intervallo richiede una definizione rigida dei concetti "veloce", "basso" "ultimi tempi", introducendo la possibilità di esclusione di risultati interessanti che con un sistema fuzzy sarebbero, comunque, tenuti in considerazione, con gli opportuni pesi.
**Sistemi di interpretazione del segnale.** Anche in questo caso, si tratta di modelli che permettono di valutare caratteristiche del

segnale. Ad esempio, si può avere un sistema di visione in cui vengono riconosciuti oggetti per il loro colore. Il colore è, generalmente, rappresentato da una tripletta di valori numerici. Anche qui, è possibile definire gli insiemi fuzzy su queste variabili che identificano il colore desiderato. Altre caratteristiche di segnali possono essere identificate in termini fuzzy per portare a una classificazione degli stessi. Ad esempio, si potrebbe dire che "se la frequenza del segnale è alta, e i picchi sono elevati, ma irregolari, allora il segnale corrisponde a una situazione non critica".

**Sistemi di previsione di serie storiche.** In questo caso, una delle modalità di applicazione di sistemi fuzzy consiste nel rappresentare in termini fuzzy particolari caratteristiche degli andamenti della serie storica (per esempio, l'andamento delle quotazioni di mercato di un bene) e nell'associarvi dei modelli previsionali, magari ricavati con sistemi automatici quali reti neurali.

## 7. APPLICAZIONI FUZZY

Vengono presentate, ora, più in dettaglio, alcune applicazioni paradigmatiche che permetteranno di capire le potenzialità di questa tecnologia, non solo nel settore del controllo, ma anche in quello dell'*information retrieval*, dei sistemi di supporto alle decisioni e nella modellistica in generale.

### 7.1. Controllo della frenata di un treno di una metropolitana e del volo di un elicottero: un'azione di controllo per ogni situazione

Esistono situazioni in cui il sistema da controllare ha configurazioni e caratteristiche diverse nel tempo. È, quindi, necessario definire un sistema di controllo per ogni condizione di esercizio. Se queste cambiano continuamente e se il cambiamento non è direttamente misurabile, è possibile definire un insieme di regole fuzzy, ognuna in grado di coprire una particolare situazione. La combinazione delle regole fuzzy permette di avere un controllo che varia in continuità al variare delle variabili di ingresso. Per questi motivi, il controllo della frenata del treno della metropolitana di Sendai, in Giappone, fu realizzato negli anni '80 con tecniche fuzzy. Il treno era in grado di frenare in maniera precisa e molto morbida qualunque fosse il carico di passeggeri. Ne risultò un maggior *comfort* per i passeggeri e un risparmio notevole di energia, rispetto alla frenata realizzata da operatori umani.

Un altro esempio è il controllo di un mini elicottero, effettuato da Sugeno sempre negli anni '80. L'elicottero era in grado di eseguire comandi vocali con buona precisione, pur sottoposto a raffiche di vento non costanti, che cambiavano di continuo le condizioni operative.

### 7.2. Controllo di una lavabiancheria: controllo efficace a basso costo su beni di largo consumo

Grazie alla possibilità di realizzare controllori fuzzy su *chip* a bassissimo costo e alla loro buona robustezza nei confronti di segnali di cattiva qualità provenienti da sensori a basso costo, sistemi fuzzy sono stati usati per rendere più semplice l'utilizzo di beni di consumo quali lavatrici, aspirapolvere, condizionatori, telecamere, automobili, frullini ecc. Così, si hanno sul mercato lavatrici (anche di produttori italiani, divenuti *leader* di questa tecnologia) in grado di capire autonomamente che tipo di panni stanno lavando e di adeguare di conseguenza il ciclo di lavaggio, tra l'altro sciacquando solo finché è necessario, con notevoli risparmi di acqua ed energia. Si hanno aspirapolvere in grado di ridurre la potenza del motore quando sono puntate contro delle tende, e aumentarla sul pavimento. Si hanno telecamere in grado di mantenere a fuoco oggetti in movimento e di ridurre le accidentali vibrazioni della mano.

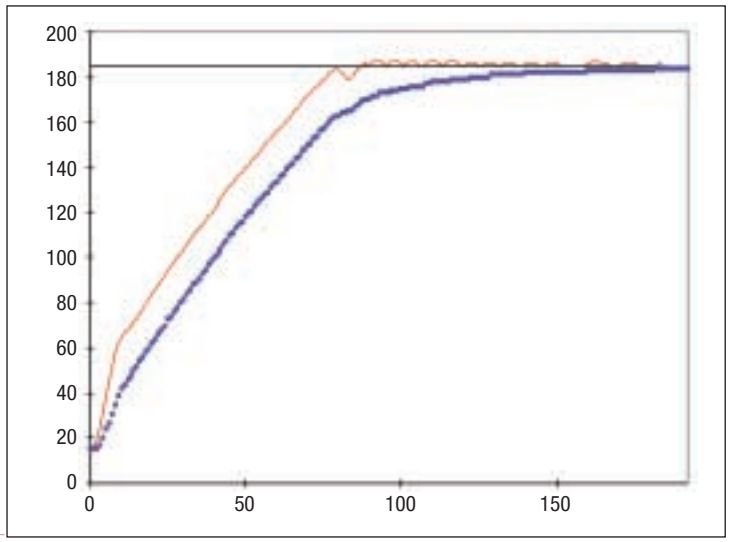
A titolo d'esempio, si veda come una lavatrice, a un costo bassissimo e compatibile con le esigenze di mercato, può capire che tipo di panni deve lavare. Ogni lavatrice carica acqua finché un semplice interruttore pneumatico (pressostato) non viene attivato dalla pressione stessa dell'acqua nella vasca. Quando l'acqua entra nella vasca viene assorbita dai panni con un tempo di solito superiore a quello di attivazione del pressostato, e diverso a seconda del tipo di tessuto usato. Dopo un certo tempo, si ha, quindi, una diminuzione della quantità d'acqua

nella vasca a spese dell'acqua assorbita dal tessuto. Il pressostato scatta e l'acqua riprende a entrare. Misurando i tempi e la quantità di attivazioni del pressostato si può ottenere, senza sensori aggiuntivi, una caratterizzazione del tipo di panni immesso, che viene usata per adattare il programma di lavaggio, ottimizzandolo e risparmiando all'operatore umano l'onere della scelta del programma.
Anche la regolazione della quantità di risciacqui può essere fatta a costo bassissimo. L'acqua nella vasca costituisce un dielettrico per un condensatore le cui armature sono la vasca e un elemento metallico isolato da essa. L'acqua ha caratteristiche dielettriche diverse a seconda che contenga o meno sapone. Viene così misurato il tempo di carica del condensatore-lavatrice con l'acqua al termine del lavaggio e si continua a sciacquare fino a quando questo tempo non è qualitativamente vicino al tempo di carica misurato all'entrata dell'acqua fresca all'inizio del lavaggio.
In entrambi questi casi, il modello è qualitativo e affetto da rumore: si presta, dunque, bene a essere realizzato con regole fuzzy.

### 7.3. Controllo di un impianto industriale: modello matematico difficile da identificare

Esistono impianti industriali il cui modello matematico è difficile da definire o da identificare con precisione, come per esempio, reattori chimici come quello oggetto del primo controllo fuzzy industriale citato all'inizio, laminatoi, forni ecc. In questi casi, in genere, ci sono operatori umani in grado di governare l'impianto manualmente utilizzando regole qualitative basate sull'esperienza. Il controllore fuzzy può sostituirli, o affiancarli, catturandone e rendendone esplicita l'esperienza.
Anche qui si veda un esempio in dettaglio. Si tratta di un forno per l'invecchiamento di alluminio, in cui bisogna portate un carico di trafilati di alluminio a una temperatura di stabilizzazione, possibilmente senza introdurre gradienti termici tra i due estremi dei trafilati, e senza produrre sovrariscaldamenti eccessivi. Il tutto nel più breve tempo possibile. In questa applicazione, il forno ad aria forzata era controllato considerando le temperature dell'aria all'ingresso e all'uscita del carico. Nell'azienda si avevano più di 10.000 tipi diversi di trafilati con diverse caratteristiche di assorbimento termico e non si voleva appesantire le operazioni di carico e scarico con procedure che indicassero il tipo di carico effettuato. In questo caso, è stato realizzato un controllo fuzzy che cattura una logica semplice e applicabile a qualunque tipo di carico, al quale il comportamento del controllore si adatta naturalmente. Espressa a parole la logica suona così: "Finché la differenza di temperatura tra ingresso e uscita dell'aria ($\Delta T$) è contenuta entro i limiti desiderati, si scaldi con la massima intensità. Se il limite di $\Delta T$ è raggiunto si tenga quell'intensità nel riscaldamento. Se la temperatura in ingresso è vicina alla temperatura obiettivo si mantenga la temperatura obiettivo".
Con un insieme di meno di 10 regole si è così ottenuto un controllo efficace, in grado di rispondere, come desiderato, ai diversi carichi. Un esempio di andamento del riscaldamento è riportato in figura 9, dove si ha la curva rappresentante la temperatura dell'aria calda in ingresso, sempre superiore a quella rappresentante la temperatura dell'aria in uscita dai trafilati. Si noti come si tratti di curve che sarebbero difficilmente rappresentabili con una funzione matematica, la quale eventualmente avrebbe un'espressività molto minore delle poche regole linguistiche menzionate, e una difficoltà di identificazione e manutenzione molto maggiori. Inoltre, la curva dovrebbe essere diversa per

**FIGURA 9**
*Andamento delle curve di riscaldamento per il forno di invecchiamento di trafilati di alluminio*



ogni tipo di carico e questo richiederebbe l'identificazione di parametri in linea di principio difficili da identificare.

### 7.4. Sistema di controllo di qualità: dati rilevati da operatori in modo qualitativo

Nel controllo di qualità il prodotto viene sottoposto a un'analisi per verificarne caratteristiche salienti ed eventualmente intervenire sulla regolazione dell'impianto per ottenere il prodotto con la qualità desiderata. In molti casi, le misure sono fatte da operatori che danno delle valutazioni qualitative delle caratteriste in gioco. Si pensi, ad esempio, alla valutazione della birra (colore, tipo e quantità di bolle, amarezza ecc.). Questi operatori possono fornire un'indicazione fuzzy a un sistema esperto basato su regole fuzzy che possa suggerire il tipo di intervento da fare sull'impianto.
In altri casi, giudizi qualitativi su aspetti specifici possono essere aggregati con operatori fuzzy per fornire indicazioni globali in maniera formalmente corretta. Per esempio, sono stati proposti dei sistemi di valutazione semiautomatica di siti *web*, in cui i criteri di valutazione si basano in parte su misure obiettive rilevate automaticamente (lunghezza della pagina, numero di *link* ecc.) e in parte su valutazioni fornite da operatori. Le caratteristiche considerate vengono poi composte con operatori fuzzy e generano indicatori globali di valutazione.
Un caso che rientra ancora in questa categoria è quello del sistema di valutazione del personale all'assunzione, sviluppato per una delle più grandi aziende italiane. Anche qui, le valutazioni qualitative vengono aggregate con operatori fuzzy a informazioni numeriche fuzzyficate. Uno dei maggiori ritorni rilevati con l'adozione di questo sistema è la possibilità di rendere esplicito, uniforme e obiettivo il processo di valutazione.

### 7.5. Controllo di una squadra di robot autonomi: modello linguistico dell'ambiente ottenuto da dati sensoriali imprecisi e affetti da incertezza

In questo caso, si ha una squadra di robot autonomi che deve svolgere un compito, ad esempio giocare una partita di calcio robotico, come avviene, dal 1997, nella competizione internazionale *Robocup*. Ogni robot si costruisce un modello del mondo sulla base dei dati sensoriali che riceve. Il sistema di controllo comportamentale del robot è in grado di scegliere tra diversi comportamenti ("porta la palla in porta", "passa", "chiudi l'avversario" ecc.) basandosi su una valutazione in termini di predicati fuzzy della situazione in cui si trova ("palla avanti", "avversario lontano" ecc.). Anche il meccanismo di coordinamento della squadra e di scelta delle strategie è basato su analoghi strumenti, e può essere realizzato da strateghi umani con relativa facilità. L'applicazione può sembrare ludica, ma è di notevolissima complessità, e l'adozione di modelli fuzzy permette di affrontarla con regole di buon senso.

## 8. COME SI FANNO I SISTEMI FUZZY?

La realizzazione di sistemi fuzzy è supportata da strumenti a diversi livelli.
Esistono numerosi sistemi di sviluppo per sistemi di controllo fuzzy, caratterizzati da regole che classificano un ingresso numerico e forniscono un'uscita fuzzy tradotta poi in termini numerici. Il supporto consiste nella possibilità di utilizzare forme predefinite e parametrizzate di funzioni di appartenenza, nella definizione di regole e nella visualizzazione dei valori ottenuti dal sistema fuzzy in tutto l'intervallo dei valori di ingresso. Alcuni di questi sistemi di supporto forniscono anche un'interfaccia di ottimizzazione che, tramite l'applicazione di reti neurali ai dati del problema, è in grado di suggerire la posizione e il numero ottimale di funzioni di appartenenza, cioè di effettuare autonomamente la parametrizzazione del sistema.
Controllori fuzzy vengono realizzati sia su PC (*Personal Computer*), sia sui microcontrollori usati per applicazioni tradizionali, sia su microcontrollori progettati e prodotti per ottimizzare le operazioni fuzzy necessarie a ottenere l'uscita. Il costo di questi chip è compatibile con i costi sopportabili per la loro adozione in prodotti normalmente con bassi margini di guadagno, quali i beni di consumo.

Sistemi di supporto alle decisioni e modelli più articolati possono essere invece realizzati con sistemi di sviluppo *ad hoc*, facilmente integrabili nei sistemi informativi in cui devono operare.

## 9. CONCLUSIONI

In conclusione, è possibile affermare che i sistemi fuzzy hanno ormai da tempo raggiunto una maturità tale che li rende un ottimo strumento per una larga quantità di applicazioni. La loro diffusione è ancora, in parte, ostacolata da una loro scarsa conoscenza e da una ingiustificata diffidenza nei confronti di uno strumento così semplice e potente. Con la diffusione della conoscenza sulla loro struttura e funzionamento, un numero sempre maggiore di applicazioni viene affrontato, in un numero sempre più grande di settori applicativi.

## Bibliografia


[1] Klir GJ, Yuan B: *Fuzzy Sets and Fuzzy Logic. Theory and Applications*. Prentice Hall, Upper Saddle River, N J, 1995.

[2] Mamdani EH, Assilian S: An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-machine Studies*, Vol. 7, 1975, p. 1-13.

[3] Sugeno M: I*ndustrial Applications of Fuzzy Control*. Elsevier, Amsterdam, NL, 1985.

[4] Zadeh LA: Fuzzy sets. *Information and Control*, Vol. 8, 1965, p. 338-353.

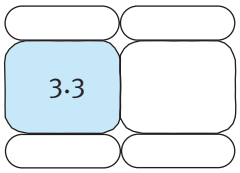


ANDREA BONARINI è professore associato presso il Politecnico di Milano e coordina dal 1984 il locale Laboratorio di Intelligenza Artificiale e Robotica. I suoi interessi di ricerca riguardano: sistemi fuzzy, sistemi di apprendimento per rinforzo e sviluppo di robot mobili autonomi di servizio, e di intrattenimento (Robocup – calcio robotico). Andrea.Bonarini@polimi.it

# EVOLUZIONE E PROSPETTIVE NELL'HIGH PERFORMANCE COMPUTING

**Ernesto Hofmann**

Attività quali le previsioni meteorologiche o la simulazione di processi nucleari sono state per molto tempo effettuate con supercomputer costruiti con una tecnologia denominata *high performance computing* (HPC). Tuttavia, i grandi progressi ottenuti nella costruzione e nell'aggregazione in *cluster* dei microprocessori hanno consentito di costruire nuovi tipi di supercomputer. In questo articolo, si esamina come l'HPC si sia evoluto e quali siano le prospettive tecnologiche e applicative che si possono intravedere per il futuro.




## 1. PREMESSA

Si immagini di aver costruito un modello computerizzato del pianeta terra e di entrare come in un videogioco nella realtà virtuale di questo modello per esplorare paesaggi, città, strade, case e uffici. *Il computer* potrebbe consentire di controllare, in ogni istante, la densità della popolazione, la distribuzione dei prodotti agricoli, la disponibilità dei beni di consumo, le congestioni del traffico, e tutto ciò su qualunque scala.
In linea di principio, ingrandendo per gradi l'immagine, si potrebbe entrare in ogni ufficio o in ogni negozio per osservarne le più minute attività. Si esaminerebbero in dettaglio, e in tempo reale, ogni transazione effettuata in una specifica agenzia bancaria così come l'acquisto di ogni singola bottiglia in un negozio di vini. Si potrebbe così osservare l'attività dell'economia di una nazione, o di una città, identificandone eventuali inefficienze e possibili miglioramenti.
Rick Stevens, direttore del dipartimento di matematica e *computer science* del *National Argonne Laboratory* di Chicago, ha denominato questo modello computerizzato il *Mirror World*, ossia il mondo riflesso in uno specchio. Dal punto di vista concettuale, un simile modello non ricade in quelle classi di problemi che per la loro natura sembrano, come si dice in linguaggio matematico, "intrattabili", ma richiede, evidentemente, un'estrema capillarità di dispositivi periferici e una prodigiosa capacità di calcolo.
Intuitivamente, si comprende che per attuarlo è necessario poter eseguire enormi quantità di istruzioni in breve tempo. Ma, volendo essere più concreti, quante? Questa domanda introduce, quasi naturalmente, al tema dei computer più potenti, ovvero dei *supercomputer*. Cosa sono i supercomputer e in cosa si distinguono dai grandi computer utilizzati, per esempio, per la gestione della prenotazione posti di una compagnia aerea o dei conti correnti di una grande banca? Per capire le diverse esigenze di un grande computer e di un supercomputer si cercherà di fare un raffronto, ancorché semplificato, tra due possibili scenari.
Si prenda, innanzitutto, il caso di una grande

banca. Le banche, notoriamente, utilizzano computer di tipo *general purpose*, ossia disegnati per un ampio spettro di possibili applicazioni, in grado di servire in maniera sicura, e pressoché istantanea, una vasta clientela.
Un cliente di una banca, che si presenti a uno sportello *bancomat* e che avvii una transazione per ottenere dallo sportello stesso un certa quantità di denaro, cosa provoca nel computer di quella banca? Deve, innanzitutto, connettersi e ciò crea un segnale che, attraverso la rete, perviene al computer il quale, a sua volta, eseguirà complesse sequenze di istruzioni per interrompere l'attività in corso, gestire quell'interruzione, decidere, quindi, se accodare tale richiesta o eseguirla direttamente, il che comporterà una serie di transazioni logico-fisiche tra computer e utente. Queste consisteranno nell'inserimento e nel riconoscimento della *password*, in una specifica richiesta di servizio, per esempio banconote per un certo ammontare e, infine, nell'eventuale emissione di una ricevuta.
Tutto ciò richiede, per dirla in maniera molto approssimativa, l'esecuzione di centinaia di migliaia, se non milioni, di istruzioni da parte del computer centrale, che deve anche accedere a specifici archivi elettronici per aggiornare i dati relativi a tale cliente.
Si consideri che una grande banca può avere migliaia di punti bancomat e che questi possono collegarsi anche quasi simultaneamente. Il computer deve, quindi, poter eseguire complessivamente molti miliardi di istruzioni in decine di secondi per fornire tempi di risposta accettabili per ciascun utente collegato.
Gli attuali grandi computer, ossia i cosiddetti *mainframe*, sono in grado di fornire questa capacità di calcolo. E, infatti, le banche, come anche le compagnie aeree, le ferrovie, gli istituti di previdenza e molte altre organizzazioni forniscono servizi applicativi molto efficienti.
Un altro tipo di problema, del tutto diverso, meno impegnativo del Mirror World ma quanto mai complesso, riguarda, invece, la biologia e la farmaceutica. Una delle sfide più affascinanti, ma anche più importanti, che oggi la scienza deve affrontare è quella del cosiddetto ripiegamento (il *folding*) delle proteine.
Una proteina è una molecola, costituita da migliaia o decine di migliaia di atomi, che viene costruita all'interno delle cellule usando come stampo il DNA. La proteina, nel tempo stesso nel quale viene costruita, si ripiega su se stessa assumendo una conformazione tridimensionale che è unica e identica per tutte le proteine di quel tipo.
Molte malattie degenerative sono causate da un errato ripiegamento (il *misfolding*) di una specifica proteina. È, quindi, di fondamentale importanza comprendere tale meccanismo di ripiegamento. Ciò, infatti, potrebbe aprire la via alla creazione di farmaci in grado di correggere simili degenerazioni durante la costruzione delle proteine stesse.
È intuibile che il computer sia lo strumento ideale per simulare matematicamente il processo di ripiegamento di una proteina.
Un modello molto semplificato potrebbe, innanzitutto, creare le sequenze di aminoacidi corrispondenti al particolare gene. Poiché la struttura di ogni aminoacido è ben conosciuta il computer è in grado di costruire un modello atomico complessivo della proteina in esame. A questo punto, il computer dovrebbe calcolare tutte le forze esistenti tra i vari atomi e valutare come esse causino il ripiegamento della proteina su se stessa nel liquido nella quale si forma all'interno della cellula.
Il processo di ripiegamento potrebbe essere modellato come una sequenza cinematografica nella quale per ogni fotogramma dovrebbero essere calcolate le forze tra i vari atomi e la progressiva deformazione della proteina in un arco di tempo che, idealmente, potrebbe durare un milionesimo di miliardesimo di secondo. È necessario utilizzare un tempo così breve per descrivere accuratamente le più rapide vibrazioni della proteina e del liquido nel quale la proteina si muove, che sono quelle associate al movimento degli atomi di idrogeno della proteina stessa e dell'acqua presente all'interno della cellula.
Si immagini, a questo punto, che il film duri un decimillesimo di secondo, che è il tempo in cui può ripiegarsi una piccola proteina.
In un simile film, si avranno, quindi, cento miliardi di fotogrammi da calcolare. Se la proteina è costituita, per esempio, da trentamila atomi le interazioni reciproche saranno circa un miliardo. E se per ogni interazione si dovessero eseguire mille istruzioni si otterrà che per ogni fotogramma si dovranno eseguire mille miliardi di istruzioni. Mille miliardi di istruzioni per

cento miliardi di fotogrammi sono, quindi, centomila miliardi di miliardi di istruzioni da dover eseguire in un decimillesimo di secondo. Si immagini, infine, di avere a disposizione un supercomputer che esegua un milione di miliardi di istruzioni al secondo: esso dovrebbe lavorare per cento milioni di secondi, ossia circa tre anni. E oggi i supercomputer più potenti sono almeno cento volte meno potenti di tale supercomputer.
Ma quello del ripiegamento delle proteine è un caso limite. Ci sono, infatti, problemi meno complessi, ancorché tuttora difficili da trattare, come le nuove prospezioni petrolifere, il comportamento delle galassie, i servizi finanziari avanzati, l'*on-line trading*, la simulazione dell'urto di due buchi neri, la progettazione di strutture complesse di edifici, automobili e aerei, la simulazione di esplosioni nucleari, l'evoluzione del clima, l'andamento delle correnti marine, la previsione dei terremoti, e altro ancora (Figura 1).
I supercomputer più potenti oggi disponibili, come il *NEC Earth Simulator* presso l'*Institute for Earth Science* di Yokohama, gli *ASCI/Q* della *HP-Compaq* installati presso il *Los Alamos National Laboratory*, l'*IBM ASCI White* al *Lawrence Livermore Laboratory*, o ancora l'*HP-Compaq Terascale Computing System* al centro di supercalcolo di Pittsburgh, dispongono oggi di capacità di calcolo che sono dell'ordine delle migliaia di miliardi di istruzioni (in virgola mobile) al secondo, ovvero teraflop/s. Quali sono allora le caratteristiche costruttive di questi computer e come si potranno evolvere per poter affrontare problemi ancora più complessi come quelli posti, appunto, dal ripiegamento delle proteine?

## 2. L'AVVENTO DEI SUPERCOMPUTER

Un'istruzione, per essere eseguita dal computer, utilizza, mediamente, alcune decine di circuiti microelettronici. Più rapidi sono questi ultimi più veloce è l'esecuzione dell'istruzione. Ma un'istruzione, per essere eseguita, ha spesso bisogno di utilizzare *operandi* che potrebbero non essere immediatamente disponibili ai circuiti utilizzati dall'istruzione stessa. È, quindi, necessaria una struttura di memoria che consenta ai dati, più frequentemente utilizzati, di essere immediatamente disponibili. Dalla metà degli anni '60 era già evidente che per superare i vari "colli di bottiglia" che si potevano presentare nell'esecuzione di un'istruzione, occorreva una complessa organizzazione circuitale.
Si era compreso che, per alcune classi di problemi, occorrevano computer che si discostassero dall'originale architettura di Von Neumann la cui caratteristica più saliente è la sequenzialità delle operazioni sia nelle unità aritmetico-logiche sia nella gestione della memoria.
A parità di tecnologia circuitale – la cosiddetta tecnologia di semiconduttore – si potevano ottenere più elevate prestazioni se venivano utilizzate, contemporaneamente, molteplici unità di elaborazione in grado di eseguire, in parallelo, gruppi differenti, e opportunamente selezionati, di istruzioni.
Negli stessi anni, si evolveva così un insieme di computer progettati per ottenere prestazioni molto più elevate rispetto ai cosiddetti *general purpose computer*. L'obiettivo di tali supercomputer era, in generale, quello di risolvere problemi caratterizzati dalla grande quantità di dati numerici sui quali occorreva eseguire ripetutamente sequenze di istruzio-

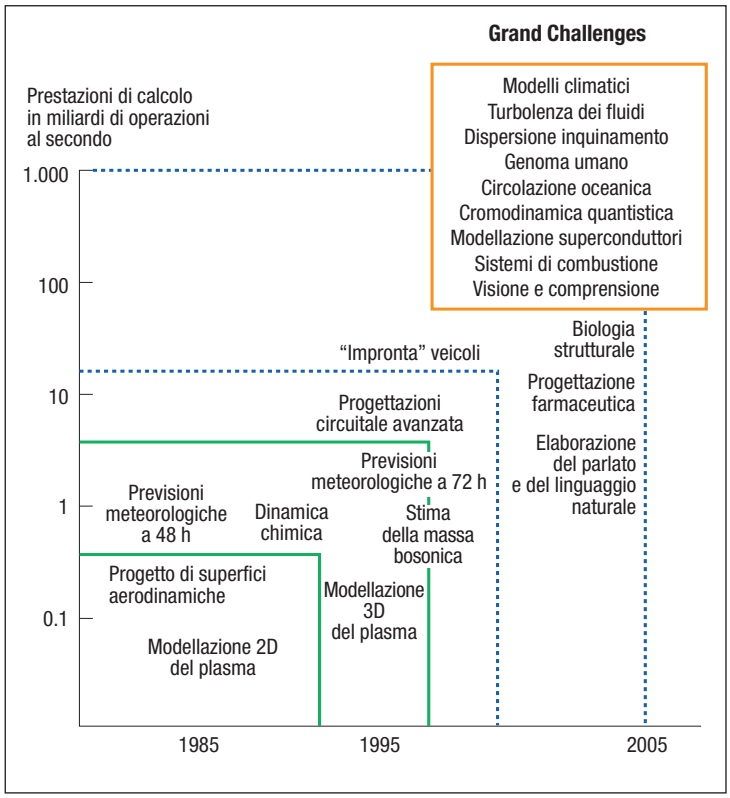



**FIGURA 1**
*Prestazioni di calcolo per problemi "Grand Challenge"*

ni abbastanza simili, piuttosto che la casualità delle sequenze di istruzioni tipiche degli ambienti commerciali.
Ciò che distingueva i supercomputer erano essenzialmente tre elementi: l'utilizzo delle più avanzate tecnologie circuitali, l'uso di molteplici unità di elaborazione e il disegno di opportuni algoritmi.
Alcuni sistemi, come l'ILLIAC IV (costruito per l'Università dell'Illinois dalla Burroughs) e il PEPE (*Parallel Ensemble of Processing Elements dell'U.S. Army Advanced Ballistic Missil Defense*), invece di risolvere i problemi attraverso una serie di operazioni successive eseguivano, simultaneamente, un elevato numero di operazioni.
Un tipico problema che l'ILLIAC IV poteva risolvere, molto più rapidamente rispetto a un computer convenzionale, era la soluzione dell'equazione alle derivate parziali di Laplace che descrive la distribuzione della temperatura sulla superficie di una piastra.
Nell'ILLIAC IV duecentocinquantasei *processing element* erano raggruppati in quattro quadranti di $8 \times 8$. Ogni processing element aveva una sua memoria ed era in grado di comunicare con i quattro processing element adiacenti.
Altri computer, come lo STARAN (della *Goodyear Aerospace Corporation*), utilizzavano il principio delle memorie associative. Nello STARAN una memoria associativa, con un accesso in parallelo a 256 voci di 256 *bit*, era gestita attraverso un aggregato di unità di elaborazioni indipendenti (una per ciascuna voce di memoria). Lo STARAN veniva utilizzato per applicazioni che permettevano un elevato livello di parallelismo intrinseco quali analisi di tracce *radar* o calcolo matriciale.
Intorno alla metà degli anni '60, furono soprattutto due costruttori, la *Control Data Corporation* (CDC) e la IBM, ad avviare una straordinaria sfida industriale per la costruzione di una nuova classe di computer che fossero almeno un ordine di grandezza più potenti rispetto ai più potenti computer per uso commerciale.
Fa ormai parte della leggenda l'aneddoto secondo cui Thomas Watson Jr., presidente della IBM, fosse molto contrariato nell'apprendere - poco dopo l'annuncio, nell'aprile del 1964, del sistema IBM 360 - che la CDC era pronta a costruire anch'essa un computer, il CDC 6600, più veloce del più potente computer della serie 360, ossia quel 360/75 che avrebbe comunque costituito uno degli elementi chiave del successo del progetto Apollo.
L'IBM avviò, immediatamente, la progettazione e costruzione del 360/91 che, come il CDC 6600, avrebbe introdotto nelle architetture dei computer il principio della *pipeline*.
La sfida sarebbe poi proseguita, negli anni successivi, con l'introduzione di due computer ancora più potenti: il CDC 7600 (1969) e il sistema IBM 360/370 195 (1970).
Con la tecnica della pipeline, che è possibile immaginare come un'estensione del principio dell'*overlap* delle istruzioni, ci si proponeva di far diventare il computer una vera e propria catena di montaggio di istruzioni che passavano via via attraverso unità funzionali specializzate.
Il termine pipeline nasceva dall'analogia con la condotta di petrolio nella quale non interessa la velocità della singola molecola bensì la quantità di petrolio che sgorga al secondo dalla condotta stessa.
Analogamente, nei computer di tipo pipeline non interessava tanto il tempo necessario a eseguire una singola istruzione quanto piuttosto la quantità di istruzioni eseguite al secondo.
In quegli stessi anni, inoltre, si cominciavano a comprendere sempre meglio i meccanismi, non solo fisici, ma anche logici sui cui si fonda un'effettiva esecuzione, in parallelo, di molteplici attività. Si immagini di avere 52 carte di uno stesso mazzo distribuite in modo casuale e di volerle mettere in ordine. Se ci fossero quattro giocatori a volersi dividere le attività secondo i colori (cuori, quadri, fiori, picche) si farebbe certamente prima che se si dovesse svolgere la stessa attività da soli. Ma fossero 52 i giocatori, tutti intorno allo stesso tavolo, a voler ordinare il mazzo si farebbe probabilmente solo una gran confusione.
Questo banale esempio può far comprendere come esista un limite al di là del quale non conviene spingere sul parallelismo degli agenti ma, piuttosto, sulla velocità del singolo.
Gene Amdahl, uno dei maggiori architetti della recente storia dei computer, enunciò all'inizio degli anni '70 un'ipotesi che venne poi denominata legge di Amdahl: un perfetto parallelismo nelle attività dei computer non è possibile perché esisteranno sempre delle sequenze ineliminabili di *software* intrinsecamente seriale [1].
Basti pensare, per esempio, a una serie di nu-

meri di Fibonacci: 1,1,2,3,5,8,13,21,... generata dal semplice algoritmo F (n + 2) = F (n + 1) + F (n). Non è possibile eseguirla in parallelo perché ogni successiva operazione dipende dalle precedenti due[1].
Esistono però anche attività che mostrano un intrinseco parallelismo. Tornando all'esempio iniziale del ripiegamento della proteina si può, per un momento, riflettere sul fatto che la conformazione finale della proteina nello spazio tridimensionale sarà quella con la minore energia potenziale. Si potrebbero, allora, calcolare indipendentemente, e in parallelo, migliaia di conformazioni indipendenti della proteina stessa. Calcolate le diverse conformazioni sarà facile trovare quella con la minima energia, che è proprio quanto è stato fatto negli ultimi anni, per molte proteine, con l'ausilio di alcuni tra i migliori supercomputer disponibili. Ma è anche intuibile che il livello di parallelismo del computer pone un limite al numero di possibili conformazioni che è possibile calcolare in parallelo che, nel caso della proteina, possono essere, anche, infinite.
Occorre, infine, ricordare che in quegli anni si attuava anche un enorme progresso nella comprensione dei meccanismi di sincronizzazione di sempre più complessi aggregati microcircuitali che comprendevano ormai memorie di transito ad alta velocità (*cache*), per istruzioni e per dati (*I-cache*, *D-cache*), gerarchie di memorie (L1, L2, L3,..), registri interni di vario tipo e, soprattutto, una crescente molteplicità di unità di caricamento (*pre-fetch* e *fetch*) di istruzioni, di decodifica (*instruction unit*) e di esecuzione (ALU, *Arithmetic Logical Unit*).
La cosiddetta prima legge di Moore cominciava a far sentire i suoi effetti. Ogni diciotto mesi raddoppiava la quantità di *transistor* e, quindi, di circuiti disponibili per *chip*. Nell'IBM, per esempio, dalla tecnologia SMS (*Standard Modular System*) dell'inizio degli anni '60 si passava alla SLT (*Solid Logic Technology*), alla ASLT (*Advanced SLT*), alla MST (*Monolithic System Technology*) via via fino alla LSI (*Large Scale Integration*) del 1978 e, infine, alla VLSI (*Very Large Scale Integration*) della metà degli anni '80.
La quantità di circuiti disponibile su di un chip passava, nel volgere di un paio di decenni, da un *transistor* a migliaia di circuiti, per la gioia dei progettisti che potevano così aggregare sempre più funzioni in uno stesso spazio senza doversi preoccupare dei ritardi di segnale.
Se gli attuali microprocessori costruiti con la tecnologia CMOS (*Complementary Metal-Oxide_Silicon*) possono oggi essere così complessi lo si deve anche al fatto che tra gli anni '60 e '70 i grandi computer costruiti con tecnologia bipolare (prevalentemente ECL, *Emitter Coupled Logic*) hanno esplorato innumerevoli possibilità di aggregare i microcircuiti per ottenere sempre più elevati livelli di parallelismo nell'esecuzione delle istruzioni.

## 3. I COMPUTER VETTORIALI

Negli anni '60, due tra i maggiori protagonisti nella progettazione dei grandi computer erano stati indiscutibilmente Gene Amdahl e Seymour Cray. Ambedue, poi, avrebbero lasciato le aziende per le quali lavoravano (rispettivamente, l'IBM e la CDC) per avviare proprie aziende.
Amdahl fondava la *Amdahl Corporation* che avrebbe costruito alcuni dei più evoluti mainframe a cavallo tra gli anni '70 e '80. Seymour Cray avviava, invece, la *Cray Research* Inc.
Nel 1975, veniva installato il primo Amdahl 470/V6 e un anno dopo presso il Los Alamos Scientific Laboratory veniva installato il primo computer vettoriale Cray 1.
Il 1976 può a ragione essere considerata la data di inizio del *supercomputing*. Prima del Cray 1, infatti, computer come il CDC 7600, che pure venivano definiti supercomputer, non differivano, realmente, dai computer più convenzionali. Ciò che li caratterizzava, a parte le più elevate prestazioni, come è stato già detto, era, essenzialmente, l'elevato livello di parallelismo interno. Occorre anche aggiungere che la CDC aveva, in realtà, realizzato un computer vettoriale, il CDC Star 100, già nel 1972; ma questa macchina non aveva avuto successo, soprattutto perché le sue prestazioni, in taluni casi, erano persino inferiori a quelle dei più potenti sistemi scalari.

[1] È anche vero che esiste da oltre un secolo una formula, quella del matematico Binet, che consente di individuare direttamente l'ennesimo numero di Fibonacci. La formula è però di difficile semplificazione per numeri di Fibonacci molto grandi per i quali si può ricorrere invece al calcolo di potenze della radice quadrata di cinque.

Con il Cray 1 nasceva un'architettura innovativa che utilizzava istruzioni vettoriali e registri vettoriali per risolvere alcune classi di problemi essenzialmente matematici.
Un processore vettoriale è un processore che può operare con una singola istruzione su di un intero vettore. Si consideri, per esempio, la seguente istruzione di somma:

C = A + B

Sia nei processori scalari sia in quelli vettoriali l'istruzione significa "aggiungere il contenuto di A al contenuto di B e mettere il risultato in C". Mentre in un processore scalare gli operandi sono numeri in un processore vettoriale l'istruzione dice al processore di sommare, a due a due, tutti gli elementi di due vettori. Un particolare registro di macchina, denominato *vector length register*, dice al processore quante somme individuali devono essere eseguite per sommare i due vettori.
Un compilatore vettoriale è, a sua volta, un compilatore che cercherà di riconoscere, per esempio, quali *loop* sequenziali possono essere trasformati in una singola operazione vettoriale. Il seguente loop può essere trasformato in una sola istruzione:

```
DO 10 I = 1, N
C(I) = A(I) + B(I)
10    CONTINUE
```

La sequenza di istruzioni del loop viene tradotta in un'istruzione che imposta a N la lunghezza del length register seguita da un'istruzione che esegue l'operazione di somma vettoriale.
L'uso delle istruzioni vettoriali consente migliori prestazioni essenzialmente per due motivi. Innanzitutto, il processore deve decodificare un minor numero di istruzioni e, quindi, il carico di lavoro della *I-unit* è fortemente ridotto e il *bandwidth* di memoria viene, di conseguenza, ridotto anch'esso. In secondo luogo, l'istruzione fornisce alle unità esecutive (le ALU vettoriali) un flusso costante di dati. Quando inizia l'esecuzione dell'istruzione vettoriale il processore sa che deve caricare coppie di operandi che sono disposte in memoria in modo regolare. Così il processore può chiamare il sottosistema di memoria per alimentare in maniera costante le ALU con le coppie successive.
Con una memoria dotata di un buon livello di parallelismo di accesso (il cosiddetto i*nterleaving*) le coppie possono essere alimentate al ritmo di una per ciclo ed essere dirette consecutivamente, come in una catena di montaggio (la pipeline, appunto) per essere via via sommate.
Occorreva, tuttavia, anche un cambio di mentalità. Programmatori abituati a ragionare in modo sequenziale dovevano acquisire una nuova mentalità e, anche se parte della conversione da software applicativo sequenziale a parallelo poteva essere eseguita in maniera automatica, restava pur sempre da svolgere un'ulteriore attività di "parallelizzazione" manuale che poteva richiedere la ristrutturazione di algoritmi che erano stati pensati in modo essenzialmente seriale.

## 4. I MASSIVE PARALLEL SYSTEM (MPP)

Nella seconda metà degli anni '80, iniziò a farsi strada una nuova classe di sistemi: i computer paralleli a memoria distribuita. Questi sistemi differivano dagli SMP (*Shared-Memory Multiprocessor*), nati all'inizio degli anni '70, poiché la memoria non veniva condivisa da un unico sistema operativo, ma ogni processore gestiva la propria memoria e il proprio software applicativo.
L'architettura degli MPP non consentiva una sincronizzazione così stretta delle attività applicative come negli SMP e richiedeva, come si può intuire, che le diverse applicazioni fossero abbastanza indipendenti l'una dall'altra. È anche evidente che nell'esecuzione di una singola applicazione, solo se la sequenza delle istruzioni di quest'ultima può essere efficacemente suddivisa (parallelizzata) in sezioni eseguibili indipendentemente sui vari processori, l'MPP può dare effettivi vantaggi.
Dal punto di vista della programmazione applicativa, i sistemi SMP erano più semplici da utilizzare. Con un'organizzazione di memoria distribuita il programmatore deve, infatti, tenere conto di chi detiene che cosa. Gli SMP, però, funzionano bene solo se i processori non sono tanti, altrimenti nascono troppi conflitti per specifiche posizioni di memoria.
I primi modelli di simili *massive parallel system* apparvero sul mercato nel 1985 in maniera abbastanza etereogenea. La *Thinking*

*Machine Corporation* (TMC) presentava la *Connection Machine-1* (CM-1), mentre la Intel annunciava l'iPSC/1 che utilizzava microprocessori Intel 80826 interconnessi da una rete *Ethernet* in un cosiddetto ipercubo.
In virtù della relativa facilità di aggregazione di simili sistemi, vennero rapidamente a costituirsi molte nuove aziende costruttrici di sistemi MPP.
Il sistema MPP più interessante di quegli anni sarebbe rimasto il CM-2 disegnato da Danny Hillis per la TMC. Costituito da 64 microprocessori interconnessi in un ipercubo, insieme a 2048 *Witek floating point unit*, il sistema poteva operare su di una singola applicazione sotto il controllo di un solo sistema di *front-end*. Il CM-2 per alcune categorie applicative riuscì persino a competere con i migliori sistemi vettoriali MP dell'epoca, come il Cray Y-MP.
Fu proprio il crescente successo di questi nuovi sistemi a determinare la necessità non solo di una rigorosa tassonomia dei vari supercomputer ma anche una classificazione relativa secondo un'opportuna metrica, quella del LINPACK, che prevede la soluzione di un cosiddetto sistema denso di equazioni lineari. Le prestazioni vengono oggi misurate in teraflop/s, ossia in migliaia di miliardi di operazioni in virgola mobile eseguite al secondo.
La lista TOP500 (Tabella 1), dei 500 più poten-

**TABELLA 1**
*La TOP500 List del Novembre 2002*

| Rank | Manufacturer Computer/Procs | $R_{max}$ $R_{peak}$ | Installation Site Country/Year |
|---|---|---|---|
| 1 | **NEC** Earth-Simulator/5120 | **35860.00** 40960.00 | **Earth Simulator Center** Japan/2002 |
| 2 | **Hewlett-Packard** ASCI Q - Alpha Server SC ES45/1.25 GHz/4096 | **7727.00** 10240.00 | **Los Alamos National Laboratory** USA/2002 |
| 3 | **Hewlett-Packard** ASCI Q - Alpha Server SC ES 45/1.25 GHz/4096 | **7727.00** 10240.00 | **Los Alamos National Laboratory** USA/2002 |
| 4 | **IBM** ASCI White, SP Power3 375 MHz/8192 | **7226.00** 12288.00 | **Lawrence Livermore National Laboratory** USA/2000 |
| 5 | **Linux NetworX** MCR Linux Cluster Xeon 2.4 GHz - Quadrics/2304 | **5694.00** 11060.00 | **Lawrence Livermore National Laboratory** USA/2002 |
| 6 | **Hewlett-Packard** Alpha Server SC ES 45/1 GHz/3016 | **4463.00** 6032.00 | **Pittsburgh Supercomputing Center** USA/2001 |
| 7 | **Hewlett-Packard** Alpha Server SC ES 45/1 GHz/2560 | **3980.00** 5120.00 | **Commissariat a L'Energie Atomique (CEA)** France/2001 |
| 8 | **HPTi** Aspen Systems, Dual Xeon 2.2 | **3337.00** 6158.00 | **Forecast Systems Laboratory - NOAA** USA/2002 |
| 9 | **IBM** pSeries 690 Turbo 1.3 GHz/1280 | **3241.00** 6656.00 | **HPCx** UK/2002 |
| 10 | **IBM** pSeries 690 Turbo 1.3 GHz/1216 | **3164.00** | **NCAR (National Center for Atmospheric Research)** - USA/2002 |
| 11 | **IBM** pSeries 690 Turbo 1.3 GHz/1184 | **3160.00** 6156.00 | **Naval Oceanographic Office (NAVOCEANO)** USA/2002 |
| 12 | **IBM** SP Power3 375 MHz 16 way/6656 | **3052.00** 9984.00 | **NERSC/LBNL** USA/2002 |
| 13 | **IBM** pSeries 690 Turbo 1.3 GHz/960 | **2560.00** 4990.00 | **ECMWF** UK/2002 |
| 14 | **IBM** pSeries 690 Turbo 1.3 GHz/960 | **2560.00** 4990.00 | **ECMWF** UK/2002 |

ti supercomputer installati nel mondo, viene pubblicata semestralmente dalle Università di Mannheim e del Tennessee e dall'U.S. *Department of Energy's National Energy Research Scientific Computing Center*.
La maggior parte dei sistemi MPP era, sostanzialmente, costituita di *cluster* più o meno integrati di microprocessori preesistenti. Ciò faceva sì che la distinzione tra sistemi convenzionali e supercomputer divenisse molto più sfumata. Un sistema convenzionale poteva, opportunamente "clusterizzato", diventare di fatto un supercomputer.
Inoltre, a livello di tecnologia di semiconduttore la crescente densità dei chip consentiva alla tecnologia CMOS di rimpiazzare, soprattutto in termini di rapporto prezzo/prestazioni e di consumi e dissipazione di calore, la tecnologia ECL.
Questo fatto, poichè la capacità industriale di produrre e integrare in un unico disegno sempre più evoluti microprocessori CMOS si restringeva via via a poche aziende, riduceva il numero di costruttori di supercomputer. Alle soglie del 2000, delle 14 maggiori aziende costruttrici di supercomputer ne rimanevano solo quattro costituite dai tre maggiori costruttori giapponesi (Fujitsu, Hitachi e NEC) e dall'IBM, cui venivano ad aggiungersi la Silicon Graphics, la Sun, la Hewlett-Packard e la Compaq. Queste ultime due si sarebbero poi consociate.

## 5. IL PROGRAMMA ASCI

Attualmente, ci sono negli USA diverse iniziative che cercano di consolidare attraverso l'*High Performance Computing* le capacità computazionali di università, laboratori di ricerca e organizzazioni militari. Il più interessante di questi programmi è, probabilmente, l'*Accelerated Strategic Computing Initiative* (ASCI) [2] il cui compito è quello di creare le capacità computazionali e di simulazione d'eccellenza necessarie per mantenere la sicurezza l'affidabilità e le prestazioni dello *stockpile* nucleare USA e ridurre il rischio nucleare. Nell'ambito di questo programma sono stati realizzati diversi sistemi.
Nella TOPLIST del Novembre 2002 al secondo, terzo e quarto posto si trovano proprio tre sistemi ASCI. Nell'ordine, al Los Alamos National Laboratory sono operativi dal 2002 due ASCI Q – Alpha Server SC della HP-Compaq costituiti da 4096 processori ES45 da 1.25GHz con una capacità di calcolo di 7, 727 Tflop/s. Al Lawrence Livermore Laboratory è operativo, dal 2000, l'IBM ASCI White costituito di 8192 processori SP POWER 3 da 375 MHz per una potenza complessiva di 7,226 Tflop/s.
L'ASCI Red, prodotto dalla Intel e installato presso il Sandia National Laboratory, è costituito da 9472 microprocessori Intel Pentium Xeon ed è stato, con 2.1 Tflop/s, il primo sistema a superare la barriera di un teraflop/s e a porsi, per un certo periodo, al vertice della TOP500.
L'ASCI Blue Mountain installato al Los Alamos Laboratory è stato costruito dalla SGI ed è costituito da 6144 processori e reaggiunge 1.6 Tflop/s.
Ci sono poi altri sistemi molto potenti. Il governo giapponese ha sovvenzionato lo sviluppo di un sistema per simulare e prevedere l'ambiente globale. Il NEC Earth Simulator è costituito di 640 nodi per un totale di 5120 unità di elaborazione, occupa uno spazio di quattro campi da tennis e utilizza 2800 km di cavo per le interconnessioni. Raggiunge una capacità di calcolo di 40 Tflop/s e appare come il supercomputer più potente secondo la classifica TOP500 di Novembre 2002.
Nell'ambito del programma ASCI, il 19 Novembre 2002 è stato annunciato che l'IBM avrebbe avviato la costruzione di due nuovi supercomputer, ASCI Purple e Blue Gene/L, che insieme dovrebbero fornire 500 Tflop/s, ossia una volta e mezzo la potenza di tutti i supercomputer della più recente TOP500. ASCI Purple, con una capacità di calcolo di 100 Tflop/s, sarà costituito da 12608 microprocessori IBM POWER5 e consumerà 4 MW (l'equivalente dell'energia eletrica necessaria per circa 5000 abitazioni). I microprocessori saranno contenuti in 196 computer, con un bandwidth di memoria di 156000 Gbyte, e interconnessi da un *data-highway* con un bandwidth di 12500 Gbyte. ASCI Purple conterrà anche 50 Tbyte di memoria e 50 petabyte di memoria su disco, che è all'incirca l'equivalente di un miliardo di libri di media lunghezza.
ASCI Purple, che verrà installato entro il 2004 presso il Lawrence Livermore National Laboratory, dov'è già installato ASCI White, servirà come il principale supercomputer del-

l'U.S. Department of Energy nell'ambito del *National Security Administration* (NNSA) dove sarà utilizzato nel *Stockpile Stewardship Program* per simulare l'invecchiamento e l'operatività delle testate nucleari americane senza dover ricorrere a test sotterranei.

## 6. LA RESURREZIONE DEL VETTORIALE

Pur producendo sistemi MPP di successo, come i TRD/T3E la Cray ha continuato a sviluppare supercomputer vettoriali anche dopo l'accordo di cooperazione con la SGI. In particolare, la Cray nel 1998 ha costruito il supercomputer SV1, poi perfezionato nel 2000.
La Cray ha poi ritenuto possibile utilizzare il meglio delle due architetture, quella vettoriale e quella MPP, per costruire un nuovo supercomputer, denominato durante la fase di progetto SV2.
Nel novembre 2002, finalmente la Cray ha annunciato il Cray X1, già disponibile, che, in sostanza, riunisce il meglio dei sistemi vettoriali SV1 e T90 e dei sistemi MPP T3D/E.
Nel sistema SV1, la Cray aveva introdotto i cosiddetti processori *multistreaming* (MSP) per sfruttare il vantaggio di avere, invece di un'unica profonda pipeline vettoriale, molteplici pipeline più brevi. Da un punto di vista puramente intuitivo, è possibile immaginare che molteplici piccole pipeline possano essere utilizzate per gestire in parallelo, più efficientemente, loop abbastanza brevi, ma frequenti nell'ambito di un programma.
Nel Cray X1 il blocco costruttivo fondamentale è un processore di tipo MSP costituito di quattro unità scalari ciascuna dotata di un paio di *vector pipe*. Ogni unità scalare può eseguire, non in sequenza su una delle due unità esecutive, due istruzioni per ciclo. Le quattro unità scalari operano a 400 MHz e forniscono una capacità di calcolo totale di 3.2 Gflop/s.
Come accennato in precedenza, i compilatori per i sistemi vettoriali sostanzialmente "aprono" i loop in sequenze di istruzioni ottimizzate per l'*hardware* vettoriale. Per loop più grandi la sequenza di istruzioni di un unico loop può essere distribuita su tutte le 8 pipeline di un processore.
Le 8 pipeline vettoriali operando a 800 MHz forniscono una capacità di calcolo complessiva di 12.8 Gflop/s per operazioni a 64 bit.
Poichè nel disegno complessivo del Cray X1 sono previsti fino a 4000 MSP, il Cray X1 stesso può fornire una capacità di calcolo di 52.4 Tflop/s.

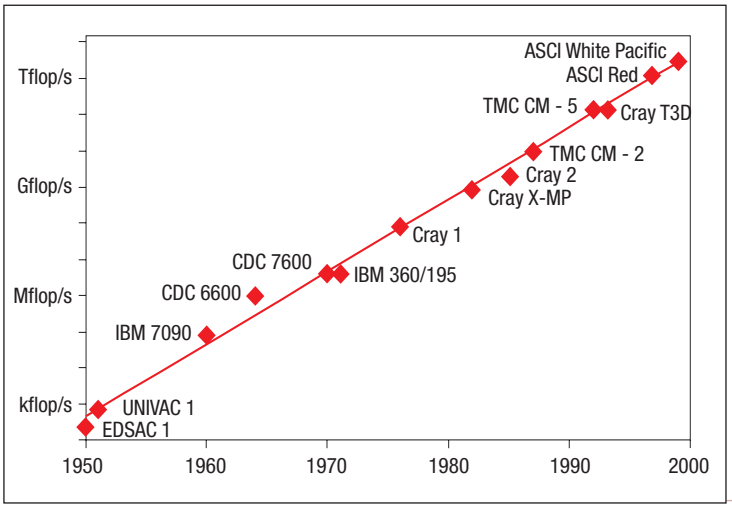
## 7. LE PROSPETTIVE DEI SUPERCOMPUTER: VERSO IL PETASCALE COMPUTING

Negli Stati Uniti i supercomputer sono stati sempre considerati un elemento strategico fondamentale per mantenere la *leadership* non solo tecnologica.
Il rapporto PITAC del 1999 (*Report from the President's Information Technology Advisory Committee*) raccomanda un programma federale di ricerca che consenta di " ... *ottenere prestazioni da petaflop/s su applicazioni reali entro il 2010 attraverso un opportuno bilanciamento di strategie hardware e software*" [5]. Il rapporto identifica molte applicazioni che potranno trarre considerevoli vantaggi dal petascale computing:

- gestione degli ordigni nucleari;
- crittologia;
- modellazione ambientale e climatica;
- ricostruzione tridimensionale delle proteine;
- previsione di uragani;
- disegno avanzato di aerei;
- nanotecnologie molecolari;
- gestione in tempo reale di immagini per la medicina.

Se si traccia su di una scala logaritmica una curva che rappresenti l'evoluzione dei supercomputer si può vedere come la prima legge di Moore sia stata in gran parte rispettata (Figura 2).

**FIGURA 2**
*L'evoluzione dei supercomputer*



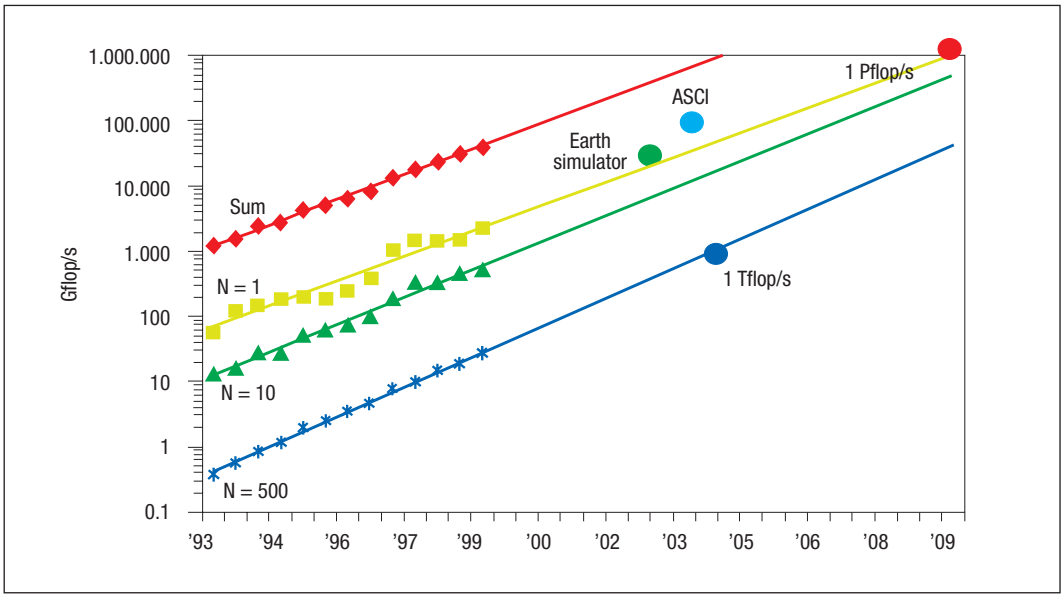



**FIGURA 3**
*Verso il petascale computing*

Si potrebbe fare un esercizio anche più interessante esaminando l'evoluzione delle prestazioni nella TOP500, prendendo di volta in volta l'evoluzione del sistema più potente, del meno potente (ossia, il sistema classificato al posto 500) e, quindi, del sistema classificato al decimo posto (Figura 3).
La curva dei supercomputer alla posizione 500 mostra un miglioramento annuale di un fattore medio di 1.8. Le altre curve mostrano un miglioramento intorno a un valore annuale pari a 2. Estrapolando si può vedere che i primi sistemi da 100 Tflop/s dovrebbero essere disponibili intorno al 2005. Nel 2005, nella TOP500 non ci dovrebbe essere più alcun supercomputer con potenza di calcolo minore di un teraflop/s.
Estrapolando ulteriormente ci si può spingere fino a un petaflop/s e vedere che questa potenza dovrebbe essere raggiunta intorno al 2009. In effetti, come si vedrà più avanti, ci sono due sistemi, l'IBM Blue Gene e l'HTMT che promettono di fornire una simile potenza intorno a quella data.
La CRAY sostiene che il sistema Cray X1 è un primo passo verso il petascale computing che potrebbe essere realizzato, attraverso un'opportuna evoluzione tecnologica e di disegno, entro il 2010.
Il petascale computing sembrerebbe, quindi, raggiungibile in tempi abbastanza brevi, ma occorrono ancora notevoli progressi tecnologici e anche progetti alternativi perché non è detto che lungo la strada qualche progetto non si dimostri, di fatto, irrealizzabile.
Si ritiene, oggi, che vi siano almeno quattro possibili alternative tecnologiche: l'evoluzione della tecnologia convenzionale, il disegno cosiddetto *processor in memory* (PIM), le tecnologie superconduttive e, infine, il *Grid Computing*.

## 7.1. Le tecnologie convenzionali

Il futuro del supercalcolo sarà condizionato sia dall'evoluzione della tecnologia circuitale sia da quella del disegno di sistema che, a sua volta, non potrà non essere influenzato dalla tecnologia disponibile.
Allo stato attuale, i microprocessori più densi hanno quasi 200 milioni di transistor: nel volgere di pochi anni, saranno disponibili chip con oltre un miliardo di transistor. La *Semiconductor Industry Association* (SIA) nel suo ultimo documento [6] ha delineato la possibile evoluzione della tecnologia di semiconduttore fino al 2014. Le sue proiezioni indicano, in estrema sintesi, che nel 2008 le memorie dinamiche (DRAM) avranno 5 miliardi di bit per cmq, mentre le memorie statiche (SRAM) avranno 500 milioni di transistor per cmq e, infine, i chip di circuitazione logica ad alte prestazioni avranno 350 milioni di transistor per cmq. Queste densità verranno quasi triplicate entro il 2011, mentre le frequenze per lo

stesso periodo arriveranno quasi a una decina di GigaHertz contro il singolo GigaHertz attuale.
Queste proiezioni sono impressionanti perché è presumibile che le dimensioni dei chip saranno tra 400 e 800 mmq, e perciò, entro pochi anni, saranno disponibili chip con miliardi di transistor. Ciò avrà profonde conseguenze sul disegno interno dei computer e quindi sulle loro prestazioni e sulle loro possibilità applicative
Si immagini, infatti, di poter disporre di un chip di 400 $mm^2$ con una densità di transistor quale quella prevista da SIA.
Un processore, di disegno sufficientemente evoluto, occuperebbe 5 $mm^2$ e, quindi, 16 processori da 6 GHz impegnerebbero 80 $mm^2$, lasciando perciò 320 $mm^2$ disponibili per le intercomunicazioni tra i processori e per la memoria. Utilizzando 50 $mm^2$ per le comunicazioni restano 270 $mm^2$ per la memoria che, alle densità previste da SIA, fornirebbe 64 Mbyt di SRAM oppure 11 Gbyt di DRAM. Nel 2011 i 16 processori lavorerebbero a 10 GHz e la memoria DRAM diventerebbe di 35 Gbyt.
Raccogliendo su di una superficie (*board*), di $20 \times 20$ cm, 64 chip si può arrivare a interconnettere oltre 1000 processori per ottenere una capacità complessiva di 20 Tflop/s, che è quella oggi fornita dai massimi supercomputer disponibili.
Simili strutture, che sembrano quasi ricordare quella di un alveare, prendono il nome di sistemi cellulari e già oggi, con la tecnologia attualmente disponibile, sono in fase di progettazione. Il supercomputer IBM Blue Gene [3] il cui progetto è stato annunciato nel dicembre del 1999, ha già queste caratteristiche.
Come accennato in precedenza, nel novembre del 2002, l'IBM ha annunciato l'avviamento di un altro progetto da completarsi in tempi più brevi rispetto all'originario Blue/Gene: Blue Gene/L.
Blue Gene/L è costituito da 64 *cabinet*, ciascuno dei quali contiene 128 *board*, ognuna delle quali alloggia 8 chip, su ciascuno dei quali prendono posto 2 processori. Blue Gene/L è costituito quindi da circa 130.000 processori di una potenza di circa 2.8 Gflop/s, operanti in Linux, per una potenza complessiva che può arrivare a 360 Tflop/s. L'obiettivo di Blue Gene/L, oltre alla simulazione del folding delle proteine, è quello di consentire la modellazione tridimensionale delle stelle per verificare perché le stelle binarie, con un elevato rapporto di masse, diventino instabili e si fondano. Sarà anche possibile utilizzare tecniche avanzate di modellazione della meccanica quantistica per studiare con grande accuratezza le proprietà della materia. Un'area di particolare interesse sarà la possibilità di studiare come riparare sequenze di DNA danneggiate dalla radiazione. Infine, la possibilità di simulare con grande precisione la propagazione di onde sismiche e acustiche potrebbe essere utilizzata per la previsione di terremoti, per le prospezioni petrolifere, per creare sempre più accurate immagini a uso medico.

### 7.2. Il Processing in memory (PIM)

Durante l'evoluzione dell'High Performance Computing, precedentemente descritta, è emerso un fattore tecnologico che, poco alla volta, è diventato uno dei maggiori vincoli alla crescita dell'effettiva capacità di calcolo dei supercomputer.
Questo vincolo è rappresentato dal progressivo sbilanciamento tra la tecnologia dei processori e quella della memoria dei supercomputer: la velocità di esecuzione delle istruzioni è aumentata molto più rapidamente del tempo di accesso alla memoria centrale.
Il *gap* tra il miglioramento nell'esecuzione delle istruzioni e nell'accesso alla memoria sta aumentando di circa il 50% annuo (Figura 4).
Già negli anni '60-'70, non era facile mantenere caricata correttamente la pipeline in ogni momento ed erano state erano progettate molteplici tecniche, come il pre-fetch delle istruzioni, e come i *buffer* di transito ad alta velocità (le cosiddette, cache) che erano successivamente stati distinti in I-cache e D-cache (per le istruzioni e i dati). Si era anche esaminato il problema della cosiddetta "località dei dati", ossia la distanza media che c'era in memoria tra le istruzioni e i dati su cui queste dovevano operare. Tale località varia a seconda della natura delle

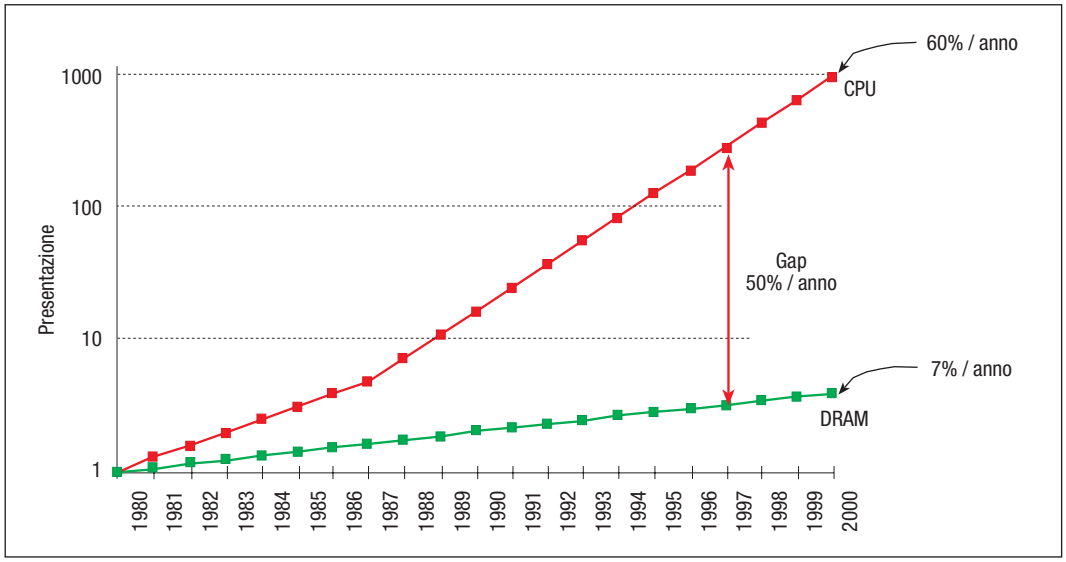



**FIGURA 4**
*Il latency gap delle DRAM*

applicazioni e può richiedere approcci sostanzialmente diversi.
Tutte queste tecniche avevano portato alla costruzione di complesse gerarchie di memoria, soprattutto nei computer più potenti e, in particolare, nei supercomputer.
Si è visto nella sezione 6.1, che la densità fotolitografia aumenterà nei prossimi anni in misura tale da consentire di alloggiare nello stesso chip memorie SRAM ovvero DRAM e molteplici microprocessori.
Numerose tecniche sono, oggi, allo studio per utilizzare al meglio le possibilità che la tecnologia sarà, quindi, in grado di fornire. Già da alcuni anni, si stanno evolvendo nuove tecniche di memoria come la PIM [8] o la IRAM (*Intelligent RAM*) che prevedono di integrare sullo steso chip logica e memoria.

## 7.3. Le tecnologie superconduttive

I petaflop/s computer sembrano, comunque, destinati a consumare enormi quantità di corrente. Si è già visto che un sistema da un decimo di petaflop/s, come ASCI Purple, richiederà almeno 4 MW. Un sistema da un petaflop/s costruito con tecnologia convenzionale potrebbe arrivare a dover dissipare 100 MW di potenza e richiederebbe un impianto di raffreddamento grande come un edificio di tre piani e al buio apparirebbe dotato di una certa luminosità.
L'idea dei sistemi cellulari, come Blue Gene, è allora quella di utilizzare un numero molto grande, fino a decine di migliaia di chip con un disegno sufficientemente semplice da non richiedere un'elevata potenza elettrica individuale.
Sembra, però, possibile anche un'altra strada che è quella di utilizzare una tecnologia superconduttiva denominata *Rapid Single-Flux-Quantum Logic* (RSFQ), inventata da Konstantin Likharev della *State University* di New York [7].
L'elemento chiave di questa tecnologia sono microscopici anelli che possono quasi istantaneamente commutare dallo stato magnetizzato a quello non-magnetizzato (o e 1) senza, praticamente, consumare potenza. Gli anelli possono rimpiazzare gli attuali transistor e consentire di costruire microprocessori fino a mille volte più rapidi di quelli attuali.
La tecnologia RSFQ viene oggi utilizzata nell'ambito di un ambizioso progetto denominato *Hybrid Technology Multi-Threaded* (HTMT) supportato dal *Defense Advanced Research Projects Agency* (DARPA), dalla *National Security Agency* (NSA), e dalla *National Aeronautics and Space Administration* (NASA). Del progetto fanno parte prestigiosi ricercatori il cui obiettivo è quello di realizzare il petascale computing entro il 2010.

## 7.4. Il Grid Computing

La potenza di calcolo complessiva del Web è certamente enorme ma è facilmente intuibi-

le, anche per quanto è stato detto finora, che a causa della *latency* di memoria e dell'eterogeneità del bandwidth tale potenza non possa facilmente essere utilizzata per eseguire, in modo unitario, applicazioni che richiedano grande capacità di calcolo.
Esistono, comunque, classi di applicazioni, le cosiddette *loosely coupled application*, che, pur richiedendo elevata capacità di calcolo, non hanno vincoli di esecuzione particolarmente stringenti e che, inoltre, si prestano a essere suddivise per essere eseguite su molteplici sistemi.
In questo caso, la definizione stessa di applicazione tende a diventare abbastanza indefinita perché, una volta che differenti sequenze di istruzioni vengano eseguite quasi indipendentemente per risolvere un problema, non è chiaro se esse possano complessivamente venire a costituire un'unica applicazione in senso classico.
Un programma di suddivisione del software applicativo sui sistemi del Web è, comunque, in atto da parte di organizzazioni e costruttori diversi nell'ambito di una strategia che prende complessivamente il nome di Grid Computing.
Secondo questo modello, non importa dove specifiche sequenze di istruzioni vengano eseguite, quanto piuttosto che la loro sincronizzazione complessiva, e quindi il raggiungimento del risultato, vengano raggiunti.
Il termine *grid* deriva dall'analogia con infrastrutture già esistenti, come quella elettrica o quella dell'acqua. Nell'aprire un rubinetto o nell'accendere una lampada non ci si domanda da dove provenga l'acqua o l'elettricità: l'importante è che queste siano rese disponibili. Analogamente, nel Grid Computing, quando una sequenza di istruzioni deve essere eseguita, non importa quale processore se ne farà carico: il sistema complessivo dovrà individuarlo nell'ambito di quelli disponibili in rete e ivi verranno eseguite le istruzioni.
Con una simile strategia è intuibile che il modello di sviluppo applicativo venga ulteriormente modificato. Già nel passaggio dall'informatica centralizzata degli anni'60-'70 a quella sempre più distribuita (soprattutto *client-server*) degli anni '80-'90, si era assistito all'evolversi delle tecniche di modularizzazione del software applicativo.
Dalle semplici chiamate *call-return* nell'ambito di un solo sistema si era passati ad *application programming interface* che dovevano coordinare logicamente (e fisicamente) l'esecuzione di sequenze di istruzioni che potevano risiedere su sistemi diversi e anche ubicati remotamente. Poiché i vari sistemi potevano avere anche un'architettura diversa, oltre ai diversi sistemi operativi attivi sui rispettivi computer, diventava necessaria l'aggiunta di una specie di "colla" software che consentisse di collegare tali sistemi in una visione logica complessiva, ma in maniera il più indolore possibile, soprattutto, per i programmatori applicativi. Tale colla prese il nome di *middleware*, ossia il software che si poneva in mezzo.
In una strategia Grid Computing il middleware, che qualcuno già scherzosamente chiama *griddleware*, diventerà elemento di fondamentale importanza e il ruolo di organizzazioni come CORBA, Legion, Globus [10, 11] è proprio quella di ottimizzare e standardizzare quanto più possibile tale middleware.

## 8. ALGORITMI E MODELLI APPLICATIVI

Quanto detto fin qui, ha soprattutto evidenziato l'evoluzione della tecnologia di base, del disegno delle macchine e in parte dei sistemi operativi. Resta, però, da considerare l'importantissimo ruolo che svolgerà il software applicativo per conseguire realmente il petascale computing. È necessario, innanzitutto, considerare l'importanza degli algoritmi. Quest'ultimo è un aspetto spesso sottovalutato. Infatti, per utilizzare al meglio un computer che esegua contemporaneamente su molteplici complessi circuitali svariati gruppi di istruzioni occorre risolvere due problemi. È necessario, innanzitutto, sviluppare in software un algoritmo che si presti a essere suddiviso in più parti, ossia in diverse sequenze di istruzioni, da eseguire in parallelo. In secondo luogo, occorre disporre di un linguaggio, quello che in gergo informatico si chiama un *compilatore*, che sappia ottimizzare nelle giuste sequenze da

distribuire in parallelo le istruzioni scritte dal programmatore.
La progettazione di un algoritmo efficiente è, spesso, più efficace di un hardware sofisticato. Basti pensare a un algoritmo tra i più banali: quello della moltiplicazione. Si potrebbe, ad esempio, sommare 57 volte il numero 23 su se stesso, ma questo procedimento, anche per un uomo, sarebbe molto più lungo del notissimo algoritmo che consente di moltiplicare i due numeri tra loro con pochi passaggi aritmetici. È intuibile che si possano sviluppare algoritmi molto raffinati ed è questo un fondamentale terreno di ricerca nel disegno dei supercomputer.
Purtroppo, non è facile trovare la soluzione giusta. Con la disponibilità del calcolo parallelo, il ricercatore che debba sviluppare una nuova applicazione potrebbe essere invogliato a cercare fin dall'inizio algoritmi intrinsecamente parallelizzabili. Più difficile è, evidentemente, il compito di chi debba riadattare software applicativo già esistente alle nuove possibilità tecnologiche.
Il paragone con il gioco degli scacchi può essere utile, non solo per comprendere l'accanimento con il quale l'industria del computer ha cercato, negli ultimi anni, di sconfiggere di volta in volta il campione del mondo in carica, ma anche per capire meglio il tema della complessità algoritmica.
L'*input* per un programma che gioca a scacchi è la specifica disposizione dei pezzi, ossia la configurazione di una scacchiera, che in un'analogia biologica può corrispondere alla sequenza di aminoacidi di una proteina. L'algoritmo scacchistico cerca, in una libreria elettronica di partite giocate dai grandi maestri, se vi sia una configurazione quasi identica. Se questa viene trovata, la mossa successiva sarà uguale a quella effettuata nella partita omologa individuata nella libreria.
In modo analogo, il modello tridimensionale della nuova proteina può essere costruito usando come stampo la struttura della proteina di riferimento.
Fino al 30% delle proteine relative a genomi, pienamente sequenziali, mostra una rassomiglianza alla sequenza genomica di proteine la cui struttura è già nota, cosicché anche per le prime può esserne costruito un buon modello tridimensionale.
Ma se non c'è una sufficiente rassomiglianza ossia se, nell'analogia, la configurazione della scacchiera non è immediatamente riconoscibile tra quelle delle partite memorizzate elettronicamente, che fare?
Nel gioco degli scacchi occorre un algoritmo di riconoscimento delle possibili rassomiglianze tra le configurazioni dei pezzi molto più raffinato che nella tecnica cosiddetta dell'omologia. In quest'ultima, come già detto, la configurazione della partita in gioco e quella della partita memorizzata sono pressoché identiche. Ma nella tecnica del riconoscimento di forma delle proteine (*fold recognition*) le rassomiglianze, riconosciute come tali, devono essere catturate in maniera più sottile.
A questo punto, può veramente entrare in gioco la potenza di calcolo del computer che, come si è visto, sembra poter ancora crescere in maniera considerevole. Le prospettive tecnologiche sono, quindi, molto incoraggianti e indicano che la tecnologia di semiconduttore, il disegno dei sistemi cellulari e la progettazione di sofisticati algoritmi saranno sinergicamente in grado di fornire nel prossimo decennio una crescita nella capacità di elaborazione che va persino al di là della cosiddetta prima legge di Moore. Si potranno, allora, affrontare con maggior ottimismo le sfide dei prossimi anni e si schiuderà, probabilmente, un'epoca nella quale sarà possibile migliorare in maniera sorprendente la qualità della vita. Forse si potranno predire terremoti e maremoti, si potrà sapere in anticipo come saranno le stagioni, comprendere l'andamento dei mercati finanziari, progettare nuovi materiali, coltivare meglio i terreni, ma, soprattutto, venire a capo di alcuni flagelli, come le malattie degenerative, che rappresentano, tuttora, un incubo per la maggior parte dell'umanità.

## Bibliografia


[1] Amdahl's law: www.sc.ameslab.gov/publications/AmdahlsLaw/Amdahls.pd

[2] Advanced Simulation and Computing at Lawrence Livermore National Laboratory: www.llnl.gov./asci

[3] Blue Gene: A vision for protein science using a petaflop supercomputer: http://www.research.ibm.com/journal/sj/402/allen.html

[4] Hybrid Technology Multi-threaded (HTMT) Architecture Project: http://ess.jpl.nasa.gov/subpages/reports/ooreport/lc_00.htm

[5] Information Technology Research: Investing in Our Future, President's Information Technology Advisory Committee Report to the President: www.ccic.gov/ac/report

[6] International Technology Roadmap for Semiconductors, SIA, Austin, Texas, 1999: http://public.itrs.net/files/1999_SIA_Roadmap/Home.htm

[7] Likharev K: *Rapid Single Flux Quantum (RSFQ) Logic*. In Encyclopedia of Material Science and TechnologyRSFQL: http://gamayun.physics.sunysb.edu/RSFQ/publications.html

[8] Mitros P: Computing in Memory: http://www.ai.mit.edu/projects/aries/course/notes/pim.html

[9] Overview of HTMT: http://www.cacr.caltech.edu/powr98/presentations/htmt/sld001.htm

[10] Foster I, Kesselman C: The Globus Project: A Status Report: http://citeseer.nj.nec.com/foster96globu.html

[11] Grimshaw Andrew S, Wulf Wm A: The Legion vision of a worldwide virtual computer: http://portal.acm.org/citation.cfm?id=242867&coll=portal&dl=ACM&ret=1

[12] van der Steen AJ, Dongarra JJ: Overview of recent supercomputers. http://www.phys.uu.nl/~steen/web02/overview02.html

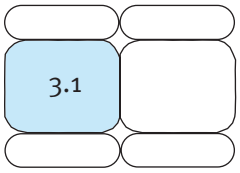
[13] TOP500 SUPERCOMPUTER SITES: www.TOP500.org

ERNESTO HOFMANN, laureato in fisica presso l'Università di Roma. È entrato in IBM nel 1968 nel Servizio di Calcolo Scientifico.Nel 1973 è diventato manager del Servizio di Supporto Tecnico del Centro di Calcolo dell'IBM di Roma. Dal 1984 è Senior Consultant. È autore di molteplici pubblicazioni sia di carattere tecnico sia divulgative, nonché di svariati articoli e interviste. Dal 2001 collabora con l'Università Bocconi nell'ambito di un progetto comune Bocconi-IBM.
Ernesto_Hofmann@it.ibm.com

# QUANTUM COMPUTING: SOGNO TEORICO O REALTÀ IMMINENTE?

**Emanuele Angeleri**

Una pregnante osservazione del premio Nobel R. Feynman, risalente a una quarantina di anni fa, ha aperto la strada a interessanti studi e a stupefacenti applicazioni della fisica quantistica nel campo delle macchine per il calcolo automatico. Si è aperto, a seguito di ciò, un nuovo settore informatico conosciuto con il nome di *Quantum Computing*. Nell'articolo qui presentato vengono forniti i primi elementi per comprendere il funzionamento di questi nuovi dispositivi dai quali si attendono, in futuro, innovative realizzazioni sperimentali.




## 1. INTRODUZIONE

Un'antica e interessante intuizione, nata a quanto pare in India e rimbalzata in Occidente attraverso il filosofo greco Democrito, ha trovato, agli inizi del secolo XX, una ragionevole conferma nelle evidenze sperimentali a cui la scienza moderna sottopone, a scopo di verifica, ogni affermazione circa la realtà fisica. La felice intuizione, secondo quanto argomenta Democrito, ha una sua base razionale: il pane, la carne e i diversi cibi di cui si nutre l'uomo vengono trasformati dal suo organismo in capelli, unghie e sangue. È pertanto assai verisimile che gli oggetti che si osservano sono prodotti da componenti elementari, gli atomi appunto, che aggregandosi diversamente producono le varie apparenze con cui si manifesta ciò che si indica, generalmente, con il termine di "realtà".
Democrito si pone a questo proposito una domanda molto intrigante: è possibile estendere agli oggetti materiali il procedimento di suddivisione all'infinito che si immagina di poter eseguire per gli enti geometrici? Se è possibile pensare di suddividere un segmento in un processo senza termine, fino a pervenire a un oggetto senza dimensioni, infinitamente piccolo, il punto geometrico[1], sarà anche possibile pensare di suddividere un oggetto materiale con un analogo processo infinito? Democrito risponde negativamente a questa domanda. L'ente geometrico astratto, il "segmento", permette la conseguente definizione di un ente astratto senza dimen-

1 L'antinomia che il problema della infinita suddivisibilità geometrica inevitabilmente propone non è sconosciuta a Democrito (se gli infiniti punti di cui si compone un segmento hanno dimensione finita, la loro somma non può che fornire un segmento di lunghezza infinita – ciò che non è vero; se poi hanno dimensione nulla, la loro somma non può che dare un segmento di lunghezza nulla – ciò che ancora non è vero). Per Democrito la suddivisone di enti geometrici, essendo riferibili a entità astratte, malgrado la difficoltà menzionata, è perseguibile all'infinito e può essere legittimamente usata per il calcolo di lunghezze, aree, volumi ecc., ma la stessa cosa non è lecita per gli oggetti materiali.

sioni, il "punto geometrico", ma per un ente fisico il processo di suddivisione avrà un termine concreto quando si raggiungeranno i "granuli" (o semi) di cui ogni oggetto materiale è costituito, o anche i suoi "atomi" (= *indivisibili*), secondo l'apposito nome che venne per essi appositamente coniato.
La scienza moderna, come si è detto, ha verificato questa ipotesi e ha costruito su di essa tutta la *teoria atomica*. Ma l'operazione non è stata indolore. La scoperta e lo studio dell'universo microscopico ha aperto, in campo fisico, un'immensa voragine teorica – il termine, in questo caso, non è affatto esagerato – in quanto le evidenze sperimentali hanno costretto a concludere che i fenomeni microscopici obbediscono a leggi nuove e diverse da quelle della fisica classica, con le quali per lungo tempo l'umanità ha creduto di poter spiegare tutta la realtà fisica. Infatti, da un lato si collocano i fenomeni macroscopici classici per i quali il *principio di causalità in senso forte deterministico* è strettamente applicabile, dall'altro i fenomeni microscopici quantistici per i quali il *principio di causalità* della fisica classica viene a cadere.

## 1.1. Principi e paradossi della fisica quantistica

Nel riquadro, si è voluto evidenziare una delle più eclatanti evidenze in base alle quali si è prodotta la menzionata frattura. Il punto focale fra le due posizioni è tutto racchiuso in tre sole parole atomi più leggeri.
**Laplace**, estendendo i brillanti risultati ottenuti dalla fisica classica, sostiene che la causalità deterministica vale anche per gli "atomi più leggeri". **Heisenberg**, sulla base delle osservazioni effettuate nella realtà microscopica e delle difficoltà apparentemente insolubili in termini classici ad esse connesse, sostiene che detto principio debba essere fatto cadere, per essere sostituito dal principio di causalità statistica.
Su questo contrasto si apriva un nuovo meraviglioso capitolo della fisica che ebbe l'effetto di imbarazzare e stupire i contemporanei e di continuare a imbarazzare e stupire anche coloro che sono venuti dopo, noi compresi. Infatti, come conseguenza del **principio di indeterminazione**, scoperto ed enunciato da W. Heisenberg, derivano una serie di circostanze che non possono non lasciare perplessi.

**Il punto di vista classico**
"Una intelligenza che, per un dato istante conoscesse tutte le forze da cui è animata la natura e la situazione reciproca degli esseri che la compongono, se per di più fosse abbastanza profonda per sottoporre questi dati all'analisi, potrebbe abbracciare nella stessa formula i moti dei più grandi corpi dell'universo e degli atomi più leggeri, nulla sarebbe incerto per essa e l'avvenire come il passato, sarebbe presente a suoi occhi".

**Laplace** PS: *da Essai philosophique sur les probabilités*

**Il punto di vista quantistico**
"Nella proposizione s*e conosciamo con precisione il presente, possiamo calcolare il futuro* non è falsa la conseguenza, bensì la premessa. Infatti siccome non possiamo partire da alcuna determinazione simultanea di impulso e posizione, la nullità del principio di causalità classica è inappellabile".

**Heisenberg** W: *da Das Naturbild der heutigen Physik*

**Il principio di indeterminazione** è il prodotto delle imprecisioni che risultano nella determinazione simultanea della coordinata di posizione "$q$" e della corrispondente cooordinata coniugata di impulso "$p$"(*) è almeno eguale alla costante $h/2\pi$.
In formula:

$$\Delta q \, \Delta p \geq h/2\pi$$

dove "$h$", costante di Planck, vale $6.63 \times 10^{-34}$ Joule $\times$ secondi.
Data l'estrema piccolezza della costante di Planck questo effetto diventa significativo solo per oggetti di dimensioni microscopiche.

(*) Si ricorda che l'impulso è dato dal prodotto m $\underline{v}$, essendo m la massa e $\underline{v}$ la velocità dell'oggetto materiale considerato

### 1.1.1. Il principio di indeterminazione

Il principio di indeterminazione di Heisenberg ha, infatti, delle conseguenze di una portata vastissima, tali da sconvolgere le consolidate certezze della più ovvia esperienza quotidiana.
Per riferirsi a una circostanza banale, all'idea che gli elettroni, costituenti ultimi della materia, siano sferette materiali di dimensioni piccolissime si deve aggiungere la constatazione che, pur trattandosi di sferette, hanno un comportamento, quanto meno, molto capriccioso. Ammesso di poter superare le difficoltà di maneggiare oggetti di dimensioni così piccole, nessuno, tanto per fare un esempio ripreso dall'esperienza quotidiana, potrebbe mai, in base al principio di indeterminazione, pensare di giocare a bocce o a biliardo con simili oggetti. Infatti, quando si gioca, ciascuno riconosce le proprie bocce, semplicemente

seguendole con gli occhi durante il loro moto. O meglio, durante il gioco, per tener sotto controllo la situazione, i giocatori sono costretti in continuazione a eseguire "a occhio" misure di posizione e di velocità. E poiché, nel caso delle microbocce, l'individuazione esatta della posizione esclude un'altrettanto esatta individuazione della velocità (e viceversa), è impossibile che con questo metodo giocatori o arbitro possano garantire con certezza l'appartenenza delle varie "microbocce" in gioco. Qualcuno potrebbe proporre, in analogia a ciò che si può fare con le normali bocce macroscopiche, di colorare i vari elettroni per indicarne l'appartenenza. Ma, purtroppo, anche questa possibilità è esclusa. Esiste, nella fisica quantistica, accanto al principio di indeterminazione un altro principio, noto come **principio di indiscernibilità**, in base al quale le particelle microscopiche – e, quindi, anche gli elettroni con cui si è ipotizzato di giocare - risultano *indiscernibili*: ovvero le particelle elementari sono tutte identiche, tanto da non poter essere distinguibili le une dalle altre né con tecniche di colorazione, né con altri sistemi. Come diceva sorridendo il vecchio professore di chi scrive di fisica teorica, "non è possibile tingere di verde un elettrone!"

C'è già, dunque, molta materia per una riflessione. Porzioni piccolissime di materia, che costituisce un solido e sicuro riferimento nell'essere umano al punto che è impossibile, per quest'ultimo, pensare alla realtà fisica se non in termini materiali[2], perdono ogni connotazione di localizzazione e distinzione provocando in esso un senso di vertigine e smarrimento.

Ma allora, sorge spontanea la domanda, che cosa è veramente ciò che viene indicato con l'espressione "realtà fisica"?

Si potrebbe agevolmente mostrare, ma

Se si parte dalla osservazione incontrovertibile che l'unico modo per distinguere due particelle è l'esame delle loro proprietà fisiche, si è necessariamente portati a concludere che due particelle saranno identiche e indistinguibili se le loro proprietà fisiche sono *identicamente* le medesime, in tal caso, infatti, viene a cadere ogni mezzo fisico per distinguerle. In particolare, se si considera una particella come un pacchetto d'onde – principio di complementarietà - è abbastanza agevole trarre la conclusione, visto che le onde non hanno una individualità propria, che, dopo che due particelle A e B hanno interagito, non sarà più possibile distinguere la particella A dalla particella B.
Pertanto, se si indica che la particella 1 si trova nello stato quantico "*m*" con funzione d'onda $\psi_m(1)$ e che la particella 2 si trova nello stato quantico "*n*" con funzione d'onda $\psi_n(2)$, segue che la probabilità di trovare certe coordinate per l'una particella e certe coordinate per l'altra - considerato che la probabilità è calcolabile col quadrato del modulo della funzione d'onda – sarà data da:

$$P(1,2) = |\psi_m(1)|^2 |\psi_n(2)|^2 = |\psi_m(1)\,\psi_n(2)|^2$$

da cui ne discende che esiste una funzione d'onda complessiva per le due particelle che verrà indicata con $\Psi(1,2) = \psi_m(1)\,\psi_n(2)$. Ora, per l'**indiscernibilità** più sopra menzionata, detta funzione d'onda dovrà essere indipendente da uno scambio di particelle (cosa che, per come è stata indicata la $\Psi(1,2)$, non risulta vera). Ma poiché è noto che, se due funzioni d'onda sono soluzioni dell'equazione di Schrödinger che governa l'evoluzione dei sistemi microscopici, anche le loro combinazioni lineari ne sono ancora soluzione, allora è chiaro che la cercata indipendenza dallo scambio di particelle sarà ottenuta considerando le due seguenti combinazioni lineari:

$$\Psi_s(1,2) = A[\psi_m(1)\,\psi_n(2) + \psi_m(2)\,\psi_\beta(1)$$

$$\Psi_a(1,2) = A[\psi_m(1)\,\psi_n(2) - \psi_m(2)\,\psi_\beta(1)$$

Ove $\Psi_s$ è funzione d'onda simmetrica e $\Psi_a$ funzione d'onda antisimetrica per scambio di particelle.
Le funzioni d'onda simmetriche corrispondono a due o più particelle che possono stare nello stesso stato quantico (bosoni), mentre quelle antisimmetriche corrispondono a particelle in cui due o più particelle non possono stare nello stesso stato quantico (fermioni). Sarà interessante notare che il Principio di Esclusione di Pauli, sul quale si fonda la varietà della materia conosciuta, deriva direttamente dalla constatazione che gli elettroni sono fermioni.

**Stato quantico e funzione d'onda di un sistema microscopico**

Lo stato quantico in cui si può trovare un sistema microscopico è rappresentabile con un vettore dello spazio hilbertiano – opportuna generalizzazione dello spazio euclideo ordinario da utilizzare per gli oggetti del mondo microscopico – e viene indicato col seguente simbolismo $|B>$, dove $B(x, y, z, t)$ rappresenta la *funzione d'onda* relativa allo stato del sistema descritto dalla coordinate $x, y, z$ nel tempo $t$.

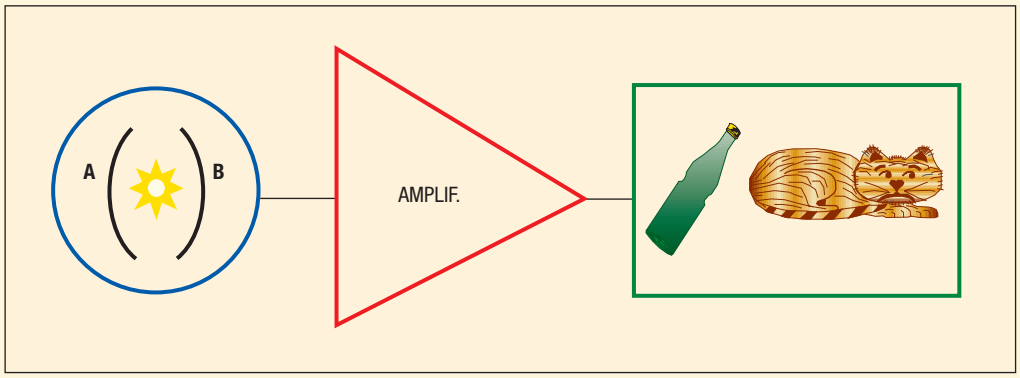
2 Questa affermazione di antichi filosofi in termini moderni deve essere estesa a tutta la realtà fisica che ricade o, direttamente, sotto i nostri sensi o, indirettamente, sotto le possibilità di rivelazione di appropriati ausili strumentali.

questa non sembra a chi scrive la sede adatta, che al principio di indeterminazione corrisponde un'altra singolare circostanza presente nella realtà microscopica, nota come principio di complementarietà. Secondo questo principio, esistono a livello microscopico proprietà complementari tali da escludersi l'una con l'altra, nel senso che la rivelazione dell'una, mediante esperimento, esclude la rivelazione dell'altra. Tali, per fare un esempio, sono l'aspetto corpuscolare e l'aspetto ondulatorio di una particella che, di volta in volta, può essere rivelata nell'una forma (*corpuscolo*) o nell'altra (*onda*), ma mai percepita nella sintesi delle due.

### 1.1.2. Il principio di sovrapposizione

Ma le sfide e la provocazione della fisica quantistica non si esauriscono qui. Un'altra circostanza, sbalorditiva per l'intuizione sviluppatasi nelle menti umane abituate all'impatto con oggetti macroscopici sottoposti alle leggi della fisica classica, è quanto asserito dal *principio di sovrapposizione*, in base al quale per gli oggetti microscopici è possibile che stati differenti relativi a un determinato ente, possano tranquillamente convivere in uno stato di non-definizione, da intendersi come combinazione lineare dei vari stati in cui esso può venirsi a trovare.
La singolarità di questa possibilità è stata messa in evidenza, in forma particolarmente icastica, dal cosiddetto paradosso del **gatto di Schrödinger**, che non fa altro che riportare gli effetti di questo principio nell'ambito della realtà macroscopica del mondo classico.
Le ragioni di stupore prodotte dalle scoperte della fisica quantistica sono tantissime e spesso, come si è già avuto occasione di osservare, così sconvolgenti da creare un profondo stato di disagio, al punto di portare il grande fisico Heisenberg a osservare che per capire la fisica quantistica bisogna sapersi liberare del pregiudizio classico, di cui ogni essere umano è inevitabilmente prigioniero, in quanto la fisica classica non

Un gatto viene chiuso in una stanza blindata in cui viene collocata una fiala sigillata contenente un gas velenoso. La fialetta è collegata attraverso un opportuno amplificatore con l'interno di un contenitore, dove un atomo in stato di eccitazione si trova situato fra le pareti "A" e "B" di materiale rispettivamente, assorbitore o fotoelettrico.




L'atomo, uscendo dallo stato di eccitazione, emetterà un fotone che andrà a colpire o la parete "A" o la parete "B" del contenitore. Quando il fotone colpisce la parete "A" verrà assorbito senza produrre alcun effetto, quando colpisce la parete "B" viene invece emesso un impulso elettrico che opportunamente magnificato dall'amplificatore riesce con un appropriato congegno a far esplodere la fialetta contenente il gas velenoso provocando la morte del gatto. Ora, in base al principio di sovrapposizione, l'atomo diseccitato si trova in uno stato di sovrapposizione fotone verso destra e fotone verso sinistra, dal quale si potrà uscire soltanto con una misura. Per cui, seguendo la catena degli effetti a cascata e ammettendo di poter descrivere il gatto in termini di vettori dello spazio hilbertiano si dovrà concludere che, finché non viene effettuata la misura, il gatto si troverà in uno stato di sovrapposizione fra vita e morte. Sarebbe, quindi, lo spettatore a far morire il gatto con una semplice osservazione! È la trasposizione del principio di sovrapposizione in ambito macroscopico che porta al paradosso.
Il **gatto di Schrödinger** ha fatto scorrere fiumi di inchiostro sul significato della partecipazione dell'osservatore al processo di misura e sulle implicazioni filosofiche di tipo idealistico che se ne possono dedurre.

è altro che la precisazione in termini matematici e rigorosi dell'esperienza quotidiana del mondo fisico, necessariamente basata su oggetti macroscopici.
Si può tentare di trovare esempi dalla fisica di tutti i giorni con i quali aiutarsi a capire alcune singolarità della fisica del microscopico, ma il gioco può essere pericoloso portando a grossolani errori e incomprensioni. Infatti, la difficoltà a comprendere lo stato di sovrapposizione, viene spesso illustrata con un esempio tratto dal mondo classico. Una moneta, sorteggiata nel gioco di "Testa" o "Croce", mentre ruota in aria, prima di impattare la superficie su cui verrà osservata, può essere assunta come immagine "classica" di uno stato non-definito di sovrapposizione Testa/Croce. Lo stato della moneta, ovviamente, si definisce in maniera stabile soltanto all'impatto - che può essere considerato come l'effettuazione di una misura - sulla superficie del tavolo.
In realtà, l'analogia è vera solo parzialmente. Intanto la moneta, mentre ruota, mantiene sempre ben distinte le due facce. Al limite con una moderna osservazione stroboscopica non sarebbe impensabile vedere a ogni istante una determinata faccia. Nel caso di una particella, viceversa, lo stato di sovrapposizione è proprio uno stato di totale non-definizione, in cui gli stati possibili sono sovrapposti e non-definiti e anche tali da prendere connotazione solo all'atto della misura. Ancora, di fatto lo stato di non definizione della moneta in volo può essere rappresentato con un *vettore di stato* dello spazio ordinario, mentre lo stato di sovrapposizione di una particella microscopica deve essere rappresentato con un vettore di stato dello spazio hilbertiano.
Le due situazioni hanno delle analogie ma sono profondamente diverse. Nel caso della moneta, i due stati sono sempre presenti e prendono forma stabile all'atto dell'impatto, nel caso di una particella microscopica lo stato di sovrapposizione è una "realtà indistinta", una sovrapposizione delle varie possibilità, che prende forma solo con la misura. È, dunque, il processo di misura che permette di configurare una delle varie possibilità altrimenti indistinguibili. Intorno a quest'ultimo concetto è nata una diatriba non ancora del tutto esaurita sul significato profondo dell'operazione di misura in fisica.

### 1.1.3. Il Paradosso EPR (Einstein-Podolsky-Rosen)

Partendo dalla considerazione che lo *spin* è una grandezza conservativa, si consideri un sistema quantistico costituito da due protoni, fra loro localmente molto vicini e con spin totale nullo. Questa situazione corrisponde ad avere gli spin dei due protoni, misurati lungo una direzione assegnata, orientati in sensi opposti, ovvero, se un protone ha spin $+h/2^{TM}$ lungo una direzione, l'altro avrà necessariamente spin $-h/2^{TM}$ lungo la stessa direzione. Per essere certi di questa situazione non occorre fare alcuna misura, esistono opportuni mezzi per accertarsene. Di fatto, non è neppure conveniente effettuare misure, perché in fisica quantistica le misure perturbano irrimediabilmente il sistema.
Se adesso si immagina che i due protoni si allontanino indefinitamente l'uno dall'altro fino a raggiungere enormi distanze reciproche (anche moltissimi anni luce, se è il caso!) si deve ammettere, per la menzionata conservazione, che la relazione di antiparallelismo degli spin resta conservata. Pertanto, ove si effettui la misura di una componente di spin di una delle due particelle lungo una direzione assegnata, forzandolo in uno stato determinato, necessariamente anche la particella lontana verrà forzata immediatamente in uno stato determinato del suo momento angolare in modo da conservare, lungo la stessa direzione, la relazione di antiparallelismo di partenza. Per indicare questa stretta interdipendenza fra particelle nel mondo anglosassone viene usata l'espressione *entanglement*. Questa azione istantanea a distanza è stata per lungo tempo considerata paradossale, finché J. Bell dimostrò che lo strano effetto, anche se apparentemente paradossale, è effettivamente verificato in natura [9].
Si osservi che il paradosso in esame era per mettere in difficoltà le conclusioni della fisica teorica: di fatto, così veniva argomentato, la possibilità di avere un effetto a distanza in un tempo nullo si configura come una violazione del cosiddetto principio di

località[3]. In realtà, il paradosso citato, non viola il principio di località, in quanto, essendo il risultato della prima misura (quella sul protone vicino) casuale come lo è quello della seconda misura (quella sul protone lontano), non è possibile con l'esperienza descritta, trasmettere informazione fra i due estremi e, quindi, produrre significativi effetti a distanza.
L'esperimento può essere descritto in termini macroscopici in forma più accessibile nel modo seguente. Un signore a Milano e il suo gemello a New York hanno ciascuno in tasca una coppia di palline, una bianca e una nera. Quando il signore di Milano estrae in successione palline bianche e nere con il 50% di probabilità il signore di New York estrarrà necessariamente palline nere e bianche con la stessa percentuale di probabilità. Con un tale procedimento è, ovviamente, impossibile trasmettere informazione fra Milano e New York. Infatti si sa che l'informazione è quella conoscenza passata al destinatario per rimuoverlo da uno stato di incertezza, ma nel caso del gemello di New York l'incertezza non viene diminuita dalla conoscenza della misura del gemello di Milano, dato che si dovrà limitare a ripetere semplicemente un'estrazione casuale[4].
La conferma di questo aspetto "paradossale" ha, tuttavia, una notevole portata e un impatto sconvolgente sul concetto di "realtà" che l'essere umano si è costruito.
In forma molto immaginifica, il fenomeno è stato descritto come un meccanismo secondo il quale due roulette, situate in località con distanze grandi a piacere, forniscono le stesse[5] estrazioni (sincronismo del caso?). Il concetto di separazione, così caro alla fisica classica, viene messo in discussione al punto di far affermare al fisico-filosofo B. d'Espagnat "*Un importante insegnamento della fisica contemporanea* "*fondamentale*" *è che la separazione spaziale degli oggetti è in parte anch'essa* un modo della nostra sensibilità" [3].

## 2. LA FISICA QUANTISTICA E IL COMPUTER

Si esamina, adesso, come la "nuova fisica" possa introdurre nell'universo del calcolo automatico i suoi strabilianti effetti. Si osserva, intanto, che per lungo tempo non si è data molta importanza alle modalità fisiche secondo le quali un dispositivo di calcolo viene realizzato. Soltanto recentemente, a seguito dell'incessante progresso della tecnologia di realizzazione dei moderni computer, si è cominciato a percepire che la forma fisica secondo cui una macchina di calcolo è realizzata non può non avere un impatto determinante sul suo funzionamento.
Nel corso della seconda guerra mondiale, il semplice passaggio dalla tecnologia elettromeccanica alla tecnologia elettronica consentì agli alleati di infrangere il segreto della mitica macchina criptografica "Enigma" di cui durante la prima parte del conflitto l'esercito tedesco disponeva per mascherare i contenuti delle proprie comunicazioni con decisivo vantaggio strategico. Il cambiamento di tecnologia portò come conseguenza una diminuzione dei tempi elementari di processo di un fattore 1000 (dal millisecondo al micro-

[3] Einstein ha mantenuto fermi due principi ritenendoli inviolabili anche per trasformazioni relativistiche, *il principio di località*, secondo il quale si afferma che ciò che avviene in *A* non può avere alcuna relazione con ciò che avviene in *B* se *A* e *B* sono separati da una distanza $\Delta\ell > c\Delta t$(eventi al di fuori del cono di luce); *il principio di causalità* secondo il quale nessuna trasformazione relativistica può capovolgere la relazione temporale fra causa ed effetto (la causa precede sempre l'effetto).

[4] Per estrema chiarezza si esamina il caso in dettaglio. Se l'estrazione *B* avviene *prima* dell'estrazione in *A*, si tratta inevitabilmente di estrazione casuale; se l'estrazione in *B* avviene simultaneamente a quella in *A* si tratta ancora di estrazione casuale (il "caso" si manifesta in *A* con palla bianca e in *B* con palla nera – e viceversa); se l'estrazione in *B* avviene *posteriormente* a quella in *A*, il risultato in *B* sarà *necessitato* da quello in *A*, ma ancora *casuale*: l'estrazione in *A forza* lo spin del protone facendolo precipitare in uno stato determinato (spin ↑ o spin ↓) *con legge casuale* e l'estrazione in *B mostra* lo spin del protone in uno stato determinato (spin ↑ o spin ↓) *con legge casuale*.

[5] In realtà, sulla base dell'esempio, i valori estratti dalle due roulette sarebbero reciprocamente negati, ma la circostanza non toglie nulla alla conclusione del ragionamento.

secondo!). Fu chiaro che la veste fisica di un computer aveva un effetto così determinante da poter trasformare un problema insolubile in problema solubile.
La questione può essere posta in termini più generali, domandandosi quali sono i limiti di calcolo raggiungibili con una assegnata realizzazione fisica. Esistono dei limiti che per un certo tipo di realizzazione fisica sono invalicabili. La questione è stata brillantemente visualizzata da Vazirani [12]: "*per la fattorizzazione di un numero di duemila cifre non si tratta del caso che non sarebbe possibile far lavorare tutti i computer del mondo per riuscirci* [...] *perché, anche se uno immaginasse che ogni particella dell'universo fosse un computer classico e calcolasse a pieno ritmo per l'intera vita dell'universo, la cosa sarebbe ancora insufficiente a fattorizzare un numero di quelle dimensioni*".
Ma la domanda può essere generalizzata ancora in maniera più drastica, ponendo la questione non in termini di tecnologia impiegata, ma in termini di fisica sottostante alle operazioni elementari di calcolo con cui funziona un computer. Questo imbarazzante versante della domanda è stato affrontato dal fisico Feynman già fin dal 1960 [4], dimostrando che *non c'è possibilità di far funzionare una macchina di Turing*[6] *classica in modo da simulare alcuni processi quantistici senza incorrere in un rallentamento di tipo esponenziale della macchina stessa*.
Poiché il computer si è dimostrato anche un potente ausilio per la simulazione e lo studio di svariate circostanze sperimentali, la conclusione di Feynman non poteva che provocare delusione e sconcerto nella comunità scientifica. Dunque, non si poteva evitare l'amara conclusione che il computer, il meraviglioso strumento che ha cambiato il volto della modernità, ha dei limiti invalicabili nelle sue prestazioni, tali da precluderne l'uso nella simulazione di determinati processi fisici.
Allo stesso tempo, non si poteva eludere neppure l'altra importante questione che si poneva spontaneamente e urgentemente sul tavolo: *è possibile immaginare un computer che funzioni secondo i principi della fisica quantistica*? O anche, *quali trasformazioni possono essere immaginate nel computer e nella grandezza che esso maneggia – l'informazione - ove si decida di riferirsi alla fisica quantistica piuttosto che alla fisica classica*?
A questa seconda possibilità cominciò subito a riflettere lo stesso Feynman [5], tentando di concepire una macchina funzionante sulla base dei principi della fisica quantistica e aprendo in tal modo un nuovo promettente capitolo per l'informatica. Lo scopo e i limiti del presente articolo ci impediscono di entrare in dettaglio sull'argomento che, in una quarantina di anni di ricerche, ha prodotto in ambito teorico notevoli risultati, per i quali si attendono nel prossimo futuro interessanti riscontri sul piano delle applicazioni. Qui basterà dire che tutta la "Teoria della Informazione classica" così come impostata da C. Shannon, è in corso di revisione.
La definizione tradizionale di unità di informazione – il *bit* – che poggia inevitabilmente su assunti di tipo classico, avendo come oggetto di riferimento un dispositivo a funzionamento classico quale un circuito *bistabile*, deve essere rivista ove si faccia riferimento a oggetti a funzionamento quantistico, quali lo spin di un elettrone o la polarizzazione di un fotone. Accanto al bit, basato su fenomeni di tipo classico, si deve collocare il *qu-bit*, la nuova unità di informazione basata su fenomeni quantistici. Il concetto di entropia informativa deve essere opportunamente riveduto, diventa essenziale il concetto di informazione accessibile, ovvero della informazione che effettivamente si riesce a estrarre da un sistema quantistico per effetto di un'operazione di misura, ma, ciò che da un punto di vista pratico è assai più importante, si comincia a valutare la possibilità teorica di concepire sistemi fisici con i quali effettuare operazioni impensabili con le tecniche classiche di computazione. Per fare un esempio, il caso, sopra menzionato, esemplificato da Vazirani per dimostrare l'impossibilità computazionale di problemi a elevata complessità combinatoria in termini classici, diventa solubile se affrontato in termini quantistici, come dimostrato dal celebre lavoro di Shor [10], successivamente confermato con una realizzazione

[6] *La macchina di Turing è il dispositivo teorico immaginato da Turing secondo cui è fondata l'architettura delle moderne macchine di calcolo*.

sperimentale [11], apparentemente risibile – nel contributo si descrive un sistema quantistico mediante il quale si riesce a fattorizzare il numero 15 - ma di grande portata in quanto dimostra la fattibilità, in termini di dispositivi a funzionamento quantistico, di una macchina in grado di risolvere l'arduo problema della fattorizzazione.
Ovvero, dopo questo risultato, le possibilità di risolvere il problema avanzato dal Vazirani diventano più concrete e più vicine.

## 2.1. Gli effetti quantistici applicati alle macchine di calcolo

Alla base del funzionamento di processi di calcolo molto promettenti stanno alcuni degli effetti quantistici sinteticamente descritti nei precedenti paragrafi. A questo proposito, però, varrà la pena di osservare che l'interesse per il calcolo quantistico non sta nel ripetere procedimenti e calcoli che possono essere eseguiti dai convenzionali computer a funzionamento classico, ma nel fatto che, mediante questa nuova tecnologia, operazioni che risultano impossibili con la tecnica tradizionale possono diventare possibili o, quanto meno, che operazioni eseguibili con scarsa efficienza con il calcolatore classico possono diventare molto più efficienti con la QC (*Quantum Computing*). A. Berthiaume e G. Brassard [1], in un celebre articolo, nel 1992, hanno posto termine alla discussione circa la superiorità del calcolo quantistico su quello classico dimostrando che la QC può battere in efficienza il computer classico sia di tipo deterministico che di tipo probabilistico. Resta, tuttavia, da osservare che i problemi sui quali si fonda la dimostrazione sono piuttosto particolari per cui la conclusione può lasciare qualche perplessità. Fra i problemi non solubili con mezzi deterministici, ma possibili in termini quantistici, si cita il *problema della generazione di numeri veramente casuali e il problema della fattorizzazione di numeri molto grandi*, di altissimo interesse per la criptografia. Ma sono già stati proposti algoritmi quantistici per il *problema della ricerca efficiente in database* (*database search problem*), per il *calcolo dei cicli Hamiltoniani*, per la soluzione del *problema del commesso viaggiatore* e di quello dei *logarimi discreti*. Infine, tutta la *simulazione di fenomeni quantistici*, così importante per l'esplorazione del mondo microscopico, precluso in forma classica, diventa possibile con la QC, come brillantemente presagito dal fisico Feynman.
Qui di seguito verrà esaminato, con maggiore dettaglio, come alcuni dei principi ricordati possano permettere di immaginare stupefacenti applicazioni. Si farà particolare riferimento a due esempi, sui quali la ricerca contemporanea si è particolarmente dedicata, in cui l'uso del computer quantistico consente di risolvere problemi di grande interesse non altrimenti solubili in forma classica.

### 2.1.1. Il principio di sovrapposizione e il parallelismo quantistico

Un sistema quantistico evolve secondo un'equazione scoperta dal fisico Schrödinger, fino alla effettuazione di una misura, all'atto della quale il sistema collassa in uno dei suoi stati possibili. L'equazione in questione ha la proprietà che la combinazione lineare delle sue soluzioni è ancora una sua soluzione, per cui se il sistema parte da una sovrapposizione di stati farà evolvere nel suo processo di evoluzione tutta la sovrapposizione di stati in blocco.
Si torni per un momento a considerare l'omologo del bit classico, il qu-bit, ossia l'informazione contenuta in un sistema quantistico a due stati, come lo spin di un elettrone.
Dove l'elettrone non sia in uno stato definito, ma in sovrapposizione di stati ↑ (Spin "su") e ↓ (Spin "giù"), qualora si assegni allo stato ↑ (Spin "su") il valore binario "0" e allo stato ↓ (Spin "giù") il valore binario "1", si dovrà concludere che il sistema elettrone si trova in uno stato che rappresenta la sovrapposizione di "0" e "1" – uno stato classicamente inimmaginabile!
Se adesso si procede a costruire un registro costituito da due elettroni, i cui stati stabili di spin saranno quattro (↑↑, ↑↓, ↓↑, ↓↓), corrispondenti ai quattro stati binari (00, 01, 10, 11), ove lo stato del registro non si trovi in uno stato definito, ma in sovrapposizione di stati, si dovrà concludere che il sistema "coppia di elettroni" (= il registro a due elettroni) si troverà in uno stato che rappresenta la sovrapposizione delle coppie di stati

00, 01, 10, 11. E dunque, in un registro a due celle, possono convivere in stato di sovrapposizione tutti e quattro gli stati indicati. Nel registro sono sovrapposti e simultaneamente scritti i simboli 00, 01, 10, 11, estraibili con un'opportuna misura.

Segue una prima importante conclusione: mentre per registrare i quattro valori indicati in forma classica occorrerebbero quattro registri a due celle, in forma quantistica i quattro valori indicati sono contenibili in un solo registro a due celle! E procedendo nella costruzione, un registro a 1 cella può contenere $2^1$ valori, un registro a 2 celle può contenere $2^2$ valori, un registro a 3 celle può contenere $2^3$ valori ... un registro a *n* celle potrà contenere $2^n$ valori.

La seconda conclusione è la seguente: se il sistema quantistico "registro" viene lasciato evolvere, esso farà evolvere simultaneamente tutti gli stati in sovrapposizione, realizzando una sorta di funzionamento in parallelo per il quale si usa l'espressione parallelismo quantistico[7]. Se l'equazione di evoluzione verrà scelta in modo tale da portare alla soluzione di un determinato problema, tutto ciò che occorrerà fare sarà lasciar evolvere il sistema verso la soluzione desiderata, alla quale esso si porterà, valutando simultaneamente tutti i dati in sovrapposizione fornitigli. Questa tecnica può risultare enormemente vantaggiosa qualora si debba utilizzare il computer per valutare una serie di dati numerosissima.

Si consideri, per esempio, il problema menzionato da Vazirani consistente nella fattorizzazione di un numero grandissimo. Il procedimento teorico, in termini di sovrapposizione quantistica sarà il seguente: si carichi il primo di due registri quantistici eguali con una sovrapposizione di ingressi che rappresentano tutti gli interi compresi fra "o" e $\lceil \sqrt{n} \rceil$ (ossia il primo intero maggiore di $\sqrt{n}$) e il secondo con una sovrapposizione di ingressi che rappresentino tutti gli interi compresi fra $\lceil \sqrt{n} \rceil$ e *n*, essendo *n* il numero di cui si cerca la fattorizzazione. Si lasci evolvere il sistema in modo che esso esegua in *parallelismo quantistico* i prodotti di tutte le coppie dei numeri inseriti nei due registri, allocando i risultati in un terzo registro. Leggendo i risultati su questo terzo registro, quando si riuscisse a trovare il numero *n*, dovrebbe essere possibile risalire alla coppia di numeri cercata che l'ha prodotto.

Tuttavia, la probabilità di trovare il numero corretto sarà in realtà molto esigua, in quanto nel registro le *non-soluzioni* saranno in stato di sovrapposizione con la *soluzione*, rendendo assai ardua l'operazione immaginata.

Per risolvere il problema, occorrerà trovare il modo di far interferire le non-soluzioni in modo da cancellarle: in questo consiste, appunto, la tecnica elaborata da Shor con la quale il problema può essere risolto.

Naturalmente, occorre poter disporre di registri quantistici di appropriate dimensioni, ciò che costituisce un problema tecnologico an-

[7] Non è possibile passare sotto silenzio l'importanza del parallelismo quantistico in grosse questioni scientifiche irrisolte dalle quali emergono contraddizioni insanabili alla luce delle conoscenze attuali. Il parallelismo quantistico è stato, infatti, invocato come possibilità per la soluzione di un rompicapo proposto dalla teoria della evoluzione. Si cita dalla "Rete del fisico" di H.P. Dürr. "*È noto che la teoria della evoluzione di Darwin, che è oggi considerata al di sopra di ogni sospetto, ha qualche difficoltà a spiegare, anche in termini quantitativi, lo sviluppo delle forme di vita, persino le più primitive, dal gioco combinato di pure mutazioni casuali e successive selezioni delle varianti più adatte alla vita. Sebbene sia notoriamente molto difficile valutare in modo affidabile il tempo necessario per un tale meccanismo, qualcosa indica che l'età della nostra terra, calcolata sulla base di considerazioni cosmologiche e geologiche (pari a 4.5 miliardi di anni) sarebbe per questo lungo e faticoso processo troppo breve, di molti ordini di grandezza. Non c'è pertanto da stupirsi se si continuino ad avanzare congetture, secondo le quali non si potrebbe, in ultima analisi, rinunziare ad una componente teleologica nell'evoluzione. Ma i processi quantomeccanici si sviluppano diversamente. Per passare da uno stato ad un altro, non è necessario che i passi intermedi siano reali, essi possono essere semplicemente effettuati in forma virtuale per mezzo del campo quantistico di possibilità le future possibilità di realizzazione vengono in certo qual modo spontaneamente attuate e utilizzate per il processo di sviluppo nel tempo. Con una coerente sovrapposizione di possibilità, uno sviluppo può pertanto procedere in modo più mirato che con una serie di prove incoerenti e casuali di queste stesse possibilità*".

cora irrisolto e di non facile soluzione. Tuttavia, come è stato anticipato la procedura è stata verificata per la scomposizione di un numero molto piccolo ($15 = 5 \times 3$), dimostrando che l'operazione è possibile. L'enorme vantaggio della procedura proposta sta nel parallelismo quantistico che permette di effettuare, per così dire, in un "colpo solo", l'enorme numero di prodotti richiesti per ottenere la fattorizzazione desiderata.
L'uso di due numeri primi molto grandi e del loro prodotto è il fondamento di un moderno processo di criptazione ritenuto molto sicuro per le sopramenzionate ragioni spiegate da Vazirani. La cripto-analisi di un testo criptato con questa tecnologia richiede per l'appunto la fattorizzazione di un numero intero molto grande. È di conseguenza evidente l'enorme interesse che la fattorizzazione di grandi numeri interi possiede per ovvi motivi di sicurezza nazionale e di controllo sociale.

### 2.1.2. Criptografia assolutamente ermetica

Un altro problema per il quale l'impiego delle proprietà quantistiche sembra schiudere promettenti orizzonti è quello relativo alla soluzione del cosiddetto problema del corriere presente nei sistemi criptografici. Questo problema è[8] relativo al fatto che qualunque trasmissione criptografica protetta include l'inevitabile impiego di un corriere per il trasporto della chiave. E, manifestamente, il corriere è il punto debole di tutto il sistema (esso stesso può "tradire", o, essere sequestrato e costretto a tradire). Non giova pensare al fatto che le due parti possano incontrarsi per lo scambio delle chiavi una volta per tutte, preventivamente a qualsiasi collegamento, perché ovvie ragioni di sicurezza consigliano di cambiare ad ogni collegamento la chiave. Dunque, alle due parti, se vogliono comunicare standosene nella propria sede, non resta altro che affidarsi ad un corriere.

A beneficio del lettore non specialista e per chiarezza di esposizione, verrà ricordato che i sistemi di criptografia hanno notevoli similitudini con una cassaforte, nella quale si possono distinguere relativamente alla sicurezza due aspetti.
***a.*** Quello della sicurezza fisica, rappresentato dalla robustezza del sistema di fronte a effrazioni operate con mezzi fisici (strumenti da scasso di vario genere, lancia termica ecc.). La sicurezza fisica di una cassaforte corrisponde al problema del corriere in un sistema di trasmissione criptografico.
***b.*** Quello della sicurezza logica, rappresentato dalla complessità e non falsificabilità della chiave di apertura. La sicurezza logica di una cassaforte corrisponde al problema di costruire algoritmi matematici sufficientemente complessi tali da non poter essere facilmente violati da chi illegittimamente tenti di infrangere il sistema.
È possibile dimostrare teoricamente che si possono ottenere messaggi criptografati a ermeticità assoluta ove, a ogni sessione, si ricorra a chiavi realizzate con sequenze casuali di dati, in modo da non fornire al criptoanalista della parte avversa "dati storici" su cui poter lavorare. La tecnica di criptazione, qui di seguito illustrata, presenta un procedimento quantistico per realizzare scambio di chiavi tali da produrre assoluta ermeticità.

#### *2.1.2.1. Il canale di Ekert*

L'idea, su cui si basa il sistema proposto, è quella che un sistema quantistico può essere descritto come una sovrapposizione di stati stazionari che rimane indeterminata, fino al momento in cui si effettua una misura, la quale ha come effetto di *forzare* il sistema in uno stato definito fra quelli possibili.
Recentemente, A. Ekert [8] sulla base delle considerazioni sopra illustrate, ha proposto un canale criptografico, che sarà indicato come *canale di Ekert*, mediante il quale è possibile trasmettere una chiave criptografica con *ermeticità assoluta*. Il funzionamento è illustrato in figura 1.
Secondo questa proposta, l'utente "*A*" e l'utente "*B*" possono scambiarsi una chiave con l'assoluta certezza della segretezza procedendo nel modo seguente. Una *Sorgente EPR* invia una successione di protoni

[8] Recentemente, sono stati inventati sistemi criptografici a "chiave pubblica", per i quali il problema della chiave può essere eluso. Detti sistemi, basati sulla difficoltà di decomposizione di numeri primi molto grandi, per la quale, come si mostra nel testo, si intravede la possibilità di effrazione con mezzi quantistici, non sono da considerarsi in prospettiva assolutamente ermetici.

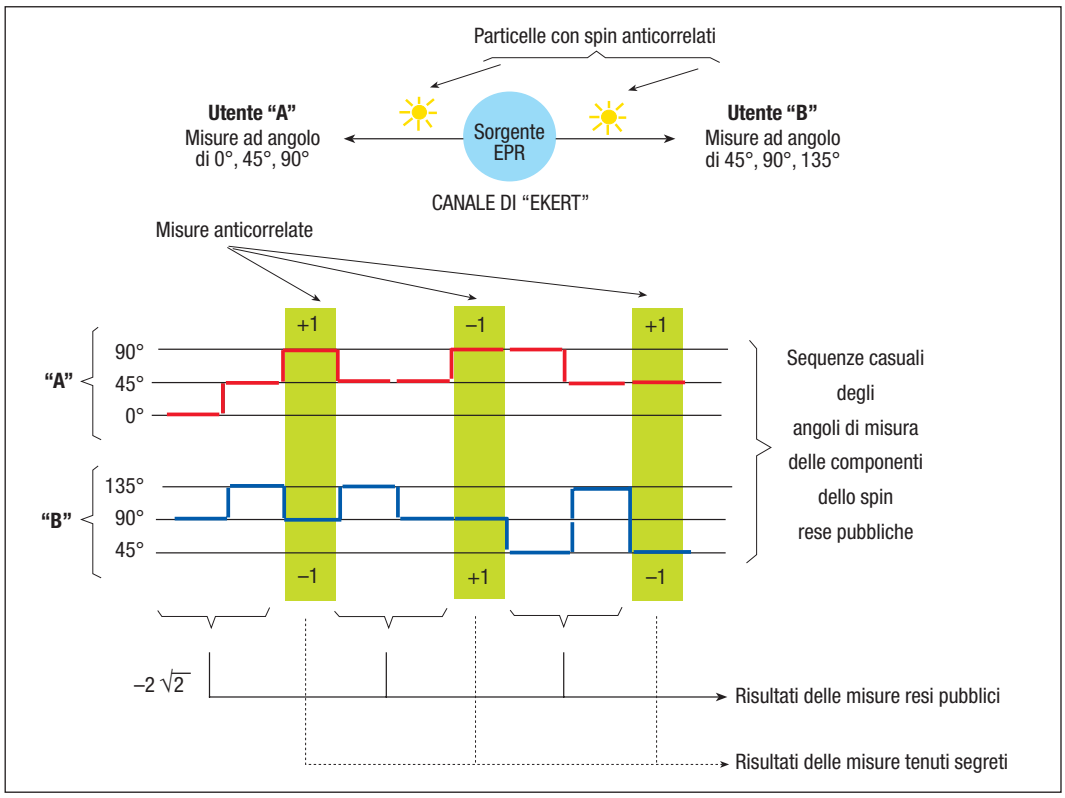



**FIGURA 1**
*Illustrazione schematica del canale di Ekert per trasmissioni assolutamente ermetiche. La sequenza di chiave è data dai risultati delle misure effettuate dalle due parti in contatto lungo le stesse direzioni*

prelevandoli da una coppia a momento angolare totale nullo, indirizzandone uno verso l'utente "*A*" e il suo associato verso l'utente "*B*". I due utenti misurano la componente angolare della particella ricevuta secondo angoli perpendicolari all'asse di comunicazione pari a 0°, 45°, 90° (utente "*A*") oppure pari a 45°, 90°, 135° (utente "*B*"). L'angolo di misura viene cambiato per ogni particella ricevuta secondo una sequenza "casuale" (differente per "*A*" e per "*B*") che viene resa di dominio pubblico: in particolare, la pubblicazione delle sequenze degli angoli di misura permette ai due interlocutori in contatto di suddividere le misure che essi effettuano in due classi, quelle compiute da entrambi con angoli concordi e quelle compiute con angoli discordi.

Secondo i risultati della fisica quantistica, se "*A*" e "*B*" misurano il momento angolare delle particelle con lo stesso angolo di orientamento i valori che ottengono sono totalmente anticorrelati (lo stato del protone ricevuto da "*A*" è esattamente l'opposto dello stato del protone ricevuto da "*B*"), se invece "*A*" e "*B*" effettuano le misure con angoli di orientamento differenti il risultato ottenuto – come verificabile sperimentalmente – sarà pari per entrambi a $S = -2\sqrt{2}$. Alla fine del processo di trasmissione, "*A*" e "*B*" rendono pubblici anche i risultati delle misure effettuate con orientamenti discordi, ma tengono rigorosamente segreti i risultati delle misure ottenute per orientamenti concordi. Se durante tutta l'operazione indicata il canale non è stato disturbato (nel senso che non sono state effettuate misure da parte di chicchessia per scopi fraudolenti nell'intento di ricavare informazione, intercettando la particella inviata a uno dei due utenti in contatto, misurandone lo spin secondo un certo orientamento per poi rinviarla al legittimo destinatario) gli utenti "*A*" e "*B*" troveranno il valore $S = -2\sqrt{2}$ per orientamenti discordi, con cui potranno concludere con certezza che non sono stati spiati. D'altro canto, i risultati delle misure per orientamenti concordi, tenuti segreti, costituiranno per entrambe le parti una sequenza, a loro soltanto nota, che potrà essere adoperata come chiave criptografica assolutamente sicura (più precisamente, ciascuna parte avrà la sequenza negata dell'altra parte, ma ciò non costituisce ovviamente difficoltà di sorta).

Con i dati resi pubblici è possibile a un utente illegittimo conoscere gli istanti in cui gli utenti legittimi effettuano misure anticorrelate trasmettendosi la sequenza di chiave, ma detta sequenza non può, da questi essere conosciuta perché tenuta segreta dagli interlocutori autorizzati. Si potrebbe, tuttavia, pensare di compiere un attacco criptoanalitico, senza essere scoperti, effettuando misure, *esattamente negli istanti* in cui "*A*" e "*B*" compiono essi stessi le misure, secondo identici orientamenti. Ma poiché le sequenze degli orientamenti secondo cui si fanno le misure sono casuali, questa possibilità è esclusa.
L'obiezione più ingegnosa che è stata avanzata è quella secondo la quale sarebbe comunque possibile ingannare gli utenti legittimi utilizzando una sorgente *EPR* fasulla in grado di trasmettere tre particelle alla volta, invece che due, procedendo, quindi, a intercettare sistematicamente il terzo fascetto di particelle per immagazzinarlo opportunamente, rinviando la misura degli spin, per non essere scoperti, in un momento successivo alle misure effettuate dai legittimi interlocutori. In tal modo, l'ente intercettatore potrebbe disporre degli stessi dati degli utenti autorizzati mettendosi in grado, sulla base dei dati da questi resi pubblici, di ricostruire la sequenza di chiave di cui essi vengono a disporre. Il problema della conservazione di particelle per memorizzarne lo stato è stato preso in esame nella valutazione della possibilità di costruire una *moneta non falsificabile* [13] e costituisce una difficoltà la cui soluzione pratica non si ritiene attualmente realistica. Di conseguenza, anche la possibilità di ingannare gli interlocutori autorizzati mediante una sorgente *EPR* fasulla non è possibile.

## 2.2. Problemi di fisica realizzabilità

La realizzabilità fisica di dispositivi per QC è fortemente condizionata da un fenomeno noto come "decoerenza quantistica", ossia l'inevitabile effetto dell'interscambio fra un sistema quantistico e l'ambiente in cui esso è immerso. Per tornare, pur con tutte le cautele del caso, a un parallelismo col mondo classico già menzionato, il fenomeno della decoerenza può essere paragonato all'impatto di una moneta lanciata con la superficie di un tavolo: all'impatto la "sovrapposizione di testa e croce", presente durante il volo conseguente al lancio, scompare immediatamente facendo precipitare la moneta in uno stato definito. Similmente, la sovrapposizione quantistica, per molti versi assai differente dalla sovrapposizione classica di testa e croce nel lancio di una moneta, viene a cadere per effetto della interazione con l'ambiente, in cui il sistema quantistico si trova necessariamente immerso. Lo stupefacente effetto della sovrapposizione degli stati con la conseguente possibilità di ciò che è stato chiamato parallelismo quantistico, viene messo in seria discussione dal fenomeno della decoerenza. Comunque, molti progressi in questa direzione sono stati effettuati ed è da ritenere che in prospettiva il fenomeno delle decoerenza possa essere in qualche modo superato [6].
Qui di seguito, si fornisce qualche informazione sulle tecnologie attualmente allo studio presso i laboratori attivi in questo tipo di ricerche per realizzare dispositivi in grado di funzionare sui principi della fisica quantistica.

### 2.2.1. Confinamento ionico lineare ("Trapped Ions")

Nel 1995 Cirac e Zoller [2] hanno messo a punto una tecnologia detta di *confinamento ionico lineare* (*trapped ions*). Secondo questa proposta un gruppo di ioni[9] viene sistemato linearmente in un'area di confinamento realizzata mediante opportuno campo elettromagnetico. I due stati stabili dei q-bit, rappresentati dai vari ioni (un q-bit per ione), sono dati dallo stato di riposo dell'ione e dallo stato di eccitazione del medesimo. I singoli ioni (Figura 2) sono allineati come a formare un registro e possono essere singolarmente irradiati mediante impulsi di luce laser. Tali impulsi laser provocano transizioni negli stati ionici eccitandoli convenientemente e, se il caso, posizionando questo particolare registro in uno stato di sovrapposizione.
Gli ioni confinati nel registro hanno tutti la stessa carica ed esercitano l'uno sull'altro

[9] In alcuni esperimenti sono stati usati ioni di Berillio $Be^{+}$.

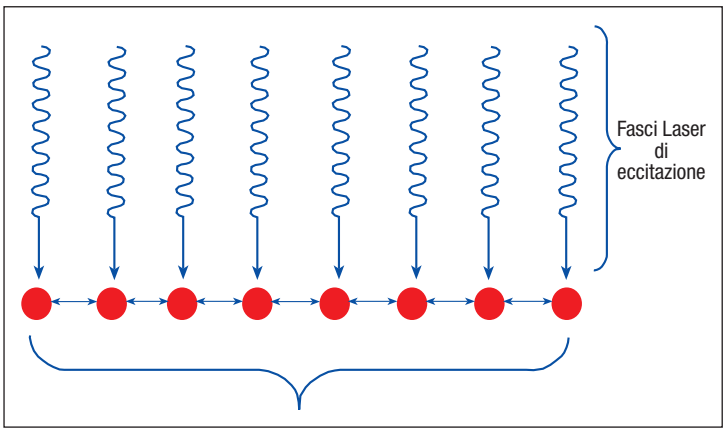



**FIGURA 2**
*Registro a confinamento ionico lineare (trapped ions)*

una repulsione mutua di tipo elettrostatico per modo che il movimento di ciascun ione si trasmette a tutti agli altri creando vari possibili movimenti collettivi detti *fononi*. Con singoli raggi laser, come indicato in figura, si può agire sui singoli ioni separatamente in quanto la distanza interionica di separazione fra le particelle è stabilita in modo da essere molto più grande della lunghezza d'onda del laser di eccitazione. A mezzo di opportune manipolazioni è anche possibile realizzare correlazione (*entanglement*) fra le singole coppie di bit. Questo dispositivo costituisce un notevole passo avanti nella tecnologia di realizzazione di computer quantistici: come si vede esso permette di realizzare *bus quantistici*, in via di principio, delle dimensioni volute.
Il tempo di decoerenza per q-bit immagazzinati con confinamento lineare è stato misurato in migliaia di secondi. Il meccanismo di decoerenza più significativo, fra l'altro ancora non perfettamente chiarito, è il riscaldamento dei *modi* dei fononi.
A partire dai risultati sperimentali finora ottenuti è prevedibile che entro i prossimi dieci anni sia possibile realizzare registri a confinamento lineare contenenti qualche decina di ioni. L'ostacolo più significativo che dovrà essere superato è costituito, come già menzionato, dal fenomeno della decoerenza.

### 2.2.2. Risonanza magnetica nucleare (NMR)

Il metodo della risonanza magnetica nucleare utilizza come supporto per il qu-bit lo spin del nucleo atomico agendo su di esso mediante campi magnetici esterni. È stato possibile con questo sistema realizzare semplici gate logici che agiscono su di un singolo qubit con l'uso dei campi prodotti con radiofrequenze, mediante i quali si può interagire sugli spin con buona precisione. Azioni più complesse dirette ad influenzare mediante un qu-bit molti altri qu-bit hanno trovato difficoltà di implementazione, per la necessità di far sì che gli spin interagiscano fra loro.
I sistemi *NMR* si distinguono per il fatto che il segnale ottenibile da una singola molecola è troppo debole per essere rivelato; è, pertanto, necessario ricorrere a molte molecole per ottenere un segnale utilizzabile. Questa circostanza non costituisce di per sé un problema in quanto per quantità minime di una sostanza chimica si ha a disposizione uno sterminato numero di molecole. La difficoltà nasce dal fatto di riuscire a fare in modo che tutte le molecole nell'effettuare il calcolo partano dallo stesso stato iniziale. Nel 1997, il problema è stato risolto da alcuni ricercatori riuscendo a ottenere uno *stato puro* da una miscela ottenendo in tal modo di far partire il sistema dallo stesso stato. Alcuni ricercatori sono piuttosto scettici sulla possibilità di poter realizzare computer basati sul principio *NMR* funzionanti con molti qu-bit, dato che il rendimento del processo per l'ottenimento dello stato puro diminuisce sensibilmente al crescere del numero delle molecole. Sono segnalati anche problemi relativamente alla possibilità di riuscire a influire sui singoli spin in presenza di numerose molecole.
Si ritiene che sia possibile arrivare in termini temporalmente accettabili alla sperimentazione di computer *NMR* con non più di 6 qubit. Questa dimensione consentirebbe, tuttavia, di risolvere alcuni interessanti problemi e sembra più a portata di mano rispetto a soluzioni ottenute con altre proposte.

### 2.2.3. Spin nucleare basato sulla tecnologia al silicio

Recentemente, si è avuto notizia [7] della possibilità di realizzare dispositivi per la QC che impieghino la tecnologia dello "stato solido" già collaudata con enorme successo in tutta la moderna tecnologia di produzione elettronica. L'idea è di incorporare gli spin nucleari in un dispositivo elettronico rivelandoli mediante opportune tecniche di controllo. Gli spin elettronici e nucleari sono accop-

piati mediante l'interazione iperfine. Sotto convenienti condizioni è possibile effettuare un trasferimento di polarizzazione fra i due tipi di spin riuscendo a rivelare lo spin nucleare attraverso i suoi effetti sulle proprietà elettroniche del campione. Sono già stati sviluppati dispositivi funzionanti a bassa temperatura su strutture di GaAs/$Al_xGa_{1-x}A_s$ che sono state incorporate in appropriate nanostrutture.
I ricercatori ritengono che la realizzazione di Quantum Computer basati sulla tecnica al silicio sia una eccezionale sfida che vada affrontata data la possibilità di sfruttare l'enorme rendita di posizione conseguibile dalla conoscenza tecnologica accumulata nella fabbricazione di dispositivi elettronici convenzionali allo stato solido. L'intento è quello di raggiungere ancora più piccole dimensioni e maggiore complessità di funzionamento.
La proposta su cui si lavora è quella di utilizzare il silicio Si come *host* e il fosforo $^{31}P$ come *donor*. A concentrazioni sufficientemente basse e alla temperatura di 1.5 K, è noto che il tempo di rilassamento degli spin elettronici ammonta a migliaia di secondi, mentre il tempo di rilassamento dello spin nucleare del $^{31}P$ è dell'ordine dei $10^{18}$ s, presentando quindi ottimi valori per la computazione quantistica. Il sistema proposto è anche in grado di fornire un buon isolamento dei qu-bit da qualsiasi grado di libertà che ne possa provocare la decoerenza. Con la tecnologia proposta diventerebbe possibile realizzare dispositivi per la computazione quantistica codificando l'informazione negli spin nucleari degli atomi "donor" di dispositivi al silicio opportunamente drogati. Le operazioni logiche sui singoli spin verrebbero eseguite applicando campi elettrici esterni e le misure sui valori di spin verrebbero eseguite usando correnti di elettroni polarizzati di spin.

## 3. SCENARI FUTURI

Cercare di prevedere il futuro è sempre un esercizio rischioso. È, tuttavia, possibile affermare che quaranta anni di studi teorici sull'argomento Quantum Computing forniscono solidi elementi per ritenere che si avranno interessanti ricadute sul piano realizzativo, se non nell'immediato, certamente a breve. L'avvento di computer quantistici "tascabili" non sembra imminente e forse nemmeno interessante. La QC non si pone come tecnologia concorrenziale alle attuali moderne macchine di calcolo in termini di realizzazione (maggiore economia, più ridotte dimensioni ecc.), ma piuttosto in termini di permettere la soluzione di problemi con la tecnica attuale dichiarati non solubili. I primi passi secondo questa modalità di implementazione saranno, pertanto, sicuramente compiuti nella direzione di macchine specializzate nella soluzione di problemi particolarmente ardui, con importanti ricadute sul piano teorico-pratico, non solubili o difficilmente solubili con le tecniche classiche tradizionali.
Si vuole chiudere con un'osservazione, ripresa dal premio Nobel H.P. Dürr, per mostrare come le idee che si sviluppano dalla fisica quantistica possano portare a conclusioni di carattere generale molto interessanti. L'esempio del sorteggio e della decoerenza hanno una evoluzione di direzione opposta a quella che generalmente si incontra nei fenomeni naturali. In questo caso, gli infiniti orientamenti della moneta durante il volo - e della corrispondente sovrapposizione quantistica - all'impatto con il piano del tavolo - con l'ambiente, nel caso quantistico - si riducono a due stati possibili soltanto (dal molteplice al semplice!), procedendo in una direzione di maggior ordine.
"*Orbene*" – osserva il Dürr, a questo proposito, - "*questo processo di ordinamento non avviene soltanto nel procedimento intenzionale che chiamiamo "misura", bensì questa trasformazione del possibile nel fattibile avviene anche senza il nostro intervento. Inoltre, questo continuo processo di coagulazione conferisce significato assoluto al tempo. Lo svolgersi del tempo rispecchia un ininterrotto processo di evoluzione. Perciò l'evoluzione non è in realtà nel tempo, ma piuttosto tempo ed evoluzione sono la stessa cosa, secondo la loro caratteristica natura. Il presente di volta in volta contrassegna il continuo formarsi del reale dal possibile, corrispondendo ad un progressivo processo di ordine*".

## Bibliografia


[1] Bethiaume A, Brassard G: *The Quantum Challenge to Complexity Theory*. Proceedings of the 7th. IEEE Conference on Structure in Complexity Theory, 1994, p. 2521-2535.

[2] Cirac JL, Zoller P: Quantum Computation with Cold Trapped Ions. *Physical Review Letters,* Vol. 74, 1995, p. 4091-4094.

[3] d'Espagnat B: *Alla ricerca del Reale*. Boringhieri – Torino, 1983.

[4] Feynman RP: There is Plenty of Room at the Bottom. *Eng. and Science*, Vol. 23, 1960, p. 22-36.

[5] Feynman R: Quantum Mechanical Computers. *Optics News*, Vol. 11, 1985, p. 11-20.

[6] Giulini D, et alii: *Decoherence and the appearance of a classical World in Quantum Theory*. Springer, Berlin 1996.

[7] Kane BE: A Silicon-based nuclear spin quantum computer. *Nature*, Vol. 393, 14 May 1998, p. 133-137.

[8] *Phys. Rev. Lett*. 67, 1991, p. 661-663.

[9] Sakurai JJ: *Meccanica quantistica moderna*, p. 220 e segg. Zanichelli- Bologna 1990.

[10] Shor P: *Algorithms for Quantum Computation: Discrete Logarithms and Factoring*. Proceedings of the 35th Annual Symposium on Foundations of Computer Science, 1994, p. 124-134.

[11] Vandersypen LKM, et alii: *Experimental realization of Shor's quantum algorithm using nuclear magnetic resonance*. IBM Almaden Research Center, San Josè, CA 95120. Solid styate and photonics Laboratory. Stanford University, Stanford CA 94305-4075. December 2001.

[12] Vazirani U: Conference Sponsored by the Santa Fe Institute, Los Alamos Nat. Lab. and the University of New Mexico, 1994.

[13] Wiesner S: *Sigact news*. Vol. 15, 1983, p. 78-88.


### *Letture consigliate*

Dürr HP: *Das Netz des Physikers*. Gutenberg, Frankfurt 1989.

Feynman RP: *La fisica di Feynman Voume III, Meccanica Quantistica*. Addison Wesely P. Co. London,1970.

Persico E: *Fundamental of Quantum Mechanics*. Prentice Hall, Inc. Englewood Cliffs, N.J.1957.

Sakurai JJ: *Meccanica quantistica moderna*. Zanichelli- Bologna 1990W.

Nolting: *Grundhurs Theoretische Physik: Quantenmechanik*. Wieweg, Wiesbaden 1997.

Williams CP, Clearwater SH: *Explorations in Quantum Computing*. Spreinger-Telos, Berlin, 1997.

Brooks M: *Quantum Computing and Communications*. Springer, London 1999.


Emanuele Angeleri, laureato in Fisica nel 1956 con il massimo dei voti. Esperienza industriale (1956-1986) nel settore nucleare e delle Telecomunicazioni. Docente di Teoria dei Codici (1979-1985) presso la Scuola Superiore di Telecomunicazioni SIP. Professore associato (Università di Milano, 1986) di Teoria della Informazione e della Trasmissione. Autore di numerose pubblicazioni scientifiche e di alcuni libri nel campo della Teoria della Informazione.
eangeleri@tiscalinet.it

# INTEGRAZIONE DI RETI E SERVIZI: IL "TRASPORTO DELLA VOCE" SULLE RETI IP

Nuove applicazioni e servizi di rete rendono difficile il compito di chi deve scegliere cosa implementare e soprattutto quali risorse sono necessarie alle sempre più diffuse applicazioni che si stanno facendo largo nell'ambito delle reti IP: tra le tante innovazioni una sfida interessante è quella di portare un servizio, come la fonia, sulla stessa infrastruttura utilizzata per il traffico dati. La migrazione verso il mondo "all IP" è sicuramente motivata dalla riduzione di tutti i costi associati al comparto fonia.

**Bruno Fadini**
**Antonio Pescapè**
**Giorgio Ventre**

3.4

## 1. INTRODUZIONE

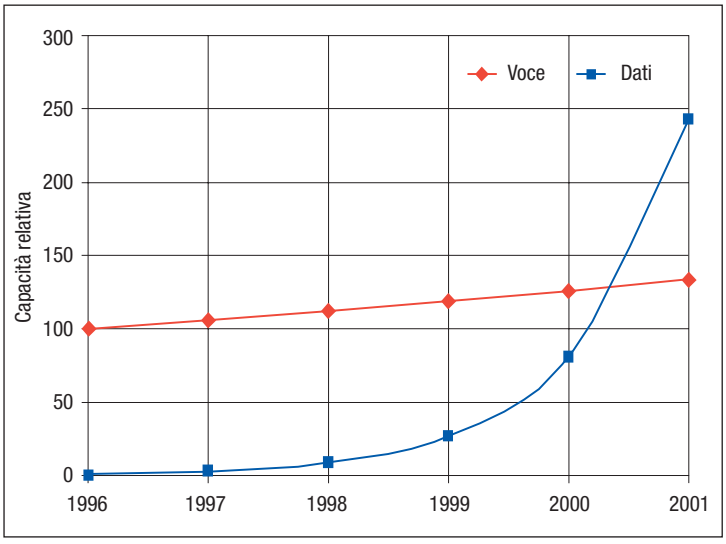
Da quando, nel febbraio 1995, Vocaltec rilasciò la prima versione dell'applicazione *Internet Phone* (versione $\beta$), la trasmissione di segnali vocali su reti IP ha fatto molta strada. Durante gli ultimi anni, si è assistito a una crescita eccezionalmente rapida della VoIP (*Voice over IP*) sia in termini di dimensioni di mercato, sia in termini di evoluzione tecnologica: la spinta principale che ha portato all'utilizzo della telefonia IP è, ad oggi, determinata dal minor costo consentito dalle reti IP rispetto a quelle tradizionali. Un utente che utilizza la VoIP per le sue comunicazioni interurbane o internazionali, a fronte di una qualità che, probabilmente, è leggermente inferiore rispetto a quella offerta dalla telefonia tradizionale, ma comunque superiore rispetto a quella, per esempio, della rete mobile GSM (*Global System for Mobile Communication*), può ottenere notevoli risparmi rispetto alle tariffe praticate dalle compagnie telefoniche. Innanzitutto, il chiamante paga una normale tariffa locale per collegarsi, con un circuito dedicato, a un *gateway* locale della rete IP e utilizza, poi, la connettività IP su distanze nazionali o internazionali, a costi marginali, fino a un gateway terminale, ove si ricollega, a circuito e con tariffe ancora locali, al chiamato (*Toll Bypass*). Inoltre, le strutture di rete IP utilizzate sono più efficienti a causa della compressione del segnale vocale e della commutazione a pacchetti. Questo vale sia a livello *Intranet* che *Internet* [12].
È pensabile, tuttavia, che questa situazione è destinata a cambiare in tempi piuttosto rapidi: nel momento in cui la VoIP raggiungerà, infatti, una qualità del servizio paragonabile a quella tradizionale, interverranno nuove normative a regolamentare il settore, e questo porterà probabilmente a un aumento dei costi. D'altra parte senza una regolamentazione delle tariffe, né i *service provider*, né i costruttori di apparecchiature VoIP possono giustificare gli investimenti necessari a ottenere una VoIP qualitativamente accettabile: il 26 aprile 2001 l'"*US House of Representatives Telecommunications Subcommitte*" ha approvato un testo che regolamenta l'offerta dei servizi voce su

Internet. Le cause "scatenanti" che sono alla base del fenomeno "voce su IP" sono fondamentalmente le seguenti:

- negli ultimi anni, il traffico dati, che tipicamente ha sempre viaggiato su reti a pacchetto, utilizzando in particolare il protocollo IP, ha superato in volume il traffico voce (Figura 1);
- si è assistito, negli ultimi tempi, a una crescita esponenziale degli utilizzatori di reti IP.

Un tale orientamento del mercato ha comportato una drastica riduzione dei prezzi di tutti gli apparati di rete concernenti il mondo IP. Pertanto, attualmente, un'impresa che decide di investire in una tecnologia di rete integrata basata su protocollo IP, può ipotizzare un risparmio sui costi globali di creazione dell'infrastruttura di rete che oscilla fra il 30 e il 50% (Figura 2).

**FIGURA 1**
*Andamento del traffico dati e del traffico voce dal 1996 al 2001*



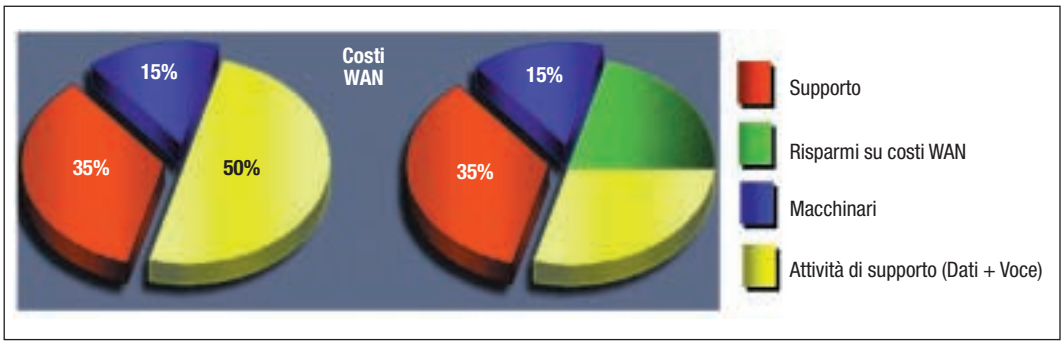

Negli ultimi anni, si è assistito, nell'ambito della Pubblica Amministrazione così come in ambito aziendale, a una ricerca sempre maggiore di efficienza e di riduzione dei costi: la possibilità di sfruttare le nuove tecnologie appare, quindi, come un elemento indispensabile per raggiungere questi obiettivi.

Le reti basate su protocollo IP sono state progettate e costruite per garantire la comunicazione ad applicazioni di tipo non *real time* (per esempio, *e-mail* e trasferimento di *file*) che sono caratterizzate da un traffico discontinuo che prevede picchi molto elevati di banda richiesta, ma anche tempi più o meno lunghi di inattività. Da questo punto di vista, quindi, voler utilizzare le reti a commutazione di pacchetto per la trasmissione di segnali vocali può apparire non conveniente, a meno di non inquadrare la cosa in un contesto più ampio che prevede la trasmissione attraverso un'unica rete di una moltitudine sempre crescente di servizi, che vanno dalla telefonia tradizionale alla telefonia con servizi avanzati, alla trasmissione dati, alla videoconferenza, al *broadcasting* televisivo ecc..

Il concetto fondamentale, dunque, è quello di integrazione: ovvero, un'integrazione di diversi servizi su di un'unica infrastruttura di rete, resa possibile da quella che in questi anni si è andata affermando come lingua franca, ossia il protocollo IP (*Internet Protocol*) (Figura 3). L'obiettivo di un'architettura di rete convergente, nel cui ambito si inquadra la VoIP, è, quindi, quello di realizzare una rete che da un lato, protegga gli investimenti economici sostenuti, supportando tutti gli elementi di rete e le interfacce analogiche e digitali presenti nella rete stessa e dall'altro, permetta di introdurre nuove tecnologie nella rete di accesso, di trasporto e negli elementi di commutazione, creando un'infrastruttura aperta e orientata ai servizi [1].

Agli evidenti benefici derivanti dall'adozione di una rete integrata si contrappongono, però, i problemi legati alla configurazione e alla gestione, che derivano dalla necessità

**FIGURA 2**
*Riduzione dei costi associata all'integrazione di Dati/Video/Voce*
*Fonte: Cisco Systems*



di supportare, garantendo gli opportuni requisiti di *Qualità del Servizio*, un traffico con caratteristiche assolutamente differenti (isocronia, *loss rate* ecc.) trasportato finora su reti differenti. Per fare ciò, è necessario capovolgere uno dei paradigmi fondamentali del mondo IP: non più tutto *Best Effort* ma classi di traffico a priorità differenti, introducendo opportune politiche di QoS (*Quality of Service*). Nella figura 4, è riportato un semplice grafico che mostra come in assenza di opportuni meccanismi di QoS la qualità di una conversazione telefonica su reti IP sia notevolmente compromessa.
La migrazione del traffico voce sulle reti IP comporta, comunque, la necessità di poter determinare l'ammontare di traffico generato dai canali vocali una volta noti il codec e il compressore di segnale.
Il modello di rete adottato sia dalle aziende che dalla Pubblica Amministrazione è tipicamente quello di una direzione generale (*headquarter* o centro stella) e di una serie di uffici distaccati (*branchoffice*) distribuiti sul territorio cittadino, nazionale o, addirittura, internazionale dipendente dalle dimensioni dell'azienda e quindi del business nella quale quest'ultima è impegnata. In un'architettura di questo tipo, i punti critici, in termini di banda, sono sicuramente i *link* di WAN (*Wide Area Network*) per i collegamenti tra le diverse sedi dell'azienda e, in situazioni di reti

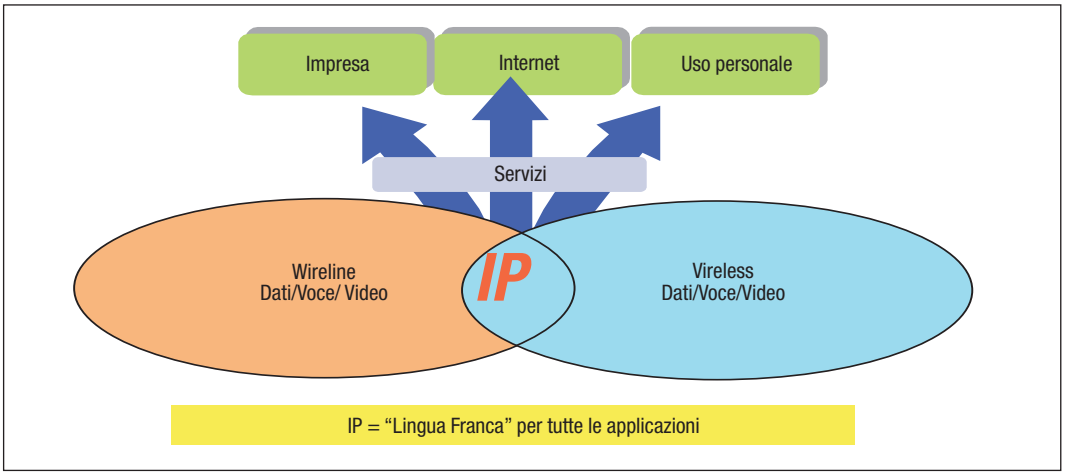



**FIGURA 3**
*Il protocollo IP diviene il comune denominatore del trasporto dell'informazione*

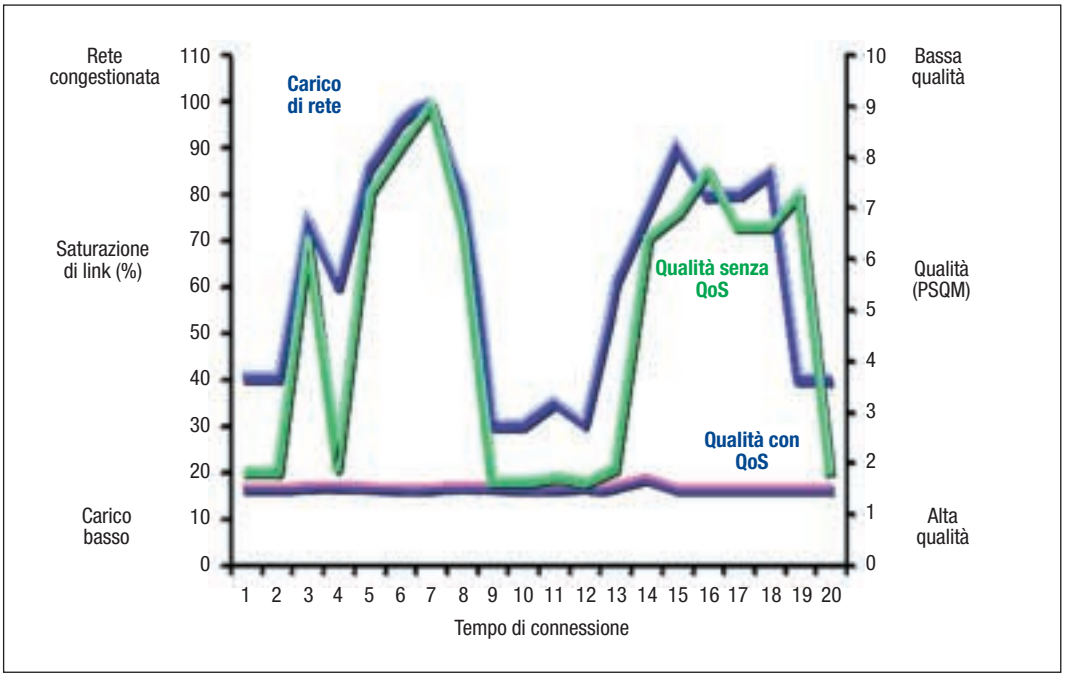



**FIGURA 4**
*Traffico VoIP con e senza QoS*
*Fonte: Cisco Labs*

particolarmente antiquate, il cablaggio interno di un edificio e, quindi, i collegamenti locali alla singola sede.
Una delle prime necessità di un progettista di architetture di rete, o semplicemente di un amministratore, propedeutica a un corretto funzionamento della rete stessa, è la corretta individuazione dei diversi applicativi che utilizzeranno tale rete, per poter poi individuare correttamente le risorse necessarie per ciascuno di essi.
Per i flussi dati tradizionali, che per molti anni sono stati i soli utilizzatori di reti IP, la progettazione è relativamente semplice. Normalmente, è già nota a priori la capacità di banda necessaria per ciascun servizio e, inoltre, poiché, in genere, si tratta di servizi sensibili alla perdita dei pacchetti (*loss sensitive*) ma non ai ritardi (*time sensitive*), anche in caso di momentanee congestioni della rete si verifica solo un rallentamento delle normali operazioni. Utilizzando reti IP per il trasporto di flussi di traffico real time, le condizioni che si presentano sono decisamente diverse, anche se il traffico voce, fra le varie caratteristiche peculiari, ha anche quella di avere un *bit rate* relativamente costante, rendendo possibile un calcolo più preciso di quella che è l'occupazione di banda. Il calcolo è reso meno agevole, anche, dalla necessità di considerare che per i segnali vocali sono stati sviluppati numerosi algoritmi di campionamento e compressione, ognuno con caratteristiche proprie e bit rate differenti [14, 17]. Inoltre, fino ad oggi, le reti dati e quelle telefoniche, anche in ambito locale, quale quello di un *Campus* o di una azienda, sono sempre state distinte, per cui non si hanno riferimenti sulle modalità di utilizzo della rete telefonica. Una soluzione, ampiamente diffusa, è basata sull'impiego di modelli più o meno raffinati per calcolare l'ammontare di traffico telefonico che bisogna supportare; inoltre, molti dei più recenti PBX (*Private Branch Exchange*) contengono una memoria storica, che consente sia di avere un *log* accurato delle chiamate effettuate, sia di conoscere verso quali direttrici principali sono state indirizzate (per esempio, se si tratta di chiamate interne alla rete locale oppure se si tratta, invece, di chiamate esterne).

## 2. STANDARD E ASPETTI CARATTERISTICI DELLE COMUNICAZIONI VOCALI SU RETI IP

Le prime applicazioni della tecnologia VoIP utilizzavano Internet quale mezzo trasmissivo per inviare traffico audio, in tempo reale, a due o più utenti collegati tramite *computer*. Il primo prodotto *software* per la telefonia su IP è stato presentato da Vocaltec all'inizio del 1995, come ricordato nell'introduzione: girando su un PC (*Personal Computer*) multimediale, questo software permetteva agli utenti di parlare nei microfoni e di ascoltare tramite gli altoparlanti o le cuffie. Da allora, la tecnologia è notevolmente migliorata, permettendo agli utenti di utilizzare anche i telefoni tradizionali per effettuare una chiamata su Internet e di gestire tipologie di servizi sempre più sofisticate, che integrano *media* differenti nelle loro comunicazioni. Sebbene all'inizio la VoIP sia stata utilizzata solo come un sistema relativamente semplice per fornire servizi di trasporto voce a basso costo fra due postazioni IP, da allora, l'interesse sempre crescente verso la possibilità di fornire servizi integrati di voce, video e dati ha provocato una notevole espansione degli scopi originari. La telefonia su IP abbraccia, oggi, un largo spettro di servizi, fra i quali, ad esempio, la conferenza tradizionale, la capacità di controllo della chiamata, il trasporto multimediale e l'integrazione fra la telefonia e il WEB, l'e-mail e l'*instant messaging*.
Inizialmente, il sistema di trasporto per i servizi VoIP era solamente la Internet pubblica, ma ora un numero sempre maggiore di società di trasporto (*carriers*) e di fornitori di servizi (service provider) stanno utilizzando (o, in alcuni casi, costruendo) le proprie dorsali (*backbone*) per fornire ai propri clienti una elevata qualità del servizio. Contrariamente a quello che si potrebbe pensare, il vero punto di forza della VoIP non riguarderà solo l'avere tariffe più ridotte rispetto alla telefonia tradizionale quanto, piuttosto, la capacità di fornire nuovi servizi a valore aggiunto.
L'introduzione della voce sulle reti a commutazione di pacchetto può essere realizzata secondo diversi modelli funzionali, ciascuno dei quali implica differenti soluzioni architetturali. I modelli più diffusi sono: *PC-to-PC*, *PC-to-Pho-*

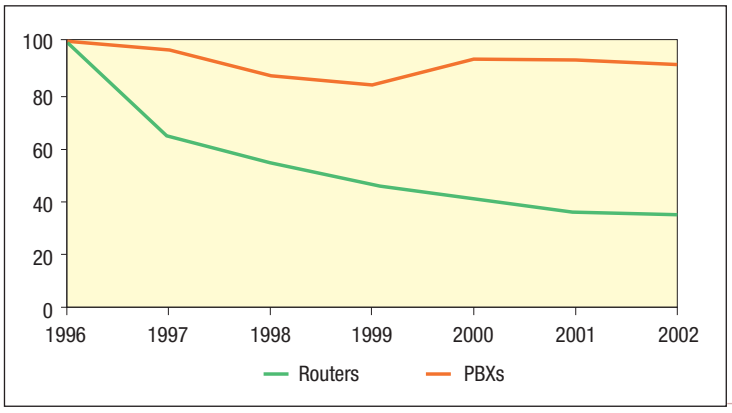
*ne*, *Phone-to-Phone*, *fax-to-fax* e, più in generale, tutti i modelli di *messaging service* (*e-mail-to-fax*, *Web-to-fax*, *fax-to-e-mail*, *universal in-box*, *WWW/fax on demand server*) [7].
Tale introduzione è, attualmente, in corso e, quindi, il processo di standardizzazione dei sistemi relativi all'*Internet Telephony* non si è ancora concluso. Al momento, sono coinvolti tre enti internazionali di standardizzazione: l'ITU (*International Telecommunications Union*), l'IETF (*Internet Engineering Task Force*) e l'ETSI (*European Telecommunication Standard Institute*) con alcuni consorzi (per esempio, Softswitch, H.323ORG, Vivida ecc.) [2]. La gestione delle chiamate voce sulla rete IP è, al momento, indirizzata da due differenti proposte, elaborate in ambito ITU e IETF, che sono rispettivamente **H.323** [11] e **SIP** (*Session Initiation Protocol*) [10].
Per quanto concerne l'interlavoro tra la rete telefonica a commutazione di circuito e la rete VoIP sembra, invece, essersi affermato il MGCP (*Media Gateway Control Protocol*), sviluppato, inizialmente, in ambito IETF e ora standardizzato, sotto l'acronimo H.248, in un *working group* congiunto ITU-IETF.
Analogamente, le soluzioni per il trasporto della segnalazione SS7 sulle reti IP sembrano, ormai, consolidate nell'ambito del gruppo di lavoro SIGTRAN del IETF.

I sostenitori di ITU-T affermano che H.323, storicamente nato prima, abbia ormai ottenuto il supporto di tutti i fornitori di apparati VoIP, mentre i sostenitori di SIP dubitano dell'interoperabilità dei prodotti H.323 di differenti *vendor* e, allo stesso tempo, evidenziano i vantaggi di SIP in particolar modo per quanto riguarda la ridotta segnalazione durante l'attivazione della chiamata. A supporto di quest'ultima tesi, vi è il fatto che SIP sempre più viene introdotto tra i protocolli supportati dai vendor, che comunque mantengono la compatibilità "storica" con H.323 [18]. Volendo confrontare **H.323** *e* **SIP** bisogna comunque osservare che lo scopo dei due standard è piuttosto differente. Mentre SIP è un protocollo per la segnalazione e il controllo di sessioni multimediali, H.323 delinea un'architettura completa per lo svolgimento di conferenze multimediali, comprendente la definizione dei formati di codifica a livello applicativo, la definizione di protocolli per la segnalazione e il controllo, per il trasporto dei flussi audio, video e dati e per la gestione degli aspetti di sicurezza, tutto ciò con riferimento ad architetture di rete locali. SIP risulta una valida alternita soprattutto per le sue caratteristiche di semplicità rispetto ad H.323 che deriva dal fatto di avere ereditato certe caratteristiche dall'ambiente delle reti locali e certe altre dallo standard H.320 (per ISDN, I*ntegrated Services Digital Network*).

## 3. IP TELEPHONY E VOIP: COSA SONO E QUANDO CONVENGONO

I motivi principali che hanno giustificato l'introduzione prima e l'affermazione poi della voce su IP sono numerosi: la vasta diffusione delle reti dati e l'enorme diffusione di Internet (*la rete delle reti*); la larga diffusione dei PC; il basso costo per la realizzazione sia dei DSP (*Digital Signal Processing*), sia delle reti IP costituite da apparati passivi, schede di rete e cablaggio strutturato (Figura 5) e, infine, la possibilità di creare nuovi servizi.
L'introduzione di questa innovazione, in un'architettura di rete esistente, può seguire due approcci differenti che portano a scegliere due diverse architetture, diverse anche nel nome: *IP Telephony* e VoIP.
Con VoIP si intende il puro trasporto: l'utente continua a usare apparecchi telefonici comuni e non ha alcuna cognizione del fatto che la sua conversazione viene trasportata dalla rete dati. Ne vi è alcuna integrazione applicativa tra telefonia e sistema informativo.
Con IP Telephony si definisce un servizio telefonico completo, che considera come parte della rete IP anche il terminale d'utente. In pratica, l'utente dispone di un terminale telefonico interconnesso alla rete con un indirizzamento IP, oppure dispone di un programma software che realizza sul suo PC, con opportuno equipaggiamento multimediale, le analoghe funzioni.
Con IP Telphony sono, quindi, possibili mag-

**FIGURA 5**
*Andamento dei prezzi relativi di router e PBX in Europa*
*Fonte: Gartner Group*



giori interazioni tra il mondo della telefonia e il mondo delle applicazioni dati. È, per esempio, possibile la gestione di una rubrica in linea, con chiamata da telefono su segnalazione iniziata da un PC, oppure l'automazione delle funzioni di ricerca da parte di un *call-center* tramite interazione diretta tra il telefono dell'operatore e il sistema informativo.

Ma quanto è realmente conveniente l'integrazione del servizio di fonia alla preesistente infrastruttura di rete dati? Tra le diverse osservazioni, quella che forse merita maggior interesse si riferisce al peso relativo che hanno la voce e i dati nei confronti della infrastruttura generale. Se si è, infatti, nella situazione in cui il numero di conversazioni contemporanee tra due sedi è "relativamente" limitato e la quantità di dati scambiati è molto grande, è allora probabile che la rete dati abbia un proprio naturale "sovradimensionamento". In queste condizioni, trasportare la voce significa semplicemente aggiungere qualche kbit/s al traffico dati senza che ciò debba comportare degli oneri aggiuntivi.

Una decina di canali fonici su VoIP richiedono, per esempio, un centinaio di kbit/s. Se la rete dati utilizza linee a 2 Mbit/s o più, la voce può transitare sopra senza che sia necessario alcun ampliamento di capacità particolare. Il costo del VoIP è, pertanto, il solo costo di ammortamento degli apparati, o meglio, delle parti di apparati necessarie a introdurre la funzione di *voice gateway*. In merito all'IP Telephony, occorre fare anche una considerazione di opportunità: è, ormai, prassi comune considerare come apparati di distribuzione di LAN (*Local Area Network*) degli *switch* (o *hub*) a 10/100 Mbit/s. I 100 Mbit/s, inoltre, per alcune stazioni *end-user* sono un prerequisito funzionale (*workstation* con sistemi CAD, sistemi per le elaborazioni e le analisi geofisiche). Esistono, tuttavia, edifici con cablaggi aventi un'età superiore ai 10/12 anni e che, probabilmente, presentano infrastrutture di rete antiquate e cablaggi "non strutturati". In queste situazioni, quindi, può nascere l'esigenza di ristrutturare i cablaggi.

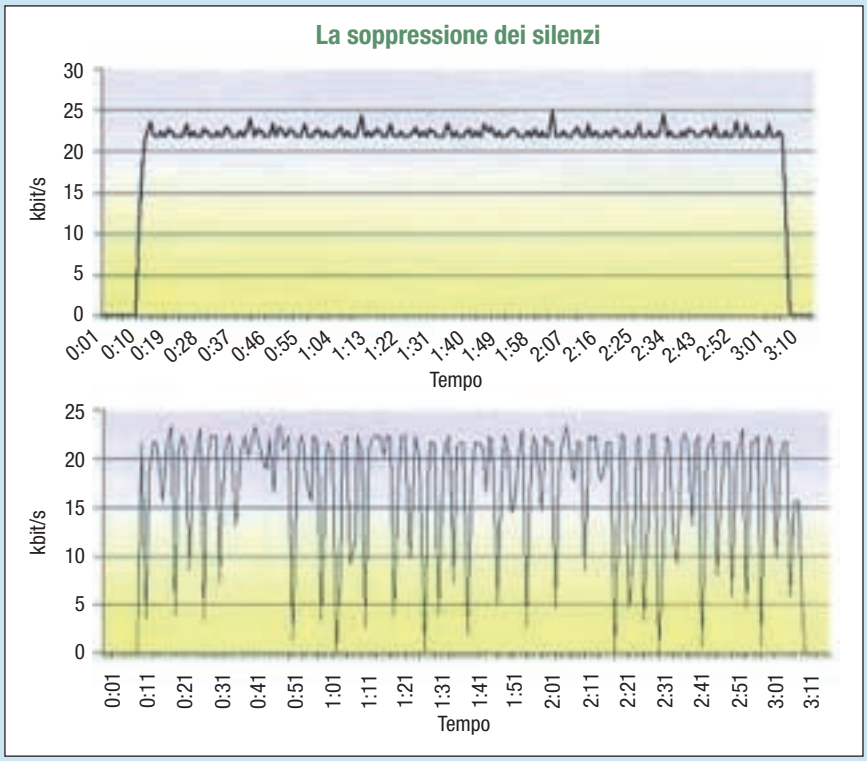
In quest'ottica, una postazione di lavoro tradizionale richiede un impianto a tre cavi/PDL (uno per i dati, uno per la fonia e un terzo per un eventuale seconda uscita dati *network printer* o per un apparato telefonico/fax), mentre l'uso di telefoni IP con funzione di hub e di switch integrato, porta all'impiego di un solo cavo per la PDL. Questo significa poter dividere per tre il costo di installazione del cablaggio. In più, si avrebbe una maggiore flessibilità conseguente al fattore IMAC (*Installation, Move, And Change*): se tutte le porte LAN in distribuzione agiscono da *trunk* e l'indirizzamento in VLAN (*Virtual LAN*) di quanto è presente sulla scrivania viene fatto dal "telefono/switch", spostare una stazione significa, semplicemente, disconnetterla dal cavo di distribuzione, trasferirla (telefono compreso, che diventa personale come il PC o la workstation) e riconnetterla con la stessa configurazione nella nuova postazione senza che sia necessario riconfigurare gli apparati di accesso e di distribuzione, o le permutazioni o le predisposizioni di utente in centrale: si conseguono, così, una serie di risparmi "indotti" che hanno, comunque, un peso di rilievo [9].

Per terminare questa semplice analisi e completare le considerazioni di ordine pratico, un'ultima variabile da tenere presente è il costo, attualmente critico, dei collegamenti geografici (WAN). Nel corso degli ultimi mesi, il costo delle CDN (*Collegamenti Diretti Numerici*) è decisamente diminuito. Nel 1996, un circuito CDN 64 kbit Milano-Roma costava quasi 18 milioni di lire, oggi, il costo è sceso di un ordine di grandezza ed è grossomodo di circa € 775. Inoltre, attualmente, esistono diverse soluzioni architetturali che prevedono la possibilità di acquistare da un operatore di rete il servizio di VPN (*Virtual Private Network*) grazie al quale è possibile ottenere la garanzia di determinati requisiti di QoS a costi contenuti. Pertanto, in situazioni in cui il costo dei *link* di WAN non è critico le ragioni dell'integrazione di dati/voce risiedono, perciò, altrove e cioè nella capacità di integrare facilmente, e a un costo marginale, applicazioni CTI (*Computer Telephony Integration*) e CRM (*Customer Relationship Management*), nella manutenzione effettuabile da persone non esperte di telefonia standard, nel poter disporre di sistemi *unified messaging* integrati, nel poter attivare *voicemail* sofisticate e di dimensioni fuori dall'ordinario e, infine, nel gestire una sola rete su cui convergono dati e voce.

## 4. LA "QUALITÀ" DEL SEGNALE VOCALE

Per comprendere gli aspetti legati al trasporto della voce su reti IP è necessario soffermarsi prima sulle caratteristiche del segnale vocale, e poi sui problemi del trasporto su reti a commutazione di pacchetto. Per garantire determinati livelli di servizio sulla rete PSTN (*Public Switched Telephone Network*) la qualità della voce è legata al livello sonoro, al ritardo, alla eco, alla comprensione, all'intelligibilità, al rumore, al *fading* e alla diafonia. In aggiunta, sulle reti IP, bisogna risolvere problemi legati al ritardo, al *jitter*, alla comprensione, alla perdita di pacchetti, alla disponibilità di banda e alla compressione. È importante sottolineare come i parametri relativi alla qualità del servizio offerto siano soggettivi; il valore che ogni utente assegnerà a ciascuno dei servizi elencati è una questione individuale: per esempio, i viaggiatori abituali daranno la preferenza a servizi quali le carte di credito o la portabilità del numero, mentre una famiglia con bambini può preferire piuttosto avere un numero maggiore di caselle di posta. Viceversa, per quanto riguarda, invece, la qualità della voce, è possibile definire in maniera oggettiva alcuni vincoli. Un approccio abbastanza frequente, anche se non sempre utile, per verificare la qualità della voce, consiste nell'utilizzare le tecniche della rete telefonica tradizionale, ovvero di confrontare a schermo le forme d'onda e misurare il rapporto segnale/rumore e la distorsione armonica totale. Quest'analisi può avere senso, però, solo nel caso in cui una qualunque differenza nella forma d'onda sia indice di una distorsione non voluta introdotta dalla rete: utilizzando alcuni *codec* a bit rate molto basso il segnale ricevuto, in realtà, non ha la stessa forma d'onda del segnale vocale trasmesso, bensì ne è una riproduzione soggettiva dovuta al tipo particolare di codec utilizzato. Pertanto, appare evidente che per misurare la qualità del traffico voce che attraversi reti IP sono necessari diversi metodi di prova.
Un metodo utilizzato è quello basato sulla misura della qualità vocale percepita (PSQM, *Perceptual Speech Quality Measure*, definito in *ITU-T Recommendation P.861*). Tale metodo analizza in modo oggettivo segnali vocali con una larghezza di banda di 300-3400 Hz.

La soppressione delle pause o dei silenzi viene effettuata mediante un rilevatore di attività vocale o **VAD** (***Voice Activity Detector***). Quando si ha l'attività vocale, il VAD consente ai pacchetti di essere trasmessi. Quando, invece, si ha lo stato di silenzio, il VAD "chiude" momentaneamente la trasmissione. Poiché, normalmente, la conversazione umana è fatta per circa la metà da silenzi, o per meglio dire è *half-duplex* (in genere, si parla uno per volta), l'utilizzo di tale dispositivo permette un risparmio medio di banda, in termini di flusso aggregato, di circa il 30-40 %. Nella figura, si può apprezzare la variazione di capacità di canale necessaria alla trasmissione di un flusso voce codificato con la raccomandazione ITU-T, G.729, nel caso in cui il VAD sia disattivato o meno: il risparmio di banda nel secondo caso è sensibile.




*Confronto tra l'occupazione di banda in presenza e in assenza di VAD*

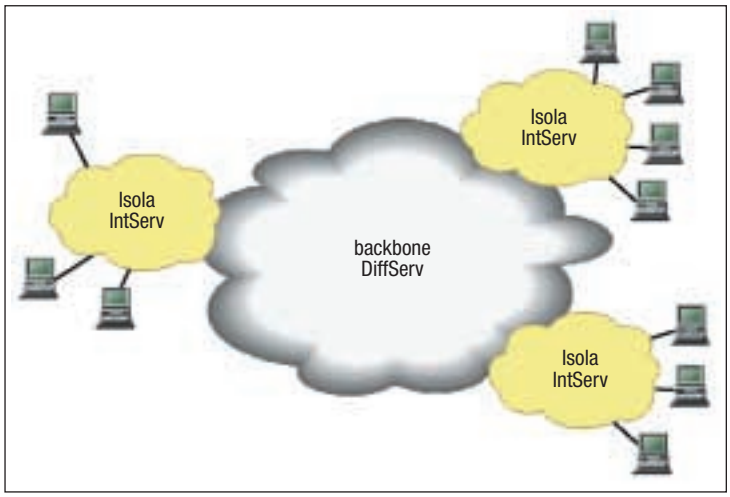



**FIGURA 6**
*Architettura integrata Int-Serv-Diff-Serv*

In modo molto semplice, si può affermare che il PSQM è una sorta di ascoltatore automatico, che confronta segnali vocali di riferimento con i campioni registrati dopo il passaggio attraverso la rete; applicando vari modelli uditivi e cognitivi, il PSQM è in grado di fornire una stima della degradazione percepita, che corrisponde abbastanza fedelmente ai risultati forniti da ascoltatori umani. Per quanto riguarda i cancellatori d'eco, anche in questo caso l'ITU-T ha definito alcuni metodi di misura (ITU-T *Recommendation* G.165 e G.168), che utilizzano metriche basate sulla risposta al rumore bianco oppure a segnali pseudo-vocali. Sono, tuttavia, ancora in fase di elaborazione soluzioni complete per la misura della qualità della voce su reti VoIP in maniera completa e affidabile, che sono la premessa indispensabile per utilizzare con successo tale tecnologia.

## 5. LA "QUALITY OF SERVICE" IN UNA RETE VOIP

Il servizio offerto dalle reti basate sull'utilizzo del paradigma Internet è, tradizionalmente, caratterizzato dall'assenza di qualsiasi tipo di garanzia sulle prestazioni della comunicazione (best effort). Tuttavia, lo sviluppo e la rapida diffusione di nuove applicazioni real-time, quali la telefonia su IP, l'audioconferenza e la videoconferenza, hanno portato a un radicale mutamento nelle richieste dell'utenza, tale da rendere più che mai urgente la realizzazione di architetture di rete che consentano di offrire prestazioni adeguate. Per realizzare una differenziazione del servizio si è reso necessario introdurre il concetto di Classi di Servizio (CoS): ogni classe raggruppa tipi di traffico con requisiti analoghi, ai quali la rete deve essere in grado di offrire prestazioni adeguate alle singole richieste. Con l'espressione "Qualità del Servizio" si fa, invece, riferimento a una serie di parametri che misurano la prestazione della comunicazione che è offerta dalla rete: caratteristiche di *throughput* e di banda, tempi di latenza, probabilità di perdita di pacchetti.
Un servizio può garantire un limite alla probabilità di scarto di pacchetti appartenenti a un certo flusso, oppure può garantirne l'assenza. Per soddisfare i requisiti sempre più stringenti relativamente ai parametri di QoS, l'IETF ha provveduto a definire due architetture avanzate per la rete Internet, denominate rispettivamente *Internet Integrated Services* e *Differentiated Services* [5,13]. Senza scendere in confronti dettagliati tra le due architetture, si può sicuramente affermare che il modello di utilizzo universalmente riconosciuto è quello mostrato in figura 6. Tecniche di ingegneria del traffico vengono poi utilizzate per l'allocazione ottima del carico e delle risorse sulla rete.
La qualità di servizio *end-to-end* relativa a una sessione VoIP è caratterizzata sia dalla qualità relativa alla fase di instaurazione della chiamata (*call set-up quality*), sia dalla qualità relativa alla fase attiva della chiamata (*call quality*). Complessivamente, la qualità di servizio end-to-end è influenzata da tutti i componenti intermedi attraversati, e cioè dal terminale IP, dalla rete d'accesso IP, dal backbone IP, dal VoIP gateway, dal VoIP *gatekeeper*, dalla rete telefonica, dal terminale telefonico.
La call set-up quality dipende dal ritardo nella instaurazione della chiamata (*call set-up delay*), percepito in termini di rapidità di risposta del servizio.
I fattori che contribuiscono al call set-up delay sono, sostanzialmente, il ritardo di set-up nella rete telefonica e nella rete di accesso IP, il ritardo di segnalazione nel backbone IP e di set-up negli elementi di rete (media gateway, *signalling gateway*, *media gateway controller, gatekeeper* ecc.), e, infine, il tempo di ac-

cesso e di elaborazione nei *server* dei servizi di *directory* o di autenticazione.
La call quality dipende, invece, dal ritardo end-to-end (*end-to-end delay*), che ha effetti sulle situazioni interattive della chiamata, ovvero in tutte quelle situazioni in cui il dialogo fra i due interlocutori necessita di risposte frequenti da parte dell'uno o dell'altro e dalla qualità del parlato end-to-end (*end-to-end speech quality*), che coinvolge anche situazioni non interattive della chiamata. In questo caso, quella che viene a mancare è la possibilità di comprendere le parole pronunciate dall'interlocutore, indipendentemente dal fatto che queste prevedano o meno replica. Anche l'eco contribuisce alla qualità della comunicazione, divenendo tanto più fastidiosa quanto maggiore è il ritardo end-to-end. Per eliminarla è necessario utilizzare appositi cancellatori d'eco.
Per quanto concerne l'end-to-end delay, il ritardo percepito dagli interlocutori deve essere costante e contenuto, al fine di mantenere la comunicazione interattiva. Ritardi inferiori a 15 ms non creano alcun fastidio, mentre l'interattività diventa difficoltosa per ritardi superiori a 400 ms. Valori intermedi di ritardo consentono un'interattività accettabile, ma richiedono la presenza di cancellatori d'eco ed è comunque consigliabile che il ritardo non superi i 150 ms.
Un problema tipico delle reti dati è la variabilità del ritardo, nota come jitter, che tende a corrompere l'intelligibilità della comunicazione. È, quindi, necessario rimuovere il jitter, compensandolo con l'ausilio di memorie tampone (*buffer*). L'end-to-end delay è determinato da molteplici contributi quali il ritardo di *bufferizzazione* nel terminale IP (scheda audio o scheda telefonica, *modem* o adattatore di rete), il ritardo di codifica e decodifica (formazione della trama, elaborazione, *look-ahead*), il ritardo per costituire i pacchetti (in trasmissione) e quello dei buffer (in ricezione) e il ritardo di rete (propagazione, trasmissione, accodamento ed elaborazione nei *router* e nei gateway e ritardi dovuti ai meccanismi protocollari), trascurabili nelle reti telefoniche standard.
Con riferimento alla qualità del servizio, è interessante valutare l'efficienza delle reti a commutazione di pacchetto nel caso del trasporto di segnali vocali. Il confronto con la commutazione tradizionale di circuito può avvenire sia da un punto di vista percettivo dell'utente finale (MOS, *Mean Opinion Score*), sia prendendo in esame la caratterizzazione di parametri end-to-end; ma deve tenere in considerazione anche l'efficienza di utilizzo delle risorse di rete [3].
Per quanto concerne quest'ultimo aspetto, sono state identificati due differenti tipi di efficienza: quella real-time, intesa come la capacità della rete di trasportare traffico real-time, senza influenzare il servizio offerto ad altri tipi di traffico e l'efficienza di trasporto che identifica, invece, la capacità di sfruttare le risorse di rete combinando al meglio traffico real time con quello best-effort, rispettando le richieste in termini di QoS.
Gli studi effettuati in proposito hanno evidenziato che garanzie deterministiche sul ritardo di trasmissione in una rete a pacchetto richiedono una sovra-allocazione di risorse; in altre parole, per ogni chiamata devono essere allocate più risorse di quante ne sarebbero state impiegate calcolando il bit rate di codifica del flusso dati e l'intestazione (l'*overhead*) introdotta dagli *header*. Sono di conseguenza definiti r*ate reale* per chiamata e *rate apparente*. Il primo considera unicamente l'occupazione di banda dovuta alla codifica e agli header, mentre il secondo considera quanta banda deve essere allocata a ogni chiamata in modo da rispettare i requisiti sui parametri end-to-end. Inoltre, è stato altresì evidenziato che l'efficienza real-time dipende direttamente dalla dimensione dei pacchetti trasportati sulla rete. A sua volta, la dimensione ottima dipende da parametri relativi alla singola chiamata, per esempio devono essere tenuti in considerazione il numero di salti (*hop*) attraversati dal flusso dati. Al crescere della dimensione dei pacchetti diminuisce l'efficienza real time in quanto si verificano ritardi di pacchettizzazione maggiori e, quindi, un maggiore *rate* apparente.
Analogamente, al crescere del numero di nodi attraversati dalla chiamata sono richieste maggiori risorse per poter rispettare i requisiti di QoS. Diminuisce di conseguenza la capacità dei link in termini di *Erlang*

(unità di misura del traffico telefonico: rappresenta il tempo di utilizzo continuativo, valutato in ore, di un canale voce). Se si vogliono, infatti, rispettare i vincoli temporali imposti in fase di progetto, nel caso aumenti il numero dei nodi da attraversare, si deve, ovviamente, diminuire il tempo che il pacchetto impiega per attraversare il singolo nodo e, quindi, anche in questo caso, aumenta il rate apparente del singolo flusso.

## 6. IMPATTO DELLA QOS SUL DIMENSIONAMENTO DELLA RETE

L'impatto dei requisiti in termini di ritardo end-to-end della chiamata sull'attività di dimensionamento delle capacità dei link della rete richiede un'analisi di tutti i fattori che influenzano il ritardo di trasmissione dei dati da sorgente a destinazione. In prima approssimazione, è possibile considerare come termini significativi i ritardi dovuti alle seguenti operazioni, effettuate dai terminali o dai nodi della rete IP: ritardo di Codifica/Decodifica, reti voce locali di origine e destinazione e *Core Network*.
Per quanto riguarda i ritardi introdotti in fase di codifica e di decodifica è conveniente utilizzare la stima proposta nell'ambito del progetto TIPHON [10], nella quale si evidenziano tre fattori: *ritardo di bufferizzazione del terminale IP* che riguarda, generalmente, ritardi introdotti dall'*hardware* del terminale utilizzato (per esempio, i buffer in ingresso ai convertitori A/D); *ritardo del codificatore* che comprende i tempi di elaborazione, di *look ahead* e di bufferizzazione di un frame di campioni su cui si esegue la codifica e *ritardo di pacchettizzazione* dovuto alla necessità di dover considerare la possibilità di inserire più di un frame generato dal codec in un solo pacchetto IP.
Ipotizzando trascurabile il primo, il progetto TIPHON suggerisce come stima del ritardo del codificatore la seguente formula:

$$\tau_c = 2f_s + \Delta_a + \Delta_{la}$$

dove $f_s$ rappresenta la durata del frame, $\Delta_a$ il ritardo di elaborazione dell'algoritmo (DSP) e $\Delta_{la}$ la componente look ahead (il fattore "look ahead" tiene conto del fatto che i codificatori, al fine di migliorare le prestazioni in termini di bit rate, considerano durante la codifica di un voice frame anche alcuni campioni del frame successivo).
Nel *decoder* si assume che vi sia ulteriore ritardo dovuto al tempo di elaborazione e all'uso di buffer di uscita. Il ritardo di elaborazione totale attraverso *encoder* e decoder ($2\,\Delta_a$) si assume sia sensibilmente inferiore alla lunghezza del buffer di uscita, solitamente scelto in modo da contenere un *voice frame*.
Questo conduce alla valutazione approssimativa del ritardo coder/decoder, nella quale si trascura il tempo di *processing* del DSP:

$$\tau_c = 3\,f_s + \Delta_{la}$$

Il ritardo di pacchettizzazione incide in modo direttamente proporzionale al numero di voice frame inclusi in ogni pacchetto IP: per ogni frame aggiuntivo si deve considerare un tempo di pacchettizzazione pari alla durata del frame.
La trattazione proseguirà ipotizzando una tipologia di rete dove sono indicati anche i vari tipi di ritardo sperimentati nelle diverse parti in cui viene suddivisa la rete. Per stimare il ritardo di accodamento dei pacchetti IP a ogni hop attraversato non è possibile utilizzare l'approccio tradizionalmente utilizzato nella telefonia, basato sui modelli markoviani o reti di code M/M/1 o più in generale sulla formula di Erlang, non essendo distribuito secondo un andamento poissoniano il tempo di interarrivo dei pacchetti al nodo IP. Al contrario, il traffico IP in ingresso a ogni nodo può essere considerato deterministico, in quanto i buffer di uscita dei codificatori tendono a generare un bit rate costante. Inoltre, assumendo di avere pacchetti a lunghezza costante, anche il traffico in uscita dal nodo in questione risulta essere di tipo deterministico, essendo uguale il tempo di smaltimento impiegato dal nodo per ogni pacchetto. Analizzato il ritardo dovuto alla codifica, si passa direttamente a definire i ritardi dovuti alla rete locale di partenza o di arrivo (*Originating/Terminating Network*): i due contributi possono es-

sere visti come simmetrici, e quindi il computo sul ritardo viene effettuato considerando una sola delle due tratte. Per avere il ritardo totale basta raddoppiare il risultato. Sostanzialmente, si possono individuare i seguenti contributi [10]:

- *Fixed Switching Delay*: è quello dovuto ai meccanismi di forwarding interni allo switch locale; può diventare significativo nel caso di switch lenti.
- *Variable Voice Contention Delay*: è il ritardo dovuto alla contesa per l'accesso alle risorse del link da parte dei pacchetti voce; poiché si tratta di pacchetti aventi stesse caratteristiche e priorità, può essere applicato il modello della teoria delle code relativo a un traffico a bit rate costante che condivide un'unica coda senza priorità;
- *Variable Data/Voice Contention Delay*: è il ritardo dovuto alla contesa fra i pacchetti voce e i pacchetti dati; poiché si ipotizza l'utilizzo di un meccanismo di code a priorità, questo caso si può verificare solo nel momento in cui un pacchetto dati abbia già impegnato il canale al momento dell'arrivo del pacchetto voce.
- *Fixed Serialization Wan Delay*: è il ritardo di trasmissione dei pacchetti sulla WAN che porta alla Core Network.

Per quanto concerne l'architettura di Core Network, in bibliografia esistono diversi modelli a diversa complessità: per una stima di primo livello è possibile scegliere un modello molto diffuso in cui si considerano significativi nei confronti del ritardo di trasmissione end-to-end solamente i nodi aventi *N* interfacce in ingresso concorrenti verso una o più interfacce di uscita. Nei casi in cui il nodo IP abbia una sola interfaccia in ingresso, si è supposto che la/e interfaccia/e di uscita e la capacità di processing siano state opportunamente dimensionate in modo da non generare ritardi di accodamento: ogni pacchetto in ingresso viene immediatamente servito e inoltrato sull'opportuna interfaccia di uscita. Considerato, quindi, un nodo IP isolato e analizzandone il comportamento in termini di ritardo di bufferizzazione, l'adozione di tali modelli porta a concludere che poiché solitamente le velocità dei link appartenenti alla Core-Network spaziano solitamente nella cosiddetta "larghissima banda", a meno di non trovarsi a coprire distanze tali da rendere significativo il ritardo dovuto alla propagazione dell'onda luminosa nelle fibre (un valore tipico è di circa 5 $\mu$s per km) oppure a effettuare un numero molto elevato di passi (hop), tale ritardo sarà trascurabile rispetto a quello dovuto alla periferia della rete e ai Codec.

## 7. CONCLUSIONI

Questa rapida carrellata sui benefici e, in parte, sulle problematiche legate all'introduzione della fonia sulle reti dati, con particolare enfasi sugli aspetti legati alle **QoS**, ha mostrato come la VoIP, intesa come tecnologia, permette riduzioni di costo in quanto utilizza l'infrastruttura di rete IP per la realizzazione delle tradizionali applicazioni di fonia: l'utilizzo di Internet per chiamate a lunga distanza e l'utilizzo di normali PC come PBX aziendali (*Computer Telephony Integration*). Utilizzando le nuove tecnologie disponibili per la codifica e la compressione della voce, in aggiunta a tecniche quali la compressione dei silenzi è possibile il trasporto di un numero di canali vocali maggiore rispetto al caso della telefonia classica. La convergenza di voce e dati permette la realizzazione di nuovi servizi realizzati secondo modelli funzionali tra cui PC-to-PC, PC-to-Phone, Phone-to-Phone, fax-to-fax, PC-to-fax, Web-to-fax, E-mail-to-fax. Infine, servizi quali IP-PBX, Web-call center, IP-call

La **QoS** end-to-end dipende da tutti i componenti coinvolti nell'attivazione e gestione della chiamata. L'interesse maggiore è rivolto agli aspetti di QoS associati ai componenti di tipo IP, essendo ormai consolidati tutti gli aspetti relativi alla QoS nelle reti telefoniche a commutazione di circuito. Tra i principali problemi che caratterizzano le reti telefoniche a circuito vi è l'eco. La criticità di tale fenomeno risulta accentuata in un contesto VoIP nel quale il ritardo introdotto dalla trasmissione della voce a pacchetto contribuisce a diminuire sensibilmente la tolleranza umana all'effetto eco. Il fenomeno è affrontato e risolto tramite l'introduzione di opportuni soppressori d'eco a livello di media gateway e, infatti, la maggior parte dei gateway disponibili sul mercato offrono questa importante funzione.

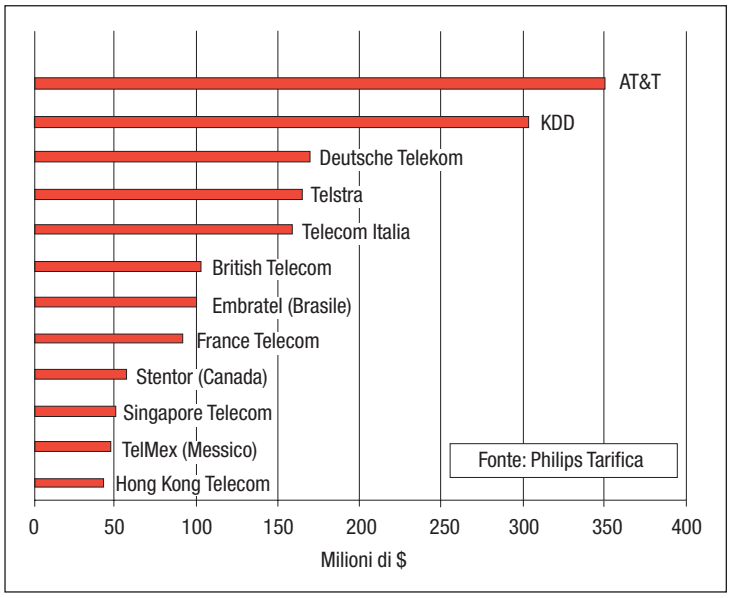



**FIGURA 7**
*Perdite sulle telefonate internazionali nel 2001*

center, *Internet Call Waiting*, fanno di VoIP una tecnologia che crea nuove fonti di fatturato per le imprese.
Per avere un'idea delle proporzioni del fenomeno in gioco, in termini di perdite economiche della telefonia classica si osservi la figura 7 che mostra la previsione fatta da Philips Tarifica riguardo le perdite sulle telefonate internazionali nel 2001 per alcuni tra i più importanti operatori telefonici mondiali. La previsione è stata ampiamente rispettata.
Se è vero che una rete IP senza il supporto della QoS non è in grado di fornire gli opportuni requisiti alle applicazioni real time, è anche vero che la QoS, e la sua gestione, possono avere un costo molto elevato in termini di risorse richieste alla rete, sia come capacità di banda che come capacità di elaborazione. È necessaria, pertanto, un'attenta pianificazione dell'utilizzo della QoS, che deve tenere conto della parte della architettura di riferimento che si sta trattando, della struttura fisica della rete, della velocità dei link e così via. È molto importante, ad esempio, valutare la capacità dei link su cui si vogliono configurare politiche di QoS. Il traffico voce ha una banda estremamente limitata, in genere si parla di poche decine di kbit/s per flusso voce, per contro, le attuali reti locali (almeno quelle basate su protocollo *Ethernet*) hanno velocità tipiche che partono dai 100 Mbit/s.
Anche considerando una rete locale che debba gestire 100 flussi voce contemporanei, considerando per ogni flusso una capacità di banda necessaria a livello *Data Link* di 40 kbit/s, che è un valore molto conservativo, si arriva a un valore di circa 4 Mbit/s, ovvero il 4% della capacità di banda disponibile, che è un valore del tutto marginale. Si nota, pertanto, che è molto più importante la corretta progettazione e realizzazione della rete, intesa come pianificazione dei domini broadcast, limitazione dei domini di collisione tramite un massiccio utilizzo di switch, ove possibile, e così via, piuttosto che cercare di trasmettere traffico real-time su un'architettura di rete mal configurata, imponendo pesanti sovrastrutture di QoS che potrebbero portare facilmente al collasso. Tipicamente, in ambito locale quello che si fa è utilizzare VLAN [15, 17] per suddividere la rete voce da quella dati, combinato con la creazione di code a stretta priorità che gestiscono il traffico real-time. Si utilizzano, poi, meccanismi quali WRED (*Weighted Random Early Detection*) per evitare per quanto possibile congestioni.
Il discorso cambia decisamente quando si passa dalla LAN alla WAN, dove al traffico locale si somma il traffico generato dalle altre LAN. Inoltre, normalmente, i collegamenti LAN-WAN sono a velocità molto più bassa, per cui l'eventuale traffico voce comincia a incidere in modo tutt'altro che marginale.
Nel caso di collegamenti a velocità inferiore ai 2 Mbit/s, è senz'altro utile prendere in considerazione strutture di QoS più stringenti, quali IP precedence o RSVP (*Resource Reservation Protocol*), l'utilizzo di code CB-WFQ (*Class Based Weighted Fair Queueing*) per una gestione ottimale della capacità di banda disponibile e, inoltre, tutti quegli accorgimenti utili ad aumentare l'efficienza della trasmissione, quali per esempio LFI (*Link Fragmentation and Interleaving*) e CRTP (*Compressed Real Time Protocol*). La qualità del servizio offerta al flusso voce è fortemente dipendente dal livello di saturazione della rete e, pertanto, l'utilizzo di politiche di QoS adeguate permette di avere una qualità costante, e rispondente alle specifiche desiderate, anche in condizioni di forte saturazione.

## Bibliografia


[1] AA.VV.: *Towards an Integrated Architecture for the Harmonization for PSTN and Internet Services*. Proceedings Actes – 6th International Conference on Intelligence in Networks, Gennaio 2000.

[2] Arora R: *Voice over IP*: Protocols and standards. 23 novembre 1999.

[3] Baldi M, Bergamasco D, Risso F: *On the Efficiency of Packet Telephony*. 7-th IFIP International Conference, on Telecommunication Systems, Nashville, Tennesee, USA, Marzo 1999.

[4] Baldi M, Bergamasco D, Guarene E: *Architectural choices for packet switched telephone networks*. International Switching Symposium (ISS '97), Settembre 1997.

[5] Braden R, Zhang L, Berson S, Herzog S, Jamin S: *Resource ReSerVation Protocol*. RFC 2205, IETF, Settembre 1997.

[6] Casner S, Jacobson V: *Compressing RTP/UDP/IP Headers for Low-Speed Serial Links*. RFC 2508, IETF, Febbraio 1999.

[7] Decina M: *Stato dell'Arte: Voice over IP Architetture e servizi*. Politecnico di Milano/Cefriel, Seminario Teach, Maggio 2000.

[8] ETSI Technical Report: *Design Guide for Elements of a TIPHON connection from an end-to-end speech transmission performance point of view*. Draft RTR 101 329-7, ETSI TIPHON Working Group 5, Febbraio 2001.

[9] Garavaglia E: *Mailing List Teach Innovazione*. www.teach.it

[10] Handley M, Schulzrinne H, Schooler E, Rosenberg J: *SIP: Session Iniziation Protocol*. RFC 2543, IETF, Marzo 1999.

[11] International Telecommunication Union (ITU-T): *Packet Based multimedia communication systems*. ITU-T Reccomendation H.323 Version 3, Maggio 1999.

[12] McPhillips E: *The Factors Affecting the Growth of VoIP*. HP Labs Technical Reports, Maggio 1999.

[13] Nichols K, Blake S, Baker F, Black D: *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. RFC 2474, IETF, Dicembre 1998.

[14] Pescapè A, De Petris R, Ventre G, Fadini B: *IP Telephony Bandwidth Calculator: a web-based application for network planning support*. AICA 2001, XXXIX Annual Congress p. 363-374.

[15] Pescapè A, Ventre G, Barone GB: V*lan Manager: An Architecture for Automatic VLAN design, configuration and management*. AICA 2000, XXXVIII Annual Congress, p. 327-342.

[16] Pescapè A, Esposito M, Romano SP, Ventre G: *A simulation model for bandwidth allocation of voice channels over IP networks*. ISCS 2001, p. 32-37 – ISBN 88 7146611-X. Cuen Editore.

[17] Pescapè A, D'Antonio S, Ventre G: *Un'applicazione per il Supporto del traffico real time su reti IP*. AICA 2002, XL Annual Congress p. 645-659.

[18] Pescapè A, Barone GB, Ventre G: *Traffico voce su reti IP: analisi e misure di protocollo*. AICA 2002, XL Annual Congress - p. 661-676.

[19] Pracht S: Factors in the success of Voice Quality in Converging Telephony and IP Networks. *International Engineering Consortium – Annual Review in Communications*, Vol. 53, section IV, Settembre 2000.

[20] Schulzrinne H, Casner S, Frederick R, Jacobson V: RTP: *A Transport Protocol for Real Time Applications*. RFC 1889, IETF, Gennaio 1996.



Bruno Fadini è Professore Ordinario di Calcolatori Elettronici presso la Facoltà di Ingegneria dell'Università di Napoli "Federico II". Tiene sin dal 1963 un corso dove tratta argomenti di architettura dei sistemi per l'elaborazione dell'informazione. È stato direttore del Dipartimento di Informatica e Sistemistica e Presidente del corso di Laurea in Ingegneria Elettronica della Facoltà di Ingegneria dell'Università di Napoli "Federico II" e presidente dell'AICA. Attualmente è direttore del CINI.
E-mail: fadini@unina.it

Antonio Pescapè è stato docente di Calcolatori Elettronici I, Elementi di Informatica e Reti Logiche presso la Facoltà di Ingegneria dell'Università di Napoli "Federico II". È membro del gruppo COMICS del GRID e attualmente frequenta il Dottorato di Ricerca in Ingegneria Informatica e Automatica presso il Dipartimento di Informatica e Sistemistica della Federico II di Napoli; svolge inoltre attività di ricerca presso il laboratorio ITEM del CINI di Napoli.
E-mail: pescape@unina.it.

Giorgio Ventre è Associato di Sistemi per l'Elaborazione dell'Informazione presso il Dipartimento di Informatica e Sistemistica dell'Università di Napoli Federico II. Dal 1990 al 1991 è stato ricercatore a contratto presso il Centro di Ricerca sul Calcolo Parallelo e i Supercalcolatori del CNR. Dal 1991 al 1993 è stato Post-doctoral fellow presso l'International Computer Science Institute di Berkeley. Dal 2000 è inoltre Direttore del Laboratorio Nazionale per l'Informatica e la Telematica Multimediali (ITEM) del Consorzio Interuniversitario Nazionale per l'Informatica.
E-mail: giorgio@unina.it

# LA TUTELA GIURIDICA DEI PROGRAMMI PER ELABORATORE

**Antonio Piva**
**David D'Agostini**

I programmi per elaboratore sono tutelati dal legislatore mediante il diritto d'autore, ossia il *copyright*, a cui si aggiungono norme speciali inerenti la copia di *backup*, la decompilazione e il *debugging*. Nell'articolo viene fornita una definizione giuridica della questione offrendone un quadro normativo; inoltre, sono illustrati i diritti riservati all'autore del programma e quelli garantiti all'utilizzatore; si passano in esame le sanzioni civili e penali predisposte per contrastare il fenomeno della dilagante pirateria; si conclude dedicando un paragrafo alla SIAE.

6

## 1. DEFINIZIONE DI SOFTWARE E POSSIBILI FORME DI TUTELA

L'Organizzazione Mondiale della Proprietà Intellettuale (OMPI), in inglese denominata WIPO (*World Intellectual Property Organization*), nel 1984 ha fornito una prima definizione giuridica di *software* quale espressione di un insieme organizzato e strutturato di istruzioni in qualsiasi forma o su qualunque supporto capace, direttamente o indirettamente, di far eseguire o far ottenere una funzione o un compito o far ottenere un risultato particolare per mezzo di un sistema di elaborazione elettronica dell'informazione.
Questa formulazione, volutamente ampia, ben si adatta al rapido evolversi della scienza informatica.
I giuristi italiani, in maniera più sintetica, per **programma** intendono "*un complesso di tutte le istruzioni necessarie a far eseguire al computer un determinato lavoro*".
Da queste definizioni deriva l'appartenenza del software ai beni giuridici immateriali, e in particolare alla categoria delle *creazioni intellettuali*.

I principi ispiratori della disciplina giuridica delle creazioni intellettuali mirano a contemperare due esigenze di segno diametralmente opposto.
***a.*** Promuovere lo sviluppo culturale e tecnologico, incentivando l'attività creativa dei privati mediante un riconoscimento tangibile all'autore o inventore. Vale a dire consentendo una soddisfacente possibilità di sfruttamento economico della creazione intellettuale, attraverso la concessione del cosiddetto *diritto di privativa*.
***b.*** Permettere nel contempo che tutti possano fruire del progresso raggiunto evitando che si creino stabili posizioni di monopolio culturale e tecnologico. Pertanto, si vuole escludere, rispetto a talune creazioni intellettuali particolarmente importanti per la so-

La definizione giuridica di **programma**: è "*complesso di tutte le istruzioni necessarie a far eseguire al computer un determinato lavoro*", ovvero "*sequenza di frasi univocamente interpretabili rivolte a un calcolatore affinché ponga in essere gli enunciati*" [2, 6].

cietà, la possibilità di sfruttamento esclusivo. Il software, dal punto di vista giuridico, è annoverato come creazione intellettuale e pertanto, nell'ordinamento italiano, la sua tutela giuridica viene ricondotta a 2 categorie:

- proteggibilità come *invenzione industriale* tutelata attraverso il brevetto;
- tutela tramite il *diritto d'autore* o copyright come opera dell'ingegno di carattere creativo.

Per quanto concerne le invenzioni industriali, ci si riferisce all'art. 2585 del Codice Civile (CC), che recita "*Possono costituire oggetto di brevetto le nuove invenzioni atte ad avere un'applicazione industriale, quali un metodo o un processo di lavorazione industriale, una macchina, uno strumento, un utensile o un dispositivo meccanico, un prodotto o un risultato industriale e l'applicazione tecnica di un principio scientifico, purché essa dia immediati risultati industriali*".

Per quanto attiene alla tutela del software tramite il diritto d'autore si cita l'art. 2577 del CC in virtù del quale "*formano oggetto del diritto di autore le opere dell'ingegno di carattere creativo che appartengono alle scienze, alla letteratura, alla musica, alle arti figurative, all'architettura, al teatro e alla cinematografia, qualunque ne sia il modo o la forma di espressione*". In questo caso ci si riferisce, per esempio, a libri, composizioni musicali, film e anche ai programmi per elaboratori.

Il brevetto e il diritto d'autore sono due modi differenti di garantire la tutela giuridica delle creazioni intellettuali. Implicano differenti modi di acquisizione dei diritti, differente durata del diritto e differenti strumenti di difesa da eventuali violazioni del diritto stesso.

In particolare, il brevetto permette lo sfruttamento della creazione con riguardo al suo contenuto (infatti, la moka serve a preparare il caffè indipendentemente dal suo *design*); il diritto d'autore, invece, protegge la forma dell'espressione creativa, a prescindere dal contenuto in essa racchiuso (pertanto, non rileva che un quadro raffiguri un paesaggio, una persona o una natura morta).

I diritti sull'invenzione industriale sorgono nel momento del conseguimento del brevetto, mentre i diritti d'autore sulle opere di ingegno sorgono nel momento stesso della creazione dell'opera [4].

## 2. IL QUADRO NORMATIVO

Il primo intervento legislativo in materia di software è stato di segno negativo in quanto il Decreto del Presidente della Repubblica (DPR) 22 giugno 1979, n.338 [12] non considera come invenzioni brevettabili i programmi per elaboratori "in quanto tali"; ciò non di meno il software può, comunque, essere una componente, a volte fondamentale, di un'invenzione industriale brevettata.

In ambito comunitario si afferma nel frattempo la necessità di adattare la legislazione sul diritto d'autore ai progressi della tecnologia perseguendo i seguenti obiettivi:

- armonizzare le legislazioni dei vari Stati membri;
- assicurare ai creatori di opere e alle imprese europee che le utilizzano una tutela che non li ponga in condizioni di svantaggio rispetto ai loro concorrenti di altri Paesi;
- ridurre l'effetto monopolistico del diritto d'autore, che conferisce una tutela troppo ampia e di durata eccessiva per questo tipo di opere.

Viene, inoltre, proposta una repressione più efficace della pirateria, mediante l'introduzione di pene più severe.

Si tratta, dunque, di assicurare una tutela flessibile che contemperi gli interessi contrapposti dei produttori, da una parte, e degli utilizzatori, dall'altra: il frutto di tali sforzi è la Direttiva CEE del 14 maggio 1991 n. 250 concernente i programmi per elaboratore espressi in qualsiasi forma, anche se incorporati nell'*hardware*, nonché i lavori preparatori di progettazione per realizzarli. Viene loro estesa la tutela riconosciuta dal diritto d'autore alle opere letterarie, determinando i diritti esclusivi dei quali i soggetti titolari possono avvalersi e la relativa durata.

Principi cardine di tale riforma sono i seguenti:

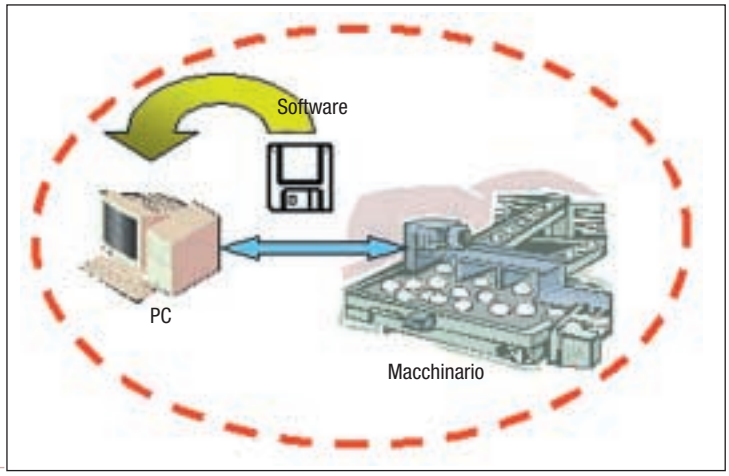
1. i programmi per elaboratori sono qualificati come opere letterarie e tutelati in base al diritto d'autore, purché originali;
2. la tutela non si estende ai principi, alla logica, agli algoritmi e al linguaggio di programmazione, nonché alle interfacce;

**3.** i diritti spettano a chi ha creato il programma: se questo è realizzato nel corso di lavoro subordinato o di un contratto d'opera spettano al datore di lavoro o al committente;
**4.** i diritti esclusivi comprendono: la riproduzione in qualsiasi forma, l'adattamento, la distribuzione sotto qualsiasi veste giuridica;
**5.** costituisce violazione dei diritti esclusivi il possesso e il commercio di copie illecite.
Il legislatore nazionale ha recepito tale direttiva mediante il Decreto Legislativo (DL) 29 dicembre 1992, n. 518 [12] che aggiorna la Legge 22 aprile 1943, n. 633, detta legge sul diritto d'autore (l.d.a.), che protegge l'espressione dei programmi, ma non le idee che ne stanno alla base.
Da ultimo, la Legge 18 agosto 2000, n. 248 [12] è nuovamente intervenuta sul punto con la dichiarata finalità di rafforzare la protezione del diritto d'autore. In realtà, i diritti e le facoltà spettanti all'autore di un'opera dell'ingegno sono rimasti quasi del tutto inalterati, tuttavia sono state inasprite le sanzioni penali.

## 3. LA TUTELA DEL SOFTWARE ATTRAVERSO IL BREVETTO

L'invenzione industriale è tutelata attraverso il brevetto e, nell'ordinamento italiano, ci si riferisce al testo delle disposizioni legislative in materia (Regio Decreto 29 giugno 1939, n. 1127, cosiddetta legge sulle invenzioni).
L'art. 12 della legge sulle invenzioni, modificato dal DPR n. 338 del 22 giugno 1979, che recepisce la Convenzione di Monaco sul Brevetto Europeo, dispone che "*non sono considerati come invenzioni... i programmi di elaboratori... in quanto tali*". Pertanto, non si considerano come invenzioni brevettabili i programmi per elaboratori intesi "in quanto tali".
Il divieto di brevettabilità del programma per elaboratore "in quanto tale" non esclude di fatto la tutela del software mediante il brevetto se inserito nell'ambito di una invenzione. Infatti, è possibile ottenere il brevetto di un'invenzione di combinazione[1] in cui si faccia uso di un elaboratore programmato in modo da ottenere un risultato tecnico nuovo, economicamente utile e non raggiunto prima. Pertanto, l'utilizzazione di un software quale componente di un prodotto non pregiudica di per sé la brevettabilità di un'invenzione purché sussistano i requisiti dell'applicabilità industriale, della novità, della liceità e dell'originalità (sentenza Cass. Civ. 14 Maggio 1981 n. 3169).
Brevettabile è quindi un'invenzione connessa da un software per esempio nella formula "*macchina comandata da un computer che attua un programma comprendente le seguenti funzioni...*" (Figura 1).
Pertanto, il software è tutelabile mediante il brevetto nel caso in cui non costituisca esso stesso l'oggetto dell'invenzione, ma sia uno strumento per raggiungere il risultato inventivo [8].
Per la tutela di un'invenzione industriale è necessario presentare domanda di brevetto presso l'*Ufficio Centrale Brevetti* (UCB). I diritti sull'invenzione sorgono nel momento del conseguimento del brevetto: in base all'art. 3 della legge sulle invenzioni, il brevetto dura vent'anni a decorrere dalla data di deposito della domanda e non può essere rinnovato né può esserne prorogata la durata.
La Commissione dell'Unione Europea ha recentemente formulato una nuova proposta di direttiva che estende la possibilità di brevettare il software, senza tuttavia importare il modello statunitense in cui è ammessa la brevettabilità di tutte le invenzioni attuate per mezzo di elaboratori elettronici, a prescindere dal requisito del contributo tecnico [13].

**FIGURA 1**
*Brevettabile è l'intero sistema*




[1] L'invenzione di combinazione realizza un risultato nuovo e originale tramite il coordinamento nuovo e originale di elementi e mezzi già conosciuti.

La proposta di direttiva definisce per la prima volta il concetto di "contributo tecnico" dell'invenzione-software, quale "*differenza tra l'oggetto delle rivendicazioni di brevetto* [...] *e lo stato dell'arte*". Per "invenzione attuata per mezzo di elaboratori elettronici", invece, si intende quella che, utilizzando una macchina programmabile, manifesti "caratteristiche di novità" ottenute anche solo parzialmente dalla macchina stessa.
La direttiva conferma le già note caratteristiche necessarie per il rilascio del brevetto, vale a dire: l'applicazione industriale, il carattere di novità, l'attività inventiva, con la precisazione che l'invenzione richieda un contributo tecnico.
La nuova tutela non pregiudica l'applicazione delle norme contenute nella direttiva 91/250/CEE relative al diritto d'autore, ma si affianca a quest'ultimo in un vero e proprio sistema a doppio binario.

## 4. LA TUTELA DEL SOFTWARE ATTRAVERSO IL DIRITTO D'AUTORE

La tutela del programma per elaboratore mediante il diritto d'autore, da un lato, protegge l'espressione dei programmi in una qualsiasi forma concreta, e cioè la forma espressiva, dall'altro esclude dalla tutela le idee e i principi che stanno alla base di un qualsiasi elemento del programma. Si è, pertanto, ritenuto di proteggere i singoli programmi senza arrestare il processo di innovazione tecnologica e scientifica, ossia consentendo la libera diffusione di idee, principi e algoritmi che stanno alla base del software. In base alla **legge sul diritto d'autore** vengono tutelati i programmi espressi in qualsiasi forma, purché originali e/o creativi, unitamente al materiale preparatorio per la progettazione dello stesso (art. 2 l.d.a.).
Non rientrano nella previsione normativa i principi che stanno alla base delle interfacce uomo-macchina del software (si pensi alla contrapposizione Apple e Microsoft riguardante l'interfaccia a icone dei propri sistemi operativi).

In base all'art 2 comma 8 della **legge sul diritto d'autore** sono protetti: "programmi per elaboratore, in qualsiasi forma espressi purché originali quale risultato di creazione intellettuale dell'autore. Restano esclusi dalla tutela accordata dalla presente legge le idee e i principi che stanno alla base di qualsiasi elemento di un programma, compresi quelli alla base delle sue interfacce. Il termine programma comprende anche il materiale preparatorio per la progettazione del programma stesso".

Il diritto d'autore sul software si divide in una componente morale, che rimane sempre attribuita all'autore del programma, e in una di natura patrimoniale che può essere oggetto di cessione.
Entrambe nascono in modo originario al momento della creazione del programma, come previsto dall'art. 6 l.d.a. senza alcuna formalità; peraltro chi voglia acquisire la certezza della data in cui il programma è venuto a esistenza può usufruire del servizio di deposito di inedito, tenuto presso la **SIAE** (*Società Italiana degli Autori ed Editori*): si tratta di un **registro pubblico** per i programmi per elaboratore con registrazione facoltativa (ai sensi dell' art. 103 l.d.a.) o di altre forme di prova documentale consentite dalla legge.

Per l'acquisto dei diritti relativi all'opera d'ingegno non è richiesta alcuna domanda: la registrazione dei software nel **registro pubblico** per i programmi per elaboratore tenuto presso la **SIAE** è facoltativa ai sensi dell'art 103 della legge sul diritto d'autore.

I *diritti morali* sono inalienabili, inespropriabili, non sottoponibili a condizioni o rinunce e comprendono:
1. diritto alla paternità dell'opera;
2. diritto di inedito;
3. diritto alla integrità dell'opera;
4. diritto di ritiro dell'opera per gravi ragioni morali.

La violazione del diritto morale costituisce illecito tutelabile quando la violazione del diritto morale è commessa insieme alla violazione di uno dei diritti di sfruttamento economico.
I *diritti patrimoniali* riguardanti l'opera di ingegno, intesi come diritti di utilizzazione economica in ogni sua forma e modo, consistono nel diritto di pubblicare l'opera, nel diritto di diffonderla, diritto di metterla in commercio, diritto di elaborarla, diritto di tradurla e diritto di cedere del tutto o in parte questi suoi diritti patrimoniali.
Per quanto riguarda i *diritti* esclusivi *patrimo-*

*niali* (elencati nell'art. 64 bis l.d.a.) il diritto d'autore sul software si discosta da quello classico in modo netto; nel diritto d'autore classico, infatti, nessuno può moltiplicare le copie, ma è libero di utilizzare il singolo esemplare (per esempio, la vendita del volume); inoltre, è prevista la possibilità di effettuare copie di un'opera per uso personale (art. 68, l.d.a.).
La disciplina riservata al software è ben diversa: all'autore è riservata la riproduzione, permanente e temporanea, totale o parziale, del programma per elaboratore con qualsiasi mezzo e in qualsiasi forma.
Sono soggette all'autorizzazione del titolare dei diritti anche il caricamento, la visualizzazione, l'esecuzione, la trasmissione o la memorizzazione se comportano una riproduzione del programma (lett. a) dell'art. 64 bis l.d.a.).
Nonostante un'improprietà terminologica da parte del legislatore nell'impiegare il medesimo vocabolo "riproduzione" per esprimere due diversi concetti (duplicazione e caricamento), con l'ovvio risultato di ingenerare confusione, non c'è dubbio sull'intenzione di limitare, oltre che la riproduzione, anche la sola utilizzazione non autorizzata del software.
Sono, inoltre, riservate la traduzione, l'adattamento, la trasformazione e ogni altra modificazione del programma per elaboratore, nonché la riproduzione dell'opera che ne risulti. Si ritiene che, mentre l'opera derivata possa essere utilizzata a fini personali da chi la realizza, salvo la subordinazione del diritto allo sfruttamento economico al consenso dell'autore dell'opera originaria, la stessa derivazione (per esempio, la modifica a un programma preesistente) non è consentita all'utilizzatore del software, senza il consenso dell'autore, stante il categorico divieto dell'art. 64 bis lett. b).
Infine, spetta all'autore qualsiasi forma di distribuzione al pubblico, compresa la locazione e la commercializzazione. Si ricorda che con la prima vendita di un programma in ambito comunitario viene esaurito il diritto di distribuzione, cioè non può essere impedita a terzi la successiva vendita di tale programma in tutta l'Unione Europea, sempre nel rispetto dei diritti dell'autore a cui vanno corrisposte le *royalty* (lett. c) dell'art. 64 bis l.d.a.) [11].
Si osserva che il trasferimento a terzi dei diritti d'autore patrimoniali deve essere provato per iscritto (art. 110 l.d.a) e può riguardare l'opera nel suo insieme o anche in parte. Inoltre, questi diritti possono essere trasferiti anche singolarmente (per esempio, la cessione del diritto di commercializzare l'opera ma non quello di modificarla).
Infine, la durata della tutela del diritto d'autore è di 70 anni dalla morte dell'autore.
Per quanto concerne i soggetti del diritto, generalmente si nega che la persona giuridica possa vantare un diritto d'autore, quantomeno a titolo originario, poiché tale disposizione della direttiva non è stata recepita nella normativa italiana; la giurisprudenza, però, considera ammissibile la titolarità in via derivata, in quanto i diritti si trasferirebbero automaticamente in capo alla persona giuridica in virtù dei contratti di lavoro o di opera (secondo il d.lgs 518/92, nel caso in cui l'autore sia un lavoratore dipendente, i diritti di utilizzazione economica del software spettano al datore di lavoro, a meno che non sia diversamente pattuito).
Invece, per quanto riguarda le opere collettive, costituite dalla riunione di più programmi o parti di essi (purché mantengano la loro individualità di creazione autonoma, per esempio i programmi personalizzati per i committenti che si avvalgono di programmi e parti di altri programmi preesistenti, adattati e collegati tra loro), soggetto del diritto d'autore è chi organizza l'opera compiendo un'attività di scelta, coordinamento e direzione della creazione (art. 7 l.d.a.).
Più frequente nella prassi è il caso di software frutto di un lavoro di gruppo in cui è impossibile individuare il contributo dei singoli programmatori: in proposito si applica l'art.10 l.d.a., che disciplinando la protezione da accordare alle opere create da più coautori prevede che il programma sia tutelato da una comunione di diritti morali e di sfruttamento economico dell'opera, disciplinata dagli artt. 1100 e seguenti del CC con qualche lieve correttivo.
Infine, per quanto concerne ordinamenti giuridici di derivazione anglosassone il diritto d'autore, o copyright, protegge l'opera,

ma non completamente l'autore in quanto vengono considerati i diritti patrimoniali ma non sono presi in considerazione quelli morali. Infatti, per esempio gli Stati Uniti hanno aderito alla convenzione di Berna con l'esplicita esenzione dal riconoscimento dei diritti morali; ciò significa che gli autori americani godono della tutela dei diritti morali in Europa, al contrario gli autori europei non godono di questi diritti negli Stati Uniti.

### 4.1. I diritti dell'utilizzatore

L'art. 64 ter l.d.a. ammette che siano consentite senza alcuna autorizzazione del titolare, le attività di riproduzione, nonché quelle di traduzione, adattamento, trasformazione e ogni altra modificazione del programma, a cui si aggiunge la correzione di errori, qualora la stessa sia posta in essere dal legittimo acquirente[2] e sia comunque necessaria per l'uso a cui il programma è destinato; in realtà, queste deroghe al diritto di esclusiva del titolare non sono vincolanti per l'autore del programma e, pertanto, hanno scarsa applicazione [1].

Salvo accordo contrario, non sono soggette all'autorizzazione del titolare dei diritti le attività indicate nell'art. 64 bis, lettere a) e b) (sopra indicate), allorché siano necessarie per l'uso del programma per elaboratore conformemente alla sua destinazione da parte del legittimo acquirente, inclusa la correzione degli errori (il debugging). Questa norma, dal contenuto palesemente dispositivo, appare come un cattivo esempio di tecnica legislativa: [11] anziché indicare con norme inderogabili quali siano le attività meritevoli dell'utilizzatore che l'esclusiva dell'autore non può assolutamente impedire, tale formulazione dispositiva rende le stesse più o meno legittime a seconda della volontà negoziale delle parti o, peggio, a seconda della loro forza contrattuale.

Il secondo comma dell'art. 64 ter è, invece, senz'altro una norma imperativa, attributiva di un diritto dell'acquirente e non di una mera facoltà contrattuale: non può essere impedito per contratto, a chi ha acquistato il diritto di usare una copia del programma per elaboratore di effettuare una copia di riserva del programma (la cosiddetta copia di backup), qualora sia necessaria per l'uso; ogni ulteriore duplicazione è senza dubbio vietata, a tutto vantaggio dell'esclusiva del titolare.

Il terzo comma, infine, conferisce a chi ha il diritto di usare una copia del programma le facoltà di osservare, studiare, sperimentare il funzionamento dello stesso durante le operazioni di caricamento, visualizzazione, esecuzione, trasmissione e memorizzazione del programma allo scopo di determinare i principi su cui è basato per una più corretta utilizzazione; ogni eventuale accordo contrario è da considerarsi nullo.

L'art. 64 quater disciplina il tema molto dibattuto della decompilazione (o *reverse engineering*), [9] procedimento che, consentendo di risalire dal programma oggetto/eseguibile al programma sorgente, è necessario al fine di sviluppare programmi che possano interagire tra loro e permettere prestazioni più complesse e complete (Figura 2).

Così la norma consente al soggetto legittimato di usare una copia del programma, di trarre tutte le informazioni necessarie ai fini dell'interoperabilità del programma (e solo per questa), senza l'autorizzazione del titolare dei diritti e a condizione, però, che le

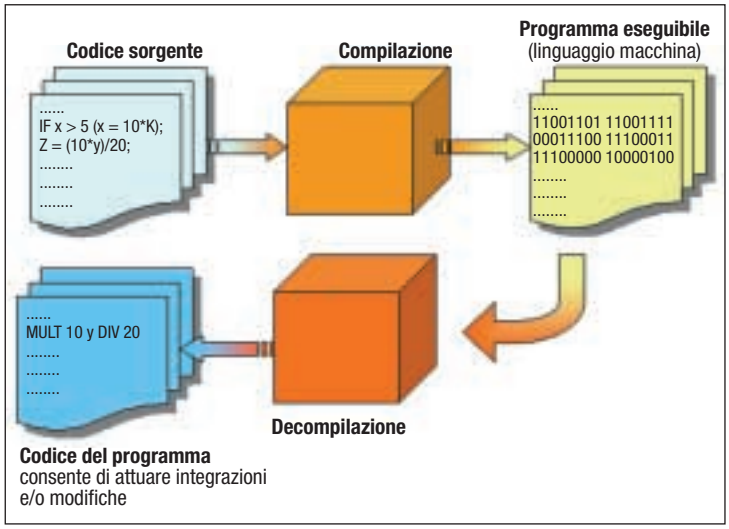



**FIGURA 2**
*La decompilazione è il percorrere a ritroso le fasi attraverso le quali viene creato il software*

[2] L'espressione "legittimo acquirente" va interpretata come "avente diritto", altrimenti la disposizione sarebbe applicabile solo nel caso di compravendita del programma e non anche nelle ipotesi di locazione, comodato o donazione.

stesse non siano diversamente disponibili.
La problematica della decompilazione si inquadra nell'esigenza di garantire una certa divulgazione delle informazioni tecniche contenute nei programmi per elaboratore, degli algoritmi che ne sono la base, della loro struttura, e in generale di tutte le conoscenze tecniche impiegate nello sviluppo dei programmi, in modo che chi ne venga a conoscenza sia messo in grado di impiegarle per programmi diversi (non sostanzialmente simili), al fine di favorire lo sviluppo tecnologico del settore.
È evidente che la conoscenza così ottenuta di nozioni tecniche è un presupposto essenziale per il progresso del settore e per un responsabile uso del programma. Per tale ragione, la disciplina brevettuale prevede addirittura il deposito dell'invenzione, che pertanto viene resa pubblica; i potenziali concorrenti così, pur non potendo utilizzare direttamente l'invenzione brevettata, potranno studiare l'invenzione per cercare soluzioni diverse, pervenendo eventualmente anche a un miglioramento dell'invenzione di partenza e contribuendo in tal modo al progresso tecnico. Attraverso questo sistema, quindi, si concilia la protezione dell'inventore con la libera circolazione delle informazioni contenute nelle creazioni protette, nell'interesse generale del progresso.
Nel campo della tutela dei programmi per elaboratore, la decompilazione può essere effettuata solo nel rispetto di precisi limiti:
- la decompilazione deve essere eseguita da chi ha il diritto di usare il *software* (art. 64 quater comma 1, lett. a);
- le informazioni non devono essere già "facilmente reperibili e rapidamente accessibili" (art. 64 quater comma 1, lett. b);
- la decompilazione deve essere limitata solo alle "parti del programma originale necessarie per conseguire l'interoperabilità" (art. 64 quater comma 1, lett. c);
- le informazioni acquisite non devono essere comunicate a terzi (art. 64 quater comma 2, lett. b);
- le informazioni acquisite non possono essere utilizzate per costruire programmi sostanzialmente simili nella loro forma espressiva (art. 64 quater comma 2, lett. c).

Infine, si osserva che, in base al terzo comma dell'art. 64 ter, il legittimo utilizzatore può, senza alcuna autorizzazione, osservare, studiare o sottoporre a prova il funzionamento del programma, allo scopo di determinare le idee e i principi sui quali è basato ogni suo elemento, purché tali atti siano compiuti durante le operazioni di caricamento, visualizzazione, esecuzione, trasmissione o memorizzazione del programma stesso. In realtà, ciò che la legge consente è solo l'osservazione dall'esterno del funzionamento del programma, che assai difficilmente può fornire al tecnico programmatore importanti indicazioni (la cosiddetta *black box analysis*) [9].
In definitiva, questa disciplina permette alle imprese di software più forti di proteggere efficacemente il proprio *know-how*, mediante l'adozione di pratiche anticoncorrenziali che vanno a discapito del progresso tecnico.

## 5. LE SANZIONI CIVILI

L'illecito di natura civile può essere contrattuale e consistere nell'inadempimento di una obbligazione, vale a dire in una violazione di clausole previste nei contratti di vendita e nelle licenze d'uso dei programmi per elaboratori. L'illecito può anche avere natura extracontrattuale, concretandosi nel danno ingiusto causato, per colpa o dolo, a un soggetto con cui non vi è legame negoziale (per esempio, nel caso di impiego di un programma senza avere acquistato il diritto di licenza d'uso). Chi causa un evento dannoso è tenuto al risarcimento dei danni patrimoniali previsto dall'art. 2043 CC e, in caso di illecito penale, anche di quelli non patrimoniali ai sensi dell'art. 2059 CC [5].
Secondo l'art. 158 l.d.a., chi ritiene che sia stato violato, o possa esserlo in futuro, un diritto di utilizzazione economica di un programma di cui è titolare può agire in giudizio per ottenere la rimozione dello stato di fatto da cui risulta la violazione, nonché il risarcimento del danno patito. Il d.lgs. 518/92 estende tutte le disposizioni riguardanti le sanzioni civili (dall'art. 156 all'art. 170 l.d.a.) a chi mette in circolazione in qualsiasi modo o semplicemente detiene per scopi commerciali copie illecite di software o utilizza mezzi

atti a rimuovere oppure a escludere dispositivi posti a protezione di programmi [3].
Tra i mezzi a disposizione dell'autorità giudiziaria, ai sensi dell'art.161 l.d.a., assumono importanza primaria, soprattutto nel corso del procedimento d'urgenza, l'accertamento, la perizia e il sequestro del materiale che il magistrato ritenga costituire violazione dei menzionati diritti.
La disciplina ex art. 100 l.d.a., d'altro canto, tutela i titoli dei programmi i quali, quando individuano l'opera, non possono essere riprodotti su altro *software* senza il consenso dell'autore; la norma è strettamente collegata all'art. 2598 CC, che si riferisce agli atti di imitazione servile idonei a creare confusione con i prodotti (si pensi anche al caso dei marchi registrati).
Quando, invece, la realizzazione di un programma è opera di un soggetto a conoscenza del know-how sottostante a un precedente prodotto perché aveva partecipato alla sua realizzazione, oppure perché era venuto in possesso delle informazioni rilevanti in occasione di pregressi rapporti con il titolare, gli istituti giuridici specificamente apprestati dall'ordinamento sono la tutela del segreto (artt. 2105 e 2598 n. 3 CC, art. 623 del *Codice Penale*, CP), la concorrenza sleale dell'ex dipendente (art. 2598 CC) e il patto di non concorrenza (art. 2125 CC) [10].
Nell'ipotesi in cui il fornitore decida di ritirare il prodotto dal mercato, risulta particolarmente delicato il problema di assicurare la continuità dell'assistenza e della funzionalità del software, anche in relazione al fatto che l'utente non è in grado di affidarne a terzi la manutenzione, non disponendo del codice sorgente.
Nel caso in cui le *software house* cedano solo il programma oggetto, riservandosi la proprietà e l'uso esclusivo del programma sorgente, è indispensabile, per l'uso corretto di tutto il sistema informativo, oltre all'assistenza per l'istruzione del personale anche la necessaria documentazione tecnico-operativa del programma: possibili rimedi sono il deposito fiduciario del programma sorgente presso terzi oppure il ricorso a provvedimenti cautelari di urgenza ex art. 700 del *Codice di Procedura Civile* (CPC) per il ripristino coatto dell'assistenza.

## 6. LE SANZIONI PENALI

La legge sul diritto d'autore, agli artt. 171 e seguenti l.d.a., disciplina le fattispecie di rilevanza penali, delitti perseguibili d'ufficio per i quali non è, quindi, necessaria la proposizione della querela da parte della persona offesa dal reato; è, pertanto, possibile che la notizia di reato emerga, come spesso avviene, nel corso di un'attività ispettiva della polizia giudiziaria che, riscontrando l'illecito, è poi obbligata a trasmettere gli atti alla procura della Repubblica.
Nel primo comma dell'art. 171 l.d.a. sono elencate le ipotesi criminose che attengono all'ambito patrimoniale, mentre nel secondo comma le stesse sono riferite alla sfera morale dell'autore: la violazione del diritto morale, pur non potendo configurare da sola un autonomo reato, costituisce circostanza aggravante della violazione patrimoniale. L'art. 171 bis l.d.a. prevede le fattispecie di duplicazione, importazione, distribuzione, vendita, detenzione a scopo commerciale, concessione in noleggio di copie non autorizzate di programmi per elaboratore e ricorre solo nel caso di condotte abusive e dolose in cui si ravvisi uno specifico fine delittuoso: il dolo specifico originariamente previsto era lo "scopo di lucro", concettualmente diverso dallo scopo commerciale richiesto dall'art. 161 per l'illecito civile; infatti per taluni la pena si intendeva applicabile solo se la riproduzione fosse stata realizzata al fine di trarre profitto, intendendo un concreto incremento patrimoniale positivo. Dopo la riforma del 2000 (legge n. 248/00, nuove norme di tutela del diritto d'autore, cd legge antipirateria), ai fini della punibilità della fattispecie è sufficiente che il soggetto ne abbia "tratto profitto", consistente in qualsiasi vantaggio o utilità economica, quindi anche nel risparmio di spesa; la nuova legge ha, quindi, imposto per esempio il divieto di copie di programma per trarne profitto, colpendo così penalmente anche le aziende che acquistano un numero di licenze software inferiori a quelle necessarie e pongono in circolazione copie "pirata" dei medesimi programmi.
Le pene previste sono notevolmente più pesanti, soprattutto se il fatto è di rilevante gra-

vità, come nel caso che oggetto della violazione sia un programma per elaboratore contrassegnato dalla SIAE e distribuito abusivamente (si può arrivare fino a 3 anni di reclusione e 15.000 € di multa; inoltre è disposta, quale pena accessoria obbligatoria, la pubblicazione della sentenza).
Tale severità è dovuta all'estrema facilità con cui è possibile duplicare un software, anche se, in realtà, spesso manca quasi la consapevolezza dell'illegalità commessa, specie nei casi in cui essa consista semplicemente nell'installazione del software sul proprio computer per uso personale o nell'utilizzo su più computer di un programma legalmente acquisito [3].
Il reato di cui all'art. 171 bis l.d.a. è plurioffensivo, cioè lede, oltre i diritti patrimoniali dell'autore, anche l'economia pubblica, su scala globale. Infatti, il volume d'affari generato da questo tipo di illecito è tale da alterare significativamente gli equilibri di un sistema economico; pertanto, la *ratio* della norma è di colpire il commercio di programmi duplicati abusivamente, perché è proprio tale condotta a provocare il danno economico e non solo l'impiego della copia in ambito privato o professionale.
L'art. 171 quater, introdotto nella l.d.a. a seguito di una specifica disciplina comunitaria[3], prevede l'irrogazione di sanzioni penali nei confronti di chiunque illecitamente concede a noleggio o in uso originali o copie di programmi, pur se originariamente ottenuti in maniera lecita; l'eventuale illiceità della riproduzione del supporto oggetto della concessione costituisce un altro, distinto e ulteriore reato perseguibile ai sensi dell'art. 171 bis l.d.a..
Nell'ipotesi in cui le medesime violazioni di cui all'art.171 l.d.a. abbiano natura colposa, cioè siano causate da negligenza, imprudenza o imperizia, si applica l'art.172 l.d.a. che stabilisce come sanzione per i trasgressori la sola pena pecuniaria.
Tutte le sanzioni previste negli articoli citati si applicano quando il fatto non costituisce più grave reato, punito direttamente dal codice penale o da altre leggi, in virtù della espressa clausola contenuta nell'art.173 l.d.a..

## 7. LA SIAE

Compito primario della SIAE, è l'intermediazione per l'esercizio dei diritti d'autore su mandato dei suoi aderenti, attività esercitata in via esclusiva ex art. 180 l.d.a.. Gli autori non sono obbligati a iscriversi potendo autonomamente avvalersi della protezione offerta dalla legge sul diritto d'autore; hanno, tuttavia, la facoltà di farsi rappresentare dalla SIAE nella tutela dei propri diritti (per esempio le azioni antipirateria).
Un'altra funzione fondamentale è la tenuta dei Pubblici Registri inerenti a formalità relative ai diritti di autori. I suoi compiti vanno al di là delle mere funzioni burocratiche o contabili: la SIAE, infatti, esercita nel campo della protezione delle opere dell'ingegno una pubblica funzione di generale tutela nell'interesse della collettività, cioè di tutti gli autori, iscritti e non, e di tutti gli utilizzatori. In questa veste difende principi generali, promuove studi e accordi che corrispondono a interessi collettivi e non dei suoi soli iscritti.
Anche per la tutela del software, come per le altre opere dell'ingegno, gli artt. 6 e 7 d.lgs. 518/92 prevedono un meccanismo di registrazione presso la SIAE, la quale si occupa di tutte le formalità della stessa.
Il deposito del software inedito, in particolare, è un contratto atipico che obbliga la SIAE a conservare per cinque anni, rinnovabili per un altro quinquennio, la copia di un programma non ancora reso pubblico, ricevendo un corrispettivo in denaro dal depositante; al fine di salvaguardare il diritto d'inedito tale istituto è caratterizzato dalla riservatezza, non potendo, infatti, essere comunicato a terzi alcun dato.
Il deposito d'inedito ha valore dichiarativo e non costitutivo, formando una prova documentale *in primis* dell'esistenza di un software a cui viene conferita data certa, secondariamente della paternità dello stesso. Non dà luogo a una presunzione legale

[3] Direttiva 92/100/CEE concernente il diritto di noleggio, il diritto di prestito e taluni diritti connessi al diritto d'autore in materia di proprietà intellettuale, attuata con d.lgs. 16 novembre 1994, n. 685.

di titolarità, ma costituisce comunque un principio di prova a cui possono aggiungersi altri mezzi istruttori: se un soggetto ha depositato un programma e un altro ne registra, una volta pubblicato, il prodotto finale dichiarandolo proprio, la presunzione opera a favore del secondo, ma il primo avrà comunque un mezzo per contestarne la validità [5].
L'art. 6 del d.lgs.518/92 ha stabilito che presso la SIAE venisse istituito un Registro pubblico speciale per i programmi per elaboratori, al fine di estendere loro il meccanismo di pubblicità legale a tutela dei diritti d'autore. Il Registro fa fede, fino a prova contraria, dell'esistenza del programma e della sua pubblicazione, evidenziando gli eventuali passaggi di proprietà; inoltre, gli autori indicati nel Registro sono reputati tali, sempre fino a prova contraria (Figura 3).
La registrazione non è obbligatoria e costituisce atto volontario del richiedente, pertanto il Registro non rappresenta un'anagrafe completa dei programmi ben potendo esistere opere protette dal diritto d'autore (che sorge dal momento della creazione dell'opera), pur se non registrate. La SIAE è obbligata a permettere la consultazione del Registro e a rilasciare estratti e copie autentiche degli atti detenuti: le dichiarazioni, le descrizioni e gli atti depositati; ovviamente rimane riservato il deposito dell'esemplare del programma.
Per poter registrare un programma è necessario farne richiesta depositando contestualmente la copia del programma riprodotta su un supporto ottico (CD-ROM), con l'indicazione del titolo del programma, del nome dell'autore e di altro titolare in via derivata (persona fisica o giuridica) dei diritti economici e, inoltre, della data e del luogo dell'avvenuta pubblicazione.
Soggetti legittimati a richiedere la registrazione sono l'autore del programma o il primo titolare dei diritti di utilizzazione economica che ha pubblicato il programma; il software proveniente da uno Stato membro dell'Unione Europea può essere registrato anche dal titolare dei diritti di utilizzazione economica legittimato a esercitarli in Italia. È consentito il deposito di programmi frutto di elaborazione, traduzione, adattamento anche se il programma originario non è stato registrato; i dati di quest'ultimo e di quello derivato devono essere comunque indicati nel momento della registrazione.
Inoltre, viene previsto che il richiedente presenti due esemplari delle etichette o dei frontespizi, al fine di una esatta individuazione del programma messo in circolazione. Secondo l'art. 8 del regolamento è considerato autore chi risulta indicato su tali etichette, mentre, in base all'art. 103 l.d.a., fa fede il nominativo iscritto nel Registro, pertanto è necessario che le due indicazioni siano conformi.
La registrazione del software è onerosa: la SIAE inserisce nel Registro i dati dichiarati e conserva nei suoi archivi, previa apposizione di un numero progressivo e della data, gli esemplari dei programmi, fornendo al richiedente un idoneo attestato di registrazione. La registrazione non è costitutiva del diritto, in conformità ai principi generali secondo cui la fonte dei diritti d'esclusiva è rappresentata dalla creazione dell'opera e non dall'espletamento delle formalità richieste dalla legge; tuttavia, mantiene importanza ai fini probatori introducendo una presunzione semplice dell'esistenza del di-

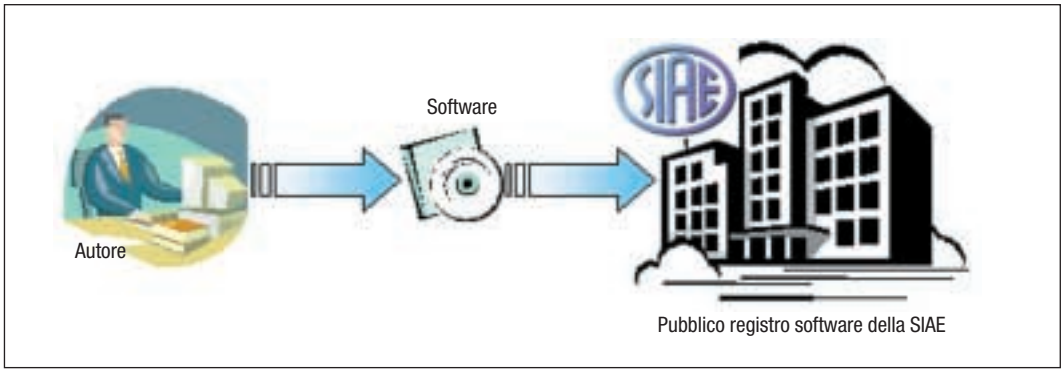



**FIGURA 3**
*La registrazione del programma presso la SIAE è facoltativa*


**FIGURA 4**
*Bollino della SIAE*

ritto di privativa nel soggetto che lo abbia registrato; infatti, prima della pubblicazione l'autore rischia di veder messi in discussione i diritti riconosciutigli dall'ordinamento se non è in grado di contrastare le illecite pretese di terzi dimostrando di essere l'autore di una determinata opera.
La SIAE, infine, è tenuta ad apporre sui supporti dei programmi un contrassegno, strumento usato dall'autorità pubblica nell'attività di certificazione o autenticazione secondo quanto disposto dall'art. 181 bis l.d.a. e dal successivo regolamento approvato con Decreto del Presidente del Consiglio dei Ministri 11 luglio 2001, n. 338; sono, pertanto, soggetti all'applicazione del cosiddetto "bollino" (Figura 4):
***a.*** i programmi aventi carattere di sistema operativo, applicazione o archivio di contenuti multimediali prodotti in serie;
***b.*** i programmi destinati alla lettura ed alla fruizione su apparati specifici per videogiochi, quali *playstation* o *consolle* comunque denominati, e altre applicazioni multimediali.
Fanno espressamente eccezione, e quindi sono esenti da contrassegno, alcuni casi tra i quali si ricordano, per importanza, quelli in cui i programmi per elaboratore ovvero multimediali vengano distribuiti:
***a*** come accessori nell'ambito della vendita di contratti di licenza d'uso multipli;
***b.*** gratuitamente dal produttore o con il suo consenso, in versione dimostrativa parziale;
***c.*** mediante *download* da *server* remoto e conseguente installazione purché detti programmi non vengano registrati a scopo di profitto in supporti diversi dall'elaboratore personale dell'utente, salva la copia privata;
***d.*** dal produttore al fine di far funzionare o gestire specifiche periferiche o interfacce (*driver*).

## 8. CONCLUSIONI

Da anni viene prefigurata la possibilità di una coesistenza del diritto d'autore e del brevetto [7], essendo il primo più adatto alla tutela della forma di espressione di un'opera, e il secondo rivolto al contenuto dell'idea inventiva; più che di coesistenza è corretto parlare di complementarietà delle due tutele.
L'intento del d.lgs. 518/92, e ancor di più della legge 248/00, è stato quello di combattere la dilagante pirateria in campo informatico indotta dalla estrema facilità e dal bassissimo costo con cui si può copiare un programma, di fronte alle ingenti spese e al tempo necessario per svilupparlo.
Si può, pertanto, attribuire al diritto d'autore il compito di impedire innanzitutto ed essenzialmente le attività di copiatura materiale, totale o parziale, del programma[4]; al diritto dei brevetti, con i suoi più restrittivi requisiti e oneri, viene demandata la tutela nei confronti di chi sviluppa programmi facendo uso di tecniche e invenzioni da altri ideati [7].
Un ulteriore caratteristica dell'approccio brevettualistico consiste nel contemperamento tra la pubblica utilità a conoscere il bene oggetto di tutela (al fine di favorire il progresso tecnico) e l'esclusiva in capo al suo inventore; nel sistema del copyright

[4] Vi rientra pure la cosiddetta "pirateria interna" ovvero la pratica assai diffusa di effettuare più copie di un singolo software acquistato legalmente per equipaggiare tutti gli elaboratori presenti in un ufficio.

questo rapporto è più favorevole all'autore, in quanto il software è generalmente distribuito sotto forma di codice oggetto, non intelligibile per l'utilizzatore. Nell'ottica brevettuale, invece, il rapporto tra conoscenza pubblica ed esclusiva dell'ideatore si dimostra maggiormente equilibrato, visto che per la concessione del brevetto è necessaria una descrizione chiara ed esaustiva dell'invenzione, che entra così a far parte del patrimonio scientifico della collettività.

## Bibliografia


[1] Alpa G: *La tutela giuridica del software*. Giuffrè Editore, Milano 1998.

[2] Borruso R: *Computer e diritto*. Giuffrè Editore, Milano 2001.

[3] Borruso R: *La tutela giuridica del software*. Giuffrè Editore, Milano 2001.

[4] Bosotti L, Jacobacci G: *I brevetti*. Torino, 1993.

[5] Chimienti L: *La tutela del software nel diritto d'autore*. Giuffrè Editore, Milano 2000.

[6] De Sanctis G: *La tutela giuridica del software tra brevetto e diritto d'autore*. Giuffrè, Milano, 2000.

[7] Dragotti G: Software, Brevetti e Copyright: le recenti esperienze statunitensi. In: *Riv. dir. ind.*, 1994, I.

[8] Finocchiaro G: *Diritto di internet*. Zanichelli, Bologna, 2001.

[9] Guglielmetti G: *Analisi e decompliazione dei programmi*. In: La legge sul software, a cura di C. Ubertazzi, Giuffrè Editore, Milano, 1994.

[10] Raffaelli: La contraffazione del software: profili di diritto d'autore e di concorrenza sleale. In: *Riv. Dir. Ind.*, 1995, I.

[11] Ristuccia, Zeno Zencovich: *Il software nella dottrina, nella giurisprudenza e nel D. L.gs. 518/92*. Cedam, Bologna, 1993.

[12] Tosi E: *Il codice del diritto dell'informatica e di internet*. La Tribuna, 2001.

[13] *The Economic Impact of Patentability of Computer Programs*. Reperibile all'URL http://europa.eu.int/comm/internal_market/en/indprop/studyintro.htm



Antonio Piva Laureato in Scienze dell'informazione, Presidente, per il Friuli - Venezia Giulia, e direttore responsabile della rivista di informatica giuridica ALSI (Associazione Nazionale Laureati in Scienze dell'Informazione ed Informatica). Docente a contratto di Informatica giuridica e di sistemi di qualità aziendali all'Università di Udine. Consulente sistemi informatici e valutatore di sistemi di qualità ISO9000.
E-mail: antonio_piva@libero.it

David D'Agostini Laureato in giurisprudenza, ha conseguito il master in informatica giuridica e diritto delle nuove tecnologie, lavora in uno studio legale fornendo assistenza e consulenza in materia di software, privacy e sicurezza, contratti informatici, commercio elettronico, computer crimes, firma digitale.
Ha rapporti di partnership con società del settore ITC nel Triveneto.
Collabora all'attività di ricerca scientifica dell'Università di Udine e di associazioni culturali.
E-mail: david.dagostini@adriacom.it