TECNICHE DI VIRTUALIZZAZIONE
TEORIA E PRATICA

Maurelio Boari
Simone Balboni

Negli ultimi anni l'interesse per il settore delle tecnologie delle macchine virtuali è cresciuto notevolmente. Si sono diffusi nuovi sistemi operativi per la virtualizzazione e si stanno sviluppando progetti per rendere più semplice ed efficiente il loro utilizzo. Un campo di applicazione interessante è la riorganizzazione di server farm di grandi dimensioni per rendere più efficiente l'uso delle risorse, semplificarne la gestione e aumentare la sicurezza. Questo articolo descrive le proprietà delle tecnologie di virtualizzazione e ne presenta l'utilizzo in un caso reale complesso.

3.2

1. INTRODUZIONE

Il concetto di virtualizzazione è da tempo ampiamente utilizzato in vari settori della computer science, dalla progettazione di sistemi software complessi (esempio, sistemi operativi), ai linguaggi di programmazione, alle architetture dei processori e alla trasmissione dei dati. Da un punto di vista generale le tecnologie di virtualizzazione puntano a disaccoppiare il funzionamento logico delle risorse hardware e software di un sistema di elaborazione dalla loro realizzazione fisica, con l'obiettivo di ottenere maggiore efficienza, affidabilità e sicurezza. Il disaccoppiamento è ottenuto introducendo tra le due viste della risorsa, la logica e la fisica, un livello di indirezione la cui realizzazione dipende dal tipo di virtualizzazione che si intende adottare.
Un primo esempio di virtualizzazione coincide con il concetto di astrazione. In questo caso l'obiettivo è semplificare l'uso di una risorsa nascondendo alcuni aspetti di dettaglio relativi alla sua realizzazione. Si parla in questo caso di risorsa virtuale (oggetto astratto) e il livello di indirezione introdotto è costituito dalle operazioni (interfaccia) con le quali è possibile accedere alle risorse.
Questo concetto di virtualizzazione viene normalmente applicato nella progettazione di sistemi di elaborazione complessi, che vengono organizzati come un insieme di livelli di astrazione strutturati gerarchicamente [9]. Al livello più esterno le applicazioni dispongono di una macchina virtuale il cui set di istruzioni è composto dalle istruzioni non privilegiate della macchina fisica e da un insieme di nuove istruzioni rappresentate dalle funzioni fornite dal S.O. (system call), mediante le quali è possibile accedere alle risorse del sistema in modo semplice e sicuro.
Il sistema di elaborazione è visto come un insieme di macchine virtuali, una per ogni processo attivo, che utilizzano tutte lo stesso livello di disaccoppiamento dalla macchina fisica rappresentato dall'interfaccia fornita dal S.O. Esse dipendono quindi dal S.O. di cui utilizzano le system call e dall'hardware sul quale sono eseguite.
Un caso diverso di macchina virtuale si presenta quando il livello di disaccoppiamento

dalla macchina fisica è rappresentato dal codice generato da un compilatore di un linguaggio del tipo HLL. Tale codice, chiamato codice astratto, risulta completamente indipendente dal set di istruzioni della macchina fisica e dalle system call del S.O. che in essa opera. Si parla in questo caso di macchina virtuale a livello di linguaggio. L'obiettivo di questa VM è permettere la portabilità dello stesso codice astratto su molteplici piattaforme (hardware e S.O.) diverse. Ciascuna di queste realizza una VM capace di caricare ed eseguire il codice astratto e un insieme di librerie specifiche. Nella sua forma più semplice, la VM contiene un interprete e, nei casi più sofisticati, un compilatore, che partendo dal codice astratto genera codice per la macchina fisica sulla quale la VM è in esecuzione. Un esempio ben noto di tale paradigma è la *Java Virtual Machine* (JVM) [6]. Lo sforzo di implementare per le architetture più comuni una VM capace di caricare ed eseguire il codice macchina astratto è ben ripagato dalla piena portabilità del software attraverso le piattaforme.
Un diverso utilizzo del concetto di virtualizzazione, prevede l'introduzione di un livello di indirezione, che si chiama *Virtual Machine Monitor* (VMM) o hypervisor il cui compito è quello di trasformare la singola interfaccia di macchina fisica in *N* interfacce virtuali. Ciascuna di queste interfacce (*macchine virtuali*) è una replica della macchina fisica dotata quindi di tutte le istruzioni del processore (sia privilegiate che non privilegiate) e delle risorse del sistema (memoria, dispositivi di I/O). Su ogni macchina virtuale può essere eseguito un sistema operativo. Compito del VMM è quindi consentire la condivisione da parte di più macchine virtuali di una singola piattaforma hardware. Esso si pone come mediatore unico nelle interazioni tra le macchine virtuali e l'hardware sottostante, garantendo un forte isolamento tra loro e la stabilità complessiva del sistema [5, 13].
Un primo esempio di tale tipo di architettura è quello introdotto da IBM negli anni '60 col sistema di elaborazione denominato originariamente CP/CMS e successivamente VM/370 [8]. Il CP (*Control Program*) svolge le funzioni del VMM, viene eseguito sulla macchina fisica ed ha il solo compito di creare più interfacce della stessa, senza fornire alcun servizio all'utente. Ciascuna di queste interfacce (macchine virtuali) è una replica del semplice hardware.
Il CMS (*Conversational Monitor System*) è il sistema operativo, interattivo e monoutente, che gira su ogni macchina virtuale.
Il CP/CMS è nato dal lavoro svolto presso il Centro Scientifico di IBM a Cambridge [3] a metà degli anni '60 con l'obiettivo di creare un sistema time-sharing. La sua adozione, nella versione VM/370, da parte di IBM fu una diretta conseguenza del sostanziale fallimento del sistema time-sharing TSS/360 costruito per il modello 67 del 360 che si rivelò non adeguato alle aspettative perché troppo complesso e poco efficiente.
L'idea architetturale delle macchine virtuali, propria del nuovo sistema, consentiva ad ogni utente, tramite il CP, di avere la propria macchina virtuale con la propria partizione sul disco e di supportare lo sviluppo dei suoi programmi in tale macchina virtuale tramite il CMS.
Tutta l'architettura risultava più semplice da gestire rispetto ad un tradizionale sistema time-sharing in quanto risultavano separate, e quindi sviluppabili indipendentemente, le due parti di suddivisione dell'uso delle risorse fisiche tra gli utenti e di mascheramento per l'utente delle peculiarità dell'hardware.
Inoltre, poiché ogni macchina virtuale era funzionalmente identica all'hardware della macchina fisica, era possibile mettere in esecuzione su di esse qualunque sistema operativo compatibile con l'hardware stesso e diverse macchine virtuali potevano eseguire differenti sistemi operativi. Nelle versioni successive del VM/370 furono messi in esecuzione sulle macchine virtuali diversi sistemi operativi, quali IBM OS/360 e DOS/360. Un altro aspetto di interesse era l'elevata affidabilità del sistema dal momento che la struttura del VMM consentiva l'esecuzione separata delle macchine virtuali garantendo quindi che un errore in un sistema operativo non avesse influenza sull'esecuzione degli altri S.O.
La diffusione in quegli anni di questo tipo di architettura portò anche alla progettazione di architetture hardware pensate per consentire in modo efficiente di soddisfare queste esigenze di virtualizzazione [19].
A partire dagli anni '70 si sono diffusi i mo-

derni sistemi operativi multitasking e contemporaneamente si è assistito ad un crollo nel costo dell'hardware. I mainframe hanno progressivamente lasciato il posto ai minicomputer e ad un paradigma del tipo "un server per ogni applicazione", e così lo sviluppo dei VMM si è interrotto al punto che le architetture non hanno più fornito l'hardware necessario per implementarli in modo efficiente. Questa tendenza ha però portato nei primi anni del 2000 ad una proliferazione dell'hardware all'interno delle sale server, con tanti minicomputer piuttosto sottoutilizzati, considerato il costante aumento della potenza di calcolo, con grossi problemi legati agli spazi, al condizionamento degli ambienti, elevati costi di alimentazione e di gestione. Per questi motivi nella seconda metà degli anni '90 sono ricomparsi sul mercato nuovi *Virtual Machine Monitor* compatibili con l'architettura più diffusa entro le sale server, ovvero IA-32, in un'ottica di consolidamento di tanti server sottoutilizzati su di un singolo calcolatore fisico. Questi sistemi saranno l'oggetto del prossimo paragrafo.
Per concludere, si può ricordare che un caso particolare di virtualizzazione si presenta quando si richiede l'esecuzione di programmi compilati per un certo insieme di istruzioni su un sistema di elaborazione dotato di un diverso insieme di istruzioni. Si parla in questo caso di emulazione.

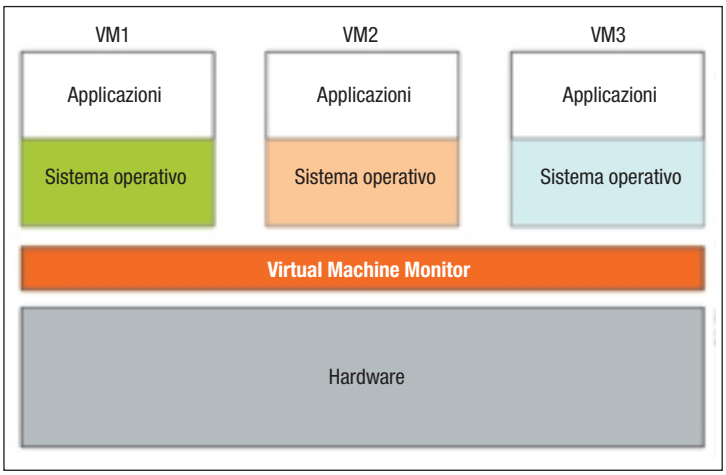


FIGURA 1

Schema logico di un ambiente virtualizzato: in questo esempio il VMM è posto direttamente sopra l'hardware, ed espone interfacce hardware virtuali funzionalmente identiche ad ognuna delle macchine virtuali in esecuzione sopra di esso

Il modo più diretto per emulare è interpretare: un programma interprete legge, decodifica ed esegue ogni singola istruzione utilizzando il nuovo set di istruzioni. È un metodo molto generale e potente, ma produce un sovraccarico mediamente molto elevato poiché possono essere necessarie decine di istruzioni dell'host per interpretare una singola istruzione sorgente.
Una famosa implementazione, molto cara agli appassionati di giochi, è MAME [15], una VM per host IA-32 in grado di caricare ed eseguire il codice binario originale delle ROM dei videogiochi da bar degli anni '80, emulando l'hardware tipico di quelle architetture. Facendo leva sull'enorme potenza di calcolo degli attuali PC rispetto ai primordiali processori per videogiochi dell'epoca, la VM può tranquillamente operare secondo il paradigma dell'interpretazione senza compromettere in modo significativo l'esperienza di gioco, almeno per i videogiochi più vecchi che non facevano ancora un uso massiccio della grafica. Sono già oltre 1900 le immagini di ROM che si possono eseguire con MAME su di un comune PC.
Un ulteriore esempio è costituito dalla classe dei microprocessori che, al fine di ridurre i consumi energetici, implementano internamente un'architettura a parole molto lunghe (VLIW); all'esterno invece, grazie ad un emulatore integrato in hardware, espongono una classica architettura IA-32, potendo così eseguire i più noti sistemi operativi e relative applicazioni. L'esempio più noto è il processore Transmeta Crusoe [21], tra i cui progettisti vi è anche Linus Torvalds.

2. REALIZZAZIONE DEL VMM

Come si è visto nel paragrafo precedente, compito del VMM è consentire la condivisione, da parte di più macchine virtuali, di una singola piattaforma hardware. Il VMM espone ad ogni macchina virtuale un insieme di interfacce hardware funzionalmente equivalenti a quelle di una macchina fisica e si pone come mediatore unico nelle interazioni tra le macchine virtuali e l'hardware sottostante, garantendo un forte isolamento tra esse e la stabilità complessiva del sistema (Figura 1).

Al di là dei modi diversi in cui si può progettare un VMM, esso deve comunque soddisfare poche condizioni essenziali: fornire un ambiente per i programmi sostanzialmente identico a quello della macchina reale; garantire una elevata efficienza nella esecuzione dei programmi; possedere caratteristiche di elevata semplicità. Il primo obiettivo richiede che qualsiasi programma eseguito all'interno di una macchina virtuale generi lo stesso risultato che si otterrebbe se il programma fosse eseguito direttamente sulla macchina reale. Le uniche differenze possono essere legate alle dipendenze temporali imposte dalla presenza delle altre macchine virtuali concorrenti e dalla disponibilità di risorse di sistema. Il secondo obiettivo, l'efficienza, richiede che l'overhead imposto dalla virtualizzazione sia comunque tale da offrire all'utente l'illusione di operare sulla macchina reale. Per ottenerlo occorre che un sottoinsieme statisticamente dominante delle istruzioni del processore virtuale siano eseguite direttamente – senza la mediazione del VMM – sul processore reale. Questo approccio, noto come *esecuzione diretta*, è centrale per potere realizzare efficacemente la virtualizzazione. In pratica esso prevede che le istruzioni *non* privilegiate, che sono la frazione più consistente di un *instruction set*, quelle da cui non può derivare l'eventuale blocco del sistema, siano eseguite direttamente in hardware senza coinvolgere in alcun modo il VMM. Quest'ultimo interviene invece solo nell'esecuzione delle istruzioni privilegiate, minimizzando così il sovraccarico. Infine, il requisito della semplicità nella realizzazione è essenziale per garantire la stabilità e la sicurezza dell'intero sistema, minimizzando l'occorrenza di malfunzionamenti che comprometterebbero l'esistenza delle macchine virtuali. In altre parole è necessario che il VMM, nonostante la disponibilità delle istruzioni privilegiate per le macchine virtuali, resti sempre nel pieno controllo delle risorse hardware e che non sia mai possibile ai programmi in esecuzione negli ambienti virtuali accedere all'hardware in modo privilegiato scavalcando il controllo del VMM.
Vi sono diversi modi per realizzare un VMM con queste proprietà; le differenze fondamentali tra le implementazioni più diffuse si possono ricondurre a due fattori discriminanti: i principi che governano il dialogo tra la macchina virtuale ed il VMM, ed il livello dove si intende collocare il VMM rispetto all'architettura del sistema di elaborazione. Rispetto alla prima scelta, che è la più importante in termini di metodo, si distingue tra i paradigmi di *virtualizzazione completa e paravirtualizzazione*. Rispetto alla seconda scelta, si distingue tra VMM posti direttamente sopra l'hardware dell'elaboratore (*VMM di sistema*) che integrano un sistema operativo "leggero" con le funzionalità di virtualizzazione e VMM che si istallano invece come applicazioni dentro un sistema operativo preesistente (*VMM ospitati*). Normalmente viene indicato con il termine *host* la piattaforma di base sulla quale si realizzano macchine virtuali, che comprende la macchina fisica, l'eventuale sistema operativo e il VMM; si indica invece con il termine *guest* tutto ciò (applicazioni e sistema operativo) che ha a che fare con la macchina virtuale.
Analizziamo ora i dettagli e le differenze delle diverse tecniche di realizzazione (Figura 2).
a. *Virtualizzazione completa*. Il paradigma della virtualizzazione completa prevede che l'hardware virtuale esposto dal VMM sia funzionalmente identico a quello della sottostante macchina fisica. In questo modo è possibile installare dentro le macchine virtuali sistemi operativi standard, senza che abbiano subito alcuna modifica specifica per eseguire in ambiente virtuale. Questo approccio semplifica notevolmente la creazione e gestione dell'ambiente guest, ma rende un po' più complesso il disegno del VMM; inoltre vedremo che per una efficace implementazione è richiesta la collaborazione dell'architettura della CPU. Negli approfondimenti che seguono supponiamo, per semplicità, che il VMM sia installato direttamente sull'hardware del calcolatore.
Un'architettura CPU in generale opera secondo livelli (ring) di protezione: per semplicità consideriamo due soli livelli, supervisore ed utente, anche se molte architetture, tra cui IA-32, implementano livelli intermedi usati ad esempio per l'I/O. Il livello supervisore è riservato al software che deve accedere alle risor-

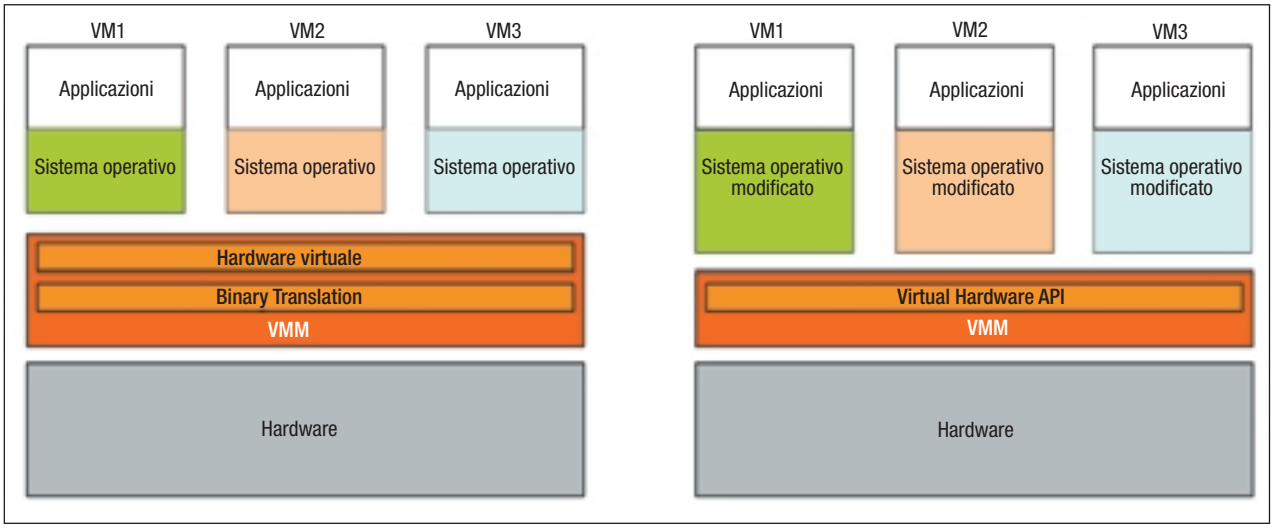


FIGURA 2

Confronto tra virtualizzazione completa (sinistra) e paravirtualizzazione (destra), su architetture IA-32. Un VMM per la virtualizzazione completa replica per ogni macchina virtuale le medesime interfacce hardware, funzionalmente identiche a quelle della sottostante macchina fisica; per questo i sistemi operativi guest non devono essere modificati. Un VMM in paravirtualizzazione espone invece una API cui i sistemi operativi guest devono interfacciarsi per accedere alle risorse

se del sistema con il massimo privilegio (sistema operativo, driver), e in tale stato si possono eseguire tutte le istruzioni proprie dell'architettura, tra cui anche le istruzioni privilegiate che danno pieno accesso alle risorse hardware e ai registri del sistema. Il livello utente è riservato al software meno privilegiato (applicazioni), e in tale stato non è possibile accedere alle istruzioni privilegiate della CPU. Se dallo stato utente si invocano istruzioni privilegiate si attiva il meccanismo di protezione della CPU che non esegue l'istruzione in questione ma notifica allo stato supervisore, mediante una eccezione (trap), la richiesta ricevuta e gli passa il controllo. Normalmente, alcune delle componenti di un sistema operativo (il kernel ed i driver) si aspettano di essere eseguite nello stato supervisore, poiché devono controllare l'hardware. In un contesto di virtualizzazione tuttavia, è solamente il VMM ad occupare lo stato supervisore, mentre tutti i software guest che vi girano sopra (applicazioni, ma anche sistemi operativi) sono spinti più in alto, nel livello utente, con i privilegi di semplici applicazioni. Vi sono dunque due ordini di problemi nella gestione di un sistema operativo guest che non si intende modificare ai fini della virtualizzazione: esso è chiamato ad operare in uno stato che non gli è proprio, poichè le chiamate di accesso alle risorse gli sono inibite (problema di ring deprivileging), inoltre esso è schiacciato nello stato utente insieme alle semplici applicazioni, con il problema di doversi proteggere da queste (problema di ring compression).
Il VMM deve dunque mascherare ai sistemi operativi guest la natura dello stato in cui sono eseguiti, facendosi carico di intercettare ogni richiesta di accesso privilegiato all'hardware ed *emulandone* il comportamento. Si tratta infatti di richieste che non possono essere eseguite direttamente, ma non possono nemmeno essere ignorate pena il malfunzionamento o il blocco del sistema guest, di cui si interromperebbe il normale flusso operazionale. Per intercettare tali chiamate il VMM è aiutato in modo determinante dalle funzionalità di protezione dell'architettura CPU: quando l'ambiente guest tenta di eseguire un'istruzione privilegiata, la CPU notifica un'eccezione al VMM (posto nello stato supervisore) e gli trasferisce il controllo. Il VMM verifica la correttezza dell'operazione richiesta e provvede ad emularne il comportamento atteso. Se per esempio, un guest tenta di eseguire l'istruzione privilegiata che disabilita gli interrupt, il VMM riceve la notifica di tale richiesta e ne emula il comportamento atteso, cioè sospende la consegna degli interrupt solamente a quella macchina

virtuale. Così facendo, la macchina virtuale prosegue secondo il normale flusso operazionale che seguirebbe in un ambiente reale ed il sistema rimane complessivamente stabile; se invece la richiesta della macchina virtuale fosse eseguita sul processore, sarebbero disabilitati tutti gli interrupt per tutti i sistemi e questo impedirebbe al VMM di riguadagnare il controllo della CPU.

Il meccanismo di notifica della CPU aiuta a mantenere piuttosto semplice il disegno del VMM, che in modo trasparente è chiamato ad intervenire solamente per mediare l'accesso alle risorse hardware, di cui per altro mantiene sempre il controllo. La soluzione è anche efficiente in quanto consente che tutte le istruzioni non privilegiate siano eseguite direttamente dall'hardware, senza alcuna supervisione da parte del VMM che non sarebbe di alcuna utilità ed introdurrebbe solo latenza.

Un'architettura CPU si dice "*naturalmente virtualizzabile*" se supporta l'invio di notifica allo stato supervisore *per ogni* istruzione privilegiata eseguita dallo stato utente. Un'architettura di questo tipo favorisce un disegno semplice del VMM e supporta nativamente la tecnica dell'esecuzione diretta, fondamentale per garantire prestazioni.

È necessario osservare tuttavia che non tutte le architetture sono naturalmente virtualizzabili e nel novero di queste vi è anche l'architettura IA-32, che pure oggi è al centro della rinascita della virtualizzazione: realizzata nell'epoca del boom del personal computer, non è stata affatto progettata tenendo presente le condizioni per una semplice virtualizzazione. Alcune istruzioni privilegiate di questa architettura se eseguite nello spazio utente *non* provocano un'interruzione da parte del meccanismo di protezione della CPU ma vengono semplicemente ignorate e non consentono quindi un trasparente intervento del VMM. Inoltre, vi sono istruzioni non privilegiate, dunque consentite liberamente anche nello spazio utente, che permettono di accedere in lettura ad alcuni registri di sistema le cui informazioni andrebbero però mascherate ad una macchina virtuale e la cui gestione dovrebbe essere riservata solo al VMM (problema di *ring aliasing*). Vi è per esempio un registro (code segment register) che segnala il livello di privilegio corrente e la cui lettura da parte di un sistema operativo guest – che come tale è eseguito nello spazio utente, che non gli sarebbe proprio – segnalerebbe al medesimo un'anomalia di collocazione.

Questa assenza di supporto da parte dell'hardware impone al VMM di implementare stratagemmi di varia natura per garantire il funzionamento della virtualizzazione completa; i problemi possono essere risolti, ma al prezzo di un aumento di complessità ed una riduzione delle prestazioni. Una soluzione tipica prevede che il VMM scansioni il codice prima di passarlo in esecuzione per impedire che blocchi contenenti queste istruzioni siano eseguiti direttamente. VMware [8], per esempio, implementa la tecnica della *fast binary translation* [16] per sostituire i blocchi contenenti simili istruzioni problematiche in blocchi equivalenti dal punto di vista funzionale e contenenti istruzioni per la notifica di eccezioni che favoriscono l'intervento del VMM; tali blocchi equivalenti sono passati poi in esecuzione diretta ed inoltre sono conservati in una *cache* apposita per riusi futuri, risparmiando il costo di ulteriori traslazioni. Questo processo di traslazione va applicato almeno all'intero kernel del guest OS, per essere certi che tutte le istruzioni privilegiate che non notificano eccezioni vengano intercettate e gestite.

b. *Paravirtualizzazione*. Il paradigma della paravirtualizzazione prevede che l'hardware virtuale esposto dal VMM sia funzional-

Virtualizzazione completa e paravirtualizzazione

La virtualizzazione completa prevede che il VMM esponga ad ogni macchina virtuale interfacce hardware simulate funzionalmente identiche alle corrispondenti interfacce fisiche: in questo modo è possibile installare dentro le macchine virtuali sistemi operativi standard, senza che abbiano subito alcuna modifica specifica per eseguire in ambiente virtuale. All'interno della macchina virtuale, tutte le istruzioni non privilegiate sono eseguite direttamente sul microprocessore del calcolatore (esecuzione diretta), ed il VMM si fa carico solamente di intercettare le richieste di accesso privilegiato all'hardware e ne *emula* il comportamento atteso.

La paravirtualizzazione prevede che il VMM esponga ad ogni macchina virtuale interfacce hardware simulate funzionalmente simili, ma non identiche, alle corrispondenti interfacce fisiche: piuttosto che emulare le periferiche hardware esistenti, il VMM espone una libreria di chiamate (Virtual Hardware API) che implementa una semplice astrazione delle periferiche. Occorre dunque modificare il kernel ed i driver dei sistemi operativi guest per renderli compatibili con la virtual hardware API del VMM utilizzato. La complicazione di dovere modificare i sistemi guest è però ripagata da una maggiore semplicità del VMM, che non deve preoccuparsi di intercettare in modo complicato accessi alle risorse hardware, ma si avvale della loro diretta collaborazione.

mente simile, ma non identico, a quello della sottostante macchina fisica. Piuttosto che emulare le periferiche hardware esistenti, il VMM espone una semplice astrazione delle periferiche. In particolare, l'interfaccia hardware virtuale che il VMM espone ai sistemi guest è ridisegnata nella forma di una *Applications Programming Interface* (*Virtual Hardware* API), che i sistemi guest devono sapere richiamare per guadagnare l'accesso alle risorse. Si richiede dunque che i sistemi operativi guest non siano più tenuti all'oscuro, ma siano consapevoli di operare in un ambiente virtuale. Evidentemente questo impone di modificarne il kernel ed i driver per renderli compatibili con le proprietà del VMM utilizzato; ma – ed è la cosa più importante – non occorre assolutamente modificare le applicazioni che girano sui sistemi operativi guest, perchè l'interfaccia tra le applicazioni ed il sistema operativo non viene toccata in alcun modo da questo approccio. Va dunque realizzato un porting dei sistemi operativi esistenti, poiché gli attuali non sono scritti per dialogare con una API di paravirtualizzazione ma solamente per gestire le interfacce hardware standard. Questo è certamente un problema, soprattutto per vecchi sistemi operativi di tipo legacy. Di riflesso però risulta enormemente semplificata la struttura del VMM, poiché esso non deve più preoccuparsi di individuare e catturare le operazioni privilegiate dei guest OS per poi emularle, ma si avvale invece della loro diretta e consapevole collaborazione.

Viene così meno il vincolo di operare con architetture CPU naturalmente virtualizzabili, non più essenziale per il funzionamento della paravirtualizzazione. Questo ha un impatto ancor più notevole nel contesto delle architetture IA-32 che non sono naturalmente virtualizzabili, e che – come poc'anzi visto – impongono al VMM l'implementazione di meccanismi complicati per prevenire anomalie. Il vantaggio maggiore di questo tipo di tecnica, rispetto alla precedente, consiste proprio nella maggiore semplicità ed efficienza di esecuzione del VMM.

Inoltre, vi sono contesti in cui la cooperazione tra il VMM ed il sistema guest è necessaria per raggiungere un risultato efficace: per esempio nell'ambito della gestione della memoria, si osserva che il VMM, come i tradizionali sistemi operativi, può fare uso della paginazione per spostare pagine di memoria dalla memoria primaria al disco, con il consueto vantaggio di potere allocare più memoria di quella strettamente disponibile. Ciò è particolarmente importante nel contesto della virtualizzazione, con molti sistemi e processi che insistono sulle stesse risorse sovra-allocate. Occorre dunque un meccanismo efficiente che consenta al VMM di reclamare ed ottenere in caso di necessità dalle diverse macchine virtuali le porzioni di memoria meno utilizzate. È chiaro che il sistema operativo di ogni macchina virtuale possiede informazioni relative a quali siano le pagine di memoria più adatte ad essere spostate su disco, decisamente migliori di quelle del VMM. Per esempio, un guest OS può notare che una pagina di memoria appartiene ad un processo terminato e dunque tale pagina non sarà più acceduta. Il VMM, operando al di fuori dei singoli guest OS, non può rendersi conto di questo e dunque non sarebbe altrettanto efficiente nel decidere quali pagine di memoria trasferire su disco per quella macchina. Una strategia comunemente adoperata per risolvere questo problema è di tipo "paravirtualizzante": su ogni macchina virtuale è in esecuzione un processo, a volte detto *balloon process* (processo aerostato) che comunica con il VMM. Quando vi è necessità di rastrellare memoria, il VMM chiede al balloon process di allocare più memoria, cioè di "gonfiarsi". Il guest OS, che meglio di tutti conosce l'utilizzo della memoria nell'ambito della macchina virtuale, seleziona le pagine da offrire al balloon process per soddisfarne la richiesta; il balloon process comunica tali pagine al VMM che ne rientra in possesso. La richiesta del balloon process provoca da parte del guest OS il trasferimento su disco delle pagine probabilmente meno utilizzate dalla macchina virtuale, con l'effetto netto di avere liberato memoria a favore del VMM, che provvede alla riallocazione. Anche VMM che per il resto operano secondo il paradigma della virtualizzazione completa, come lo stesso VMware, fanno largo uso di questi meccanismi tipici della paravirtualizzazio-

ne, in specifico nella gestione della memoria che – effettuata completamente dal di fuori – sarebbe altrimenti molto complessa e poco efficiente. In particolare, VMware offre l'opzione di installare dentro i sistemi guest un pacchetto di programmi (VMware Tools) in cui sono presenti questa e altre "sonde" per migliorare lo scambio di dati tra VMM e guest.
Nonostante la difficoltà di dovere realizzare il porting dei sistemi operativi esistenti, la paravirtualizzazione sta catalizzando un'attenzione sempre crescente. Il progetto attualmente più promettente di un VMM che opera secondo tale paradigma è XEN, un VMM open source sviluppato dall'Università di Cambridge [6]. Nell'ambito del progetto è stato realizzato il porting di Linux su XEN (XenoLinux), con un costo in termini di righe di codice del Kernel modificato per dialogare con la API del Virtual Hardware pari a circa 3000 righe, cioè circa 1,36% del totale. Un lavoro analogo, in collaborazione con Microsoft, è in corso per consentire il porting di Windows XP (XenoXP), ma il lavoro non è ancora terminato e risulta essere più che altro una sperimentazione. Attualmente lo sviluppo di XEN è alla versione 3; tra le funzionalità più rilevanti vi è il supporto per sistemi multiprocessore e per i più recenti kernel di Linux. Sono supportati inoltre meccanismi di "live migration" di VM da un host con XEN ad un altro: ciò permette di eseguire manutenzioni su di un singolo calcolatore senza interrompere i servizi, oppure bilanciare il carico complessivo tra i vari host con XEN di cui si dispone. Inoltre le principali distribuzioni di Linux commerciali (Suse e RedHat) offrono già i pacchetti precompilati per installare XEN, nonchè le immagini della propria distribuzione modificate per la paravirtualizzazione. XEN 3 supporta le istruzioni per la virtualizzazione che AMD ed Intel hanno di recente inserito nei propri processori basati su IA-32 e questo consente di eseguire sistemi guest Windows non modificati, in una modalità di virtualizzazione completa. In tal modo si sono superati i problemi legati al porting di sistemi guest proprietari. Sta infine crescendo l'offerta di strumenti di lavoro che permettono una gestione centralizzata di varie installazioni di XEN e delle relative macchine virtuali, da un'unica console di controllo, in analogia al prodotto VirtualCenter venduto da VMware. Il progetto più rilevante in tale ambito è Xensource, gestito dal gruppo storico che coordina lo sviluppo di XEN. Si tratta di una piattaforma commerciale per controllare più installazioni di XEN, con le relative macchine virtuali, da un'unica console centralizzata; grazie a questa piattaforma si possono compiere le operazioni fondamentali di gestione come installare un nuovo sistema guest, monitorare l'utilizzo delle risorse, modificare la configurazione di un guest, accederne la console ecc..

c. *VMM di sistema e VMM ospitati*. Soffermiamoci su un'ultima distinzione rilevante che caratterizza i sistemi di virtualizzazione: il livello dove si intende collocare il VMM rispetto all'architettura del sistema di elaborazione. Abbiamo già accennato che si distingue in questo caso tra VMM posti direttamente sopra l'hardware dell'elaboratore (*VMM di sistema*) e VMM che si istallano invece come applicazioni dentro un sistema operativo preesistente (*VMM ospitati*) (Figura 3). Nella prima opzione si integrano ad un sistema operativo "leggero" le funzionalità di virtualizzazione, in un unico sistema che esegue direttamente sopra l'hardware dell'elaboratore. Un'implementazione così a basso livello offre migliori prestazioni, anche se si rende necessario corredare il VMM di tutti i

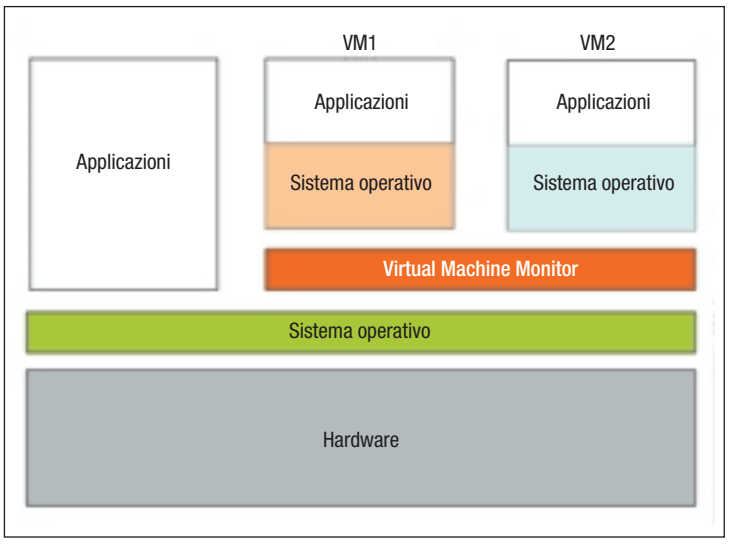


FIGURA 3
Schema puramente indicativo di un VMM ospitato su di un sistema operativo preesistente, con cui condivide l'accesso alle risorse

driver necessari per pilotare le periferiche. In generale i prodotti implementati secondo questo modello supportano un numero molto limitato di hardware certificato, rendendo meno impegnativa la gestione delle periferiche altrimenti molto difficile vista l'enorme varietà degli hardware nel mercato consumer. Un esempio di VMM di sistema è per esempio la versione ESX di VMware, non a caso quella più diffusa in ambiti professionali, o anche lo stesso XEN. Quest'ultimo adotta driver derivati dal kernel di Linux ottimizzati per offrire le migliori prestazioni sull'I/O in un contesto di virtualizzazione (varie VM in concorrenza); naturalmente tali driver sono compatibili con una lista molto ristretta di schede di rete e FibreChannel.

La seconda opzione prevede invece l'installazione del VMM come un'applicazione sopra un sistema operativo preesistente e non direttamente sull'hardware del calcolatore. Con un certo livello di approssimazione per non entrare troppo nei dettagli, si può dire che il VMM, anziché collocarsi sotto tutti gli altri livelli software, opera nello spazio utente e accede all'hardware attraverso le system call messe a disposizione dal sistema operativo su cui si installa. Le performance sono certamente inferiori, ma ci sono alcuni vantaggi importanti: il VMM è più semplice da installare poiché è come un'applicazione; inoltre potrà fare affidamento sul sistema operativo sottostante - sicuramente più fornito di driver per l'hardware più diffuso – per la gestione delle periferiche; infine, potrà servirsi di vari altri servizi dell'host OS come lo scheduling e la gestione delle risorse. È dunque una soluzione comoda, che sacrifica le performance ma semplifica il disegno del VMM ed è supportata da qualsiasi piattaforma IA-32 sulla quale sia già installabile un sistema operativo. Spesso tale opzione è più che sufficiente per un utente che abbia solo l'esigenza di avere contemporaneamente attivi sul proprio PC diversi sistemi operativi per sviluppare o testare applicazioni. In questo segmento, per le architetture x86, si trovano la versione gratuita di VMware Server (prima noto come GSX), Virtual Server di Microsoft, il software opensource User Mode Linux.

VMM di sistema e VMM ospitati

I VMM di sistema si installano direttamente sull'hardware del calcolatore, ed integrano le funzionalità di virtualizzazione ad un sistema operativo minimale. Un'implementazione a basso livello offre migliori prestazioni, anche se si rende necessario corredare il VMM di tutti i driver necessari per pilotare le periferiche. In generale i prodotti implementati secondo questo modello supportano un numero molto limitato di hardware certificato, risultando compatibili con un piccolo sottoinsieme dell'hardware presente nel mercato consumer.

I VMM ospitati si installano come un'applicazione all'interno di un sistema operativo preesistente e non direttamente sull'hardware del calcolatore. Con un certo livello di approssimazione si può dire che il VMM opera nello spazio utente e accede l'hardware attraverso le system call messe a disposizione dal sistema operativo su cui si installa. Le performance sono certamente inferiori, ma il VMM è più semplice da installare ed inoltre si affida al sistema operativo sottostante - sicuramente più fornito di driver per l'hardware più diffuso – per la gestione delle periferiche. Esso potrà inoltre utilizzare servizi dell'host OS come lo scheduling e la gestione delle risorse. È dunque una soluzione comoda, che sacrifica le performance ma semplifica il disegno del VMM.

3. VIRTUALIZZAZIONE E CONSOLIDAMENTO DEI SERVER ALL'UNIVERSITÀ DI BOLOGNA

3.1. Il contesto

L'Università di Bologna ha concentrato presso il proprio Centro Servizi Informatici d'Ateneo (CeSIA) la gestione della rete accademica ALMAnet e della Server Farm che ospita i sistemi per i servizi centralizzati: posta elettronica, DNS, file server, portale web d'Ateneo, sistemi per l'autenticazione, strumenti per il monitoraggio e la gestione della rete ALMAnet, le basi dati di personale e studenti, le applicazioni per le segreterie, servizi per docenti, studenti, e per l'amministrazione generale. All'inizio del 2005 ci si è posti il problema del rinnovamento della Server Farm, che aveva raggiunto dimensioni critiche: 200 server stand alone, Linux e Windows, ognuno con il proprio disco in locale, ognuno dedicato in modo esclusivo ad una singola applicazione o servizio, e come tale piuttosto sottoutilizzato. Un modello così rigido non era chiaramente in alcun modo scalabile e già così poneva grossi problemi sui costi di approvvigionamento e di manutenzione; vi erano inoltre problemi di ordine logistico come il condizionamento dell'ambiente e la distribuzione dell'energia elettrica. Ma, soprattutto, si osservava che la sola gestione ordinaria di un parco così ampio assorbiva completamente il personale preposto, che vedeva ridursi il tempo per la parte più qualificante del lavoro: lo sviluppo dei ser-

vizi a valore aggiunto che sono oggi considerati fondamentali, come la remotizzazione dei dati, le politiche di recupero da disastro e di continuità operativa (Figura 4).
Si è così stabilito di avviare un progetto di razionalizzazione della Server Farm secondo linee più moderne, con l'obiettivo primario della diminuzione del mero lavoro gestionale e dei costi di manutenzione, da attuarsi mediante una riduzione del parco macchine e concentrando più servizi sui medesimi sistemi (*consolidamento hardware ed applicativo*). Era poi richiesto che la nuova architettura supportasse in modo semplice e nativo funzionalità di recupero da disastro e di continuità operativa, da attuarsi con l'appoggio di un sito secondario a circa un chilometro di distanza dal CeSIA e connesso con esso tramite fibre ottiche, di cui l'Ateneo si era nel frattempo dotato. Si richiedeva inoltre il supporto per sistemi Windows e Linux, entrambi ampiamente utilizzati in Ateneo e la possibilità di sviluppare in modo più semplice, economico e standardizzato i servizi intorno ai server, in particolare backup/restore, monitoraggio, ridondanze.
Sono state studiate le tecnologie per il consolidamento ed acquisiti gli elementi di base per realizzare un'architettura consolidata: hardware scalabili che consentono di dimensionare la potenza di elaborazione e lo spazio su disco in modo dinamico al crescere della necessità, e sistemi operativi per la virtualizzazione, che consentono di partizionare le risorse hardware disponibili tra diversi sistemi che ne fanno un uso concorrente.
I calcolatori scelti sono quadriprocessori dualcore in architettura IA-32; essi accedono ai sottosistemi disco attraverso la rete Fibre-Channel, una tecnologia di trasporto ottimizzata per la comunicazione dei comandi SCSI.
Dei diversi VMM per architetture IA-32 è stato preferito VMware ESX per le seguenti ragioni: supporta sistemi operativi guest sia di tipo Windows che Linux; è dotato di una console di gestione centralizzata (VirtualCenter) che permette di tenere sotto controllo ed eseguire operazioni su tutte le macchine virtuali installate su qualsiasi calcolatore con VMware; dispone di meccanismi per la continuità operativa (Vmotion) che permettono la migrazione a caldo di una macchina virtuale

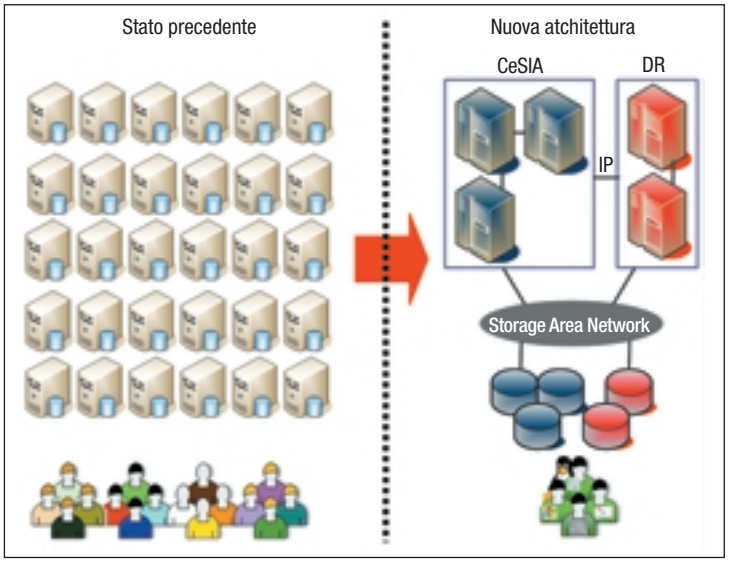


FIGURA 4

Modelli di gestione a confronto: rispetto ad un modello classico di gestione tipo "un server fisico per un'applicazione", si evidenza la modularità dell'architettura nella nuova infrastruttura, la semplicità nell'estendere su siti comunicanti la Server Farm per ottenere continuità operativa, il ridotto numero di persone necessarie al presidio

da un calcolatore con VMware ad un altro; dispone di driver per la gestione del più diffuso hardware professionale in commercio, con funzionalità avanzate per supportare la ridondanza dei percorsi di I/O; è dotato di un sistema molto raffinato di gestione della memoria (risorsa scarsa e costosa) che consente una forte sovra-allocazione di memoria alle VM rispetto alla memoria fisica presente.
Va comunque osservato che le funzionalità descritte non sono in alcun modo prerogativa del prodotto di VMware, ma con gradi di sviluppo diversi sono caratteristiche comuni di quasi tutti i sistemi di virtualizzazione commerciali ed opensource presenti sul mercato ([5, 6, 7, 12] e altri ancora). In particolare, ciò che ha permesso il cambiamento radicale nel modello di gestione della Server Farm è stata la combinazione di hardware scalabile di tipo "*commodity*" (cioè facilmente reperibile sul mercato, e indifferentemente da produttori diversi) unito al partizionamento delle risorse operato secondo principi molto simili da qualsiasi VMM per architetture IA-32. In particolare, è patrimonio comune di queste tecnologie operare il *multiplexing* delle risorse disponibili ed incapsulare un intero server in un solo file su

disco, azioni che sono la chiave di volta di tutte le funzionalità sopra descritte.

3.2. La soluzione

È nato così il nucleo della nuova Server Farm: in una tale architettura, una macchina virtuale è un processo in esecuzione su di un calcolatore, ed il suo disco è interamente incapsulato in un file posizionato sullo Storage Array. Tutti i nuovi server sono stati sistematicamente creati come VM all'interno di questa infrastruttura. I vecchi server sono stati migrati uno ad uno all'interno della nuova infrastruttura, alcuni ricostruendoli da zero in ambiente virtuale, altri trasferiti da fisico a virtuale mediante strumenti automatici che si occupano di ricostruire in una VM un server fisico ed il contenuto dei suoi dischi. Tali software sono spesso indicati come P2V, che sta per "Physical to Virtual".

Per supportare la continuità operativa, le reti di trasmissione dati sono state estese dal CeSIA verso il sito secondario mediante fibre ottiche di proprietà dell'Ateneo; nel sito secondario è stata predisposta un'uscita di backup verso Internet, e lì sono stati posti alcuni calcolatori con VMware ed uno dei due Storage Array acquistati. Attraverso la rete FibreChannel, i calcolatori in un sito vedono anche il disco che è nell'altro sito: così le macchine virtuali in esecuzione al CeSIA possono essere migrate a caldo – tramite Vmotion – anche sui calcolatori presenti nel sito secondario in pochi secondi, consentendo ad esempio di effettuare manutenzioni su un singolo calcolatore senza interrompere i servizi. Inoltre, in caso di rottura di un calcolatore nel sito primario i calcolatori del sito secondario possono intervenire riavviando i server caduti. Mediante meccanismi di replica sincrona e asincrona dei dati attuati tra i controller dei due Storage Array, è possibile mantenere sul sito remoto una copia allineata in tempo reale dei dati più critici del sito primario, per esempio le basi dati di personale e studenti. In modo analogo si può mantenere nel sito secondario una copia delle immagini disco delle macchine virtuali più importanti, per potere ripristinare i servizi nel sito secondario anche nel caso di un disastro che paralizzi totalmente il CeSIA.

Dopo un'attenta fase di test, si è compreso che era possibile concentrare fino a 40 VM su di un unico quadriprocessore (8 core) con 32 GB di RAM. La vecchia sala server di 200 macchine fisiche è stata così ristretta a soli 7 quadriprocessori, con risparmi ingenti dal punto di vista dei costi di gestione. Le operazioni quotidiane di presidio e gestione ordinaria si sono radicalmente semplificate poiché l'intera Server Farm è diventata controllabile da un unica interfaccia – VirtualCenter – attraverso cui è possibile accedere alle console dei server, monitorarne l'uso delle risorse, cambiarne a caldo la connettività di rete, attivare allarmi al superamento di soglie critiche, comandarne la migrazione a caldo nel caso occorra compiere manutenzioni sull'hardware sottostante. È inoltre diventato semplice aggiornare l'hardware virtuale di una VM: si può incrementare la RAM o il numero di processori, aggiungere spazio disco o ulteriori schede di rete nello spazio di un riavvio.

Un altro vantaggio fondamentale dell'ambiente virtuale è legato alla creazione di un nuovo server: si possono infatti confezionare dei *template* di installazioni tipiche (per esempio, una immagine di Windows 2003 con a bordo antivirus, agente di backup e altre applicazioni già configurate) ed eseguire installazioni di nuovi server virtuali a partire da essi, con un notevole risparmio di tempo ed energie. Si può anche creare un nuovo server clonando una macchina virtuale già in esecuzione, ottenendo in modo molto rapido un ambiente di test fedele all'originale per provare ad esempio l'effetto di aggiornamenti di sistema, variazioni di configurazione etc. Inoltre, liberarsi di questi ambienti è facile come cancellare un file: un vantaggio enor-

Virtual Appliance

Le Virtual Applicance sono macchine virtuali confezionate e configurate con a bordo tutto l'occorrente per svolgere funzioni applicative di un certo tipo, come web server, firewall, applicazioni di fonia su IP, ecc.. Esistono in rete repositori che contengono centinaia di immagini di Virtual Appliances divise per categorie e scaricabili gratuitamente. Ad oggi tali immagini sono eseguibili esclusivamente sul Virtual Machine Monitor per cui sono state create, poichè non esistono standard condivisi per il formato del disco delle VM o per le interfacce di paravirtualizzazione. Sono comunque già molto utilizzate per testare e prototipare rapidamente ambienti operativi.

me rispetto all'approccio classico che impone investimenti in acquisto di hardware dedicato, l'installazione del sistema operativo e delle applicazioni ecc. con pochissima flessibilità e costi elevati in termini di risorse umane e materiali.
La modularità dell'architettura, ad ogni suo livello, permette di acquisire autonomamente ed in stock le parti necessarie se le risorse per nuove VM scarseggiano: ulteriori calcolatori per aumentare la capacità di elaborazione, o ulteriori dischi per lo Storage Array. Si beneficia così di una riduzione dei costi sia in virtù di acquisti in lotto, sia in virtù della gestione centralizzata dell'hardware e della elevata standardizzazione delle componenti. Dal punto di vista organizzativo poche persone, esperte sui temi della virtualizzazione dei sistemi operativi, dei sistemi di calcolo e disco, delle reti, sono in grado di governare senza affanno un'architettura come quella descritta, poiché la quota di lavoro meramente gestionale si riduce in modo notevole, e non scala in modo lineare con il numero dei server.

Bibliografia

[1] Abramson, at al.: Intel Virtualization Technology for directed I/O. *Intel Technology Journal,* Vol. 10, Issue 3, 2006.

[2] Adair R.J., Bayles R.U., Comeau L.W., Creasy R.J.: *Virtual Machine for the 360/40*. IBM Corp., Cambridge Scientific Center, Report, n. 320-2007, May 1966.

[3] Adams K., Agesen O.: *A Comparison of Software and Hardware Techniques for x86 Virtualization*. ASPLOS'06 Conference Proceedings, October 21–25, 2006, San Jose, California, USA.

[4] AMD Corporation: *AMD64 Virtualization Codenamed "Pacifica" Technology: Secure Virtual Machine*. Architecture Reference Manual, May 2005.

[5] AA.VV.: *User Mode Linux*. 2006, http://user-mode-linux.sourceforge.net/

[6] Barham P., Dragovic B., Fraser K., Hand S., Harris T., Ho A., Neugebauer R., Pratt I., Warfield A.: *Xen and the Art of Virtualization*. Proc. of the 19-th ACM SIGOPS, October 2003.

[7] Bellard F.: *QEMU opensource process emulator*, 2006. http://fabrice.bellard.free.fr/qemu/

[8] Creasy R.J.: *The Origin of the VM/370 Time Sharing System*. IBM J., Research and Development, Semptember 1981.

[9] Dijkstra E.W.: The Structure of the Multiprogramming System. *Communication of the ACM*. Vol. 8, n. 9, 1968.

[10] Goldberg R.P.: *Survey of Virtual Machines*. Computer, June 1974.

[11] Goslin B., Steele G.: *The Java Language Specification*. Addison Wesley,1996.

[12] Microsoft Corporation: *Microsoft Virtual Server 2005 R2 Technical Overview, 2005*. http://www.microsoft.com/windowsserversystem/virtualserver/overview/vs2005tech.mspx

[13] Popek G.J, Goldberg R.P: Formal Requirements for Virtualizable Third Generation Architectures. *Communication of the ACM*, July 1974.

[14] Rosemblum M., Garfinkel T.: *Virtual Machine Monitors: Current Technology and Future Trends*. IEEE Computer, May 2005.

[15] Salmoria N.: *MAME, the Multiple Arcade Machine Emulator*. http://www.mame.net

[16] Sites R., et al.: Binary Translation. *Comm. ACM*, Febr. 1993.

[18] Sugerman J., Venkitachalam G., Beng-Hong Lim: *Virtualizing I/O devices on VMware Workstation Hosted Virtual Machine Monitor*. Proc.Usenix Annual Technical Conference., June 2001.

[17] Smith J.E, Ravi Nair: *The Architecture of Virtual Machines*. IEEE Computer, May 2005.

[19] Uhlig C., Neiger G., et altri : *Intel Virtualization Tecnology*. IEEE Computer, May 2005.

[20] Waldspunger C.: *Memory Resource management in VMware ESX Server*. ACM SIGOPS Operating Systems Rew, Winter 2002.

[21] Klaiber A.: *The technology beyind Crusoe processors: low-power x86-compatible processors implemented with code morphing software*. Tech. brief, Transmeta Corp., 2000.

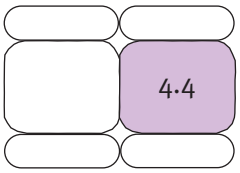
Simone Balboni è responsabile dei sistemi e servizi di rete presso il Centro Servizi Informatici d'Ateneo dell'Università di Bologna (CeSIA). Ha conseguito il dottorato di ricerca in Fisica computazionale ed è autore di alcuni articoli sul calcolo scientifico, sulle trasmissione in rete di dati multimediali, sulla sicurezza informatica.
E-mail: simone.balboni@unibo.it

Maurelio Boari è professore ordinario di calcolatori elettronici presso la Facoltà di Ingegneria dell'Università di Bologna. È autore di numerosi articoli scientifici ed alcuni libri. Ha interessi di ricerca nel settore dei sistemi distribuiti, linguaggi di programmazione e sistemi operativi. È attualmente delegato del Rettore per l'informatica e le reti di Ateneo.
E-mail: maurelio.boari@unibo.it

L'ETICA AL TEMPO DEI ROBOT

Giuseppe O. Longo

Il robot, unione di mente sintetica e di corpo sintetico, rappresenta l'ultima versione del nostro tentativo plurisecolare di costruire l'uomo artificiale. La somiglianza sempre più spinta tra robot e uomo, che si estende alle capacità cognitive, all'autonomia e in prospettiva anche alle emozioni e forse alla coscienza, pone interrogativi inquietanti. La crescente diffusione dei robot in tutti i settori della società ci obbliga a considerare il rapporto di convivenza uomo-macchina in termini inediti, che coinvolgono in primo luogo l'etica. Affrontare questi problemi è importante e urgente.



1. INTRODUZIONE

Forse, in realtà, stiamo assistendo a una graduale fusione della natura generale delle attività e delle funzioni umane con le attività e le funzioni di ciò che noi umani abbiamo costruito e di cui ci siamo circondati.

Philip Dick, *Mutazioni.*

Mentre l'evoluzione biologica ha dotato gli organismi viventi prima di un corpo e poi di un cervello, avente funzioni di controllo centrale e dotato in certi casi di proprietà cognitive superiori, non strettamente necessarie alla regolazione del corpo, l'intelligenza artificiale funzionalistica ha invece cercato di costruire una mente senza corpo, cioè un'intelligenza che imitasse le funzioni simboliche e astratte del cervello biologico evitando ogni interazione con un ambiente considerato fonte di disturbo. Tuttavia, le difficoltà di estendere questa forma d'intelligenza artificiale al di fuori dei domini simbolico-formali, hanno fatto ritenere che soltanto accoppiando la mente artificiale all'ambiente, attraverso un corpo artificiale dotato di sensi e di organi attuatori, si potesse ottenere un'intelligenza flessibile e ad ampio spettro come è quella biologica.

Il recupero della dimensione corporea e sensoriale ha portato ai robot e ha aperto una serie di interrogativi che vanno dagli aspetti tecnici della loro costruzione fino a sottili questioni di natura etica. Infatti, il robot è un artefatto capace di apprendere e dotato di una certa autonomia di decisione e comportamento e queste caratteristiche, in una prospettiva di stretta convivenza uomo-robot, non possono non sollevare certe domande come: fino a che punto siamo disposti a convivere coi robot, ad affidarci a loro nella vita quotidiana, nell'accudimento e nelle cure?

Se i robot dovessero un giorno diventare intelligenti e sensibili (quasi) quanto gli umani, potremmo continuare a considerarli macchine, come le lavatrici o le automobili? O dovremmo adottare atteggiamenti empatici e comprensivi come nei confronti degli animali domestici? Dovremmo arrivare a conferire loro dignità etica?

E viceversa: quali comportamenti dei robot dovremmo tollerare, incoraggiare o vietare? E di chi sarebbero le responsabilità di un loro eventuale comportamento dannoso?
L'ultima domanda è importante perché rivela il conflitto tra la natura artificiale dei robot, che dovrebbe renderli obbedienti alla nostra programmazione e la loro parziale autonomia (se un robot non è autonomo non è un robot) che, in linea di principio, potrebbe indurli a decisioni nocive nei nostri confronti. Il conflitto diventa drammatico nel caso dei "robot soldati". Erano problemi di questo genere che aveva in mente Asimov quando postulò le "Leggi della robotica", che vietano ai robot di compiere azioni dannose per gli esseri umani e che costituiscono il primo embrione di un'etica dei robot o, con un espressivo neologismo, di una "roboetica".
In questo ambito le previsioni si mescolano facilmente con la fantascienza e accanto alle speculazioni ci sono le realtà: in Giappone (il Paese di gran lunga più avanzato nella costruzione e nell'impiego dei robot) si tocca con mano quanto possa diventare intenso il rapporto uomo-macchina quando il robot sia un (o una) "badante" con sembianze umane oppure quando abbia più o meno le fattezze e il comportamento di un animale domestico (si pensi ad Aibo, il robot cane della Sony, ormai fuori produzione, che per anni ha svolto la funzione di "animale" da compagnia, Figura 1). La proiezione affettiva è tanto forte da suscitare problemi psicologici e, ancora una volta, etici. E poi, in generale, la marcia sempre più convulsa di una tecnologia invasiva e onnipresente non può non avere effetti profondi sull'immagine che abbiamo di noi stessi e sul nostro stesso essere "umani": specchiandoci in quello straniante *alter ego* che sta diventando il robot, quale immagine ce ne ritorna? Riusciremo, per differenza o per similarità, a capire qualcosa di più di noi stessi? Che questi problemi siano importanti e urgenti, è confermato dall'istituzione di un Comitato tecnico per la roboetica in senso alla *Robotics and Automation Society* dell'*IEEE*.
Nei paragrafi che seguono, dopo un breve inquadramento storico che descrive in particolare il passaggio dall'intelligenza artificiale funzionalistica alla robotica, sottolineando l'importanza del corpo sotto il profilo cognitivo e attivo, si considerano i problemi etici sollevati dalla presenza sempre più diffusa dei robot. Tali problemi sono acuiti dalla somiglianza crescente che presentano con gli umani, oggi sul piano cognitivo e attivo e, domani, forse, anche sul piano emotivo e della coscienza.

FIGURA 1
Il cane robot Aibo, di cui la Sony ha prodotto 150.000 esemplari tra il 1999 e il 2006

2. UNA STORIA MILLENARIA

L'impresa della robotica si colloca nel solco di un tentativo millenario, quello di imitare l'atto divino della creazione. Più o meno dichiarata, questa ambizione risale all'antichità biblica e classica, e la leggenda del Golem ne è forse l'esempio mitologico e letterario più noto (Figura 2). In questa impresa si intrecciano la vertigine del creatore e il timore per la creatura, che talora minaccia di ribellarsi e distruggere l'inesperto demiurgo. Anche nel caso del mostro di Frankenstein (Figura 3) la creatura trascende il progetto e si rivolta, suscitando negli uomini angoscia e terrore.
Talvolta gli umani subiscono invece il fascino degli esseri artificiali: nei racconti di Hoffmann gli uomini si innamorano perdutamente di bambole meccaniche, imitazioni perfette della donna, in cui la differenza tra il modello e la sua riproduzione si attenua fino a scomparire, inducendo in inganno anche l'osservatore più attento. Invece, per l'imperizia del costruttore, il mostro di Frankenstein è segnato da una diversità

che suscita orrore perché è interpretata come segno di malvagità.
Questi temi - orgoglio e timore, fascino e orrore - sembrano appartenere a un passato ormai lontano, eppure a ben guardare sono ancora presenti non solo nelle opere di fantascienza, ma anche nell'immaginario collettivo e nel nostro atteggiamento nei confronti delle tecnologie di punta, in particolare delle "tecnologie della mente" come i computer, l'intelligenza artificiale e specialmente i robot. Ciò sembra confermare il profondo substrato mitopoietico ed emotivo che ha sempre accompagnato l'attività tecnologica e la nostra interazione con la macchina.
Accanto ai miti e ai racconti, l'ambizione di costruire l'uomo artificiale produsse nei secoli una fioritura di opere artigianali, gli *automi*, artefatti spesso zoomorfi o antropomorfi, che, mossi dalla forza idraulica, dalla gravità o da un meccanismo nascosto al loro interno, sembrano comportarsi come esseri viventi. Col tempo l'elemento meraviglioso e ludico fu sostituito dalle finalità pratiche: non si trattò più di costruire macchine che tentassero di compiere le mille diverse azioni di *un unico* uomo, ma al contrario di ottenere una macchina che compisse *un'unica* azione, però sostituendo *mille* uomini. È il passaggio dall'androide elegante e variopinto alla nera e possente macchina a vapore.
Gli automi, raffinati e suggestivi prodotti dell'ingegno umano, oggi non si costruiscono più e sono rimpiazzati dovunque, se non nei musei e nei teatri della nostalgia, dai robot, manufatti in cui la tecnologia punta sempre più all'efficienza e sempre meno all'imitazione puntuale della natura. Eppure gli automi, specie quelli antropomorfi, cioè gli androidi e le andreidi, continuano a popolare di inquiete proiezioni e torbidi sogni la dimensione immaginaria del nostro tempo e da qui, soffusi di suggestioni mitologiche, travalicano nelle creazioni artistiche (letterarie e cinematografiche) e nelle attuazioni tecniche (Figura 4). Tanto che anche la robotica si confronta con la costruzione di macchine antropomorfe, gli *umanoidi*, residuo di una storia affascinante e tenebrosa di meccanica onirica, dove magia e occultismo s'intrecciano con la genialità inventiva, in un turbinio di personaggi eterogenei, in-

FIGURA 2
Il Golem, costruito dall'uomo per farsi aiutare, a volte si ribella al suo creatore

FIGURA 3
Il mostro di Frankenstein nella celebre maschera cinematografica di Boris Karloff

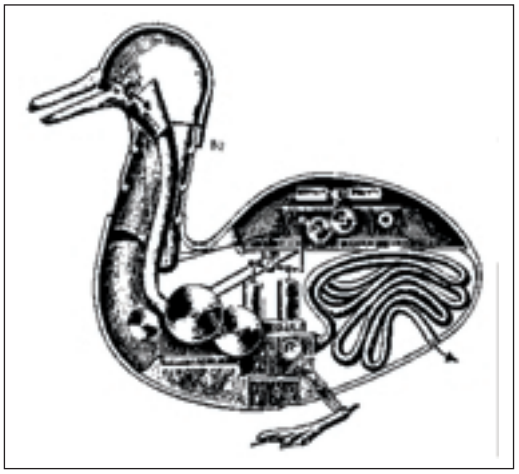


FIGURA 4
L'anatra digerente, uno dei celebri automi di Jacques de Vaucanson (1709-1782)

ventori, maghi, affaristi, ciurmadori, studiosi, prestidigitatori e creduloni.

3. DAL GOLEM AL FUNZIONALISMO

Per quanto stupefacenti, i prodotti artigianali del passato restavano comunque lontanissimi dal modello, uomo o animale, cui li avvicinava soltanto la forma esteriore ma non certo una puntuale somiglianza strutturale e funzionale. Le cose cambiarono radicalmente grazie alle ricerche stimolate dalla seconda guerra mondiale nel campo dei calcolatori e delle telecomunicazioni. Ben presto si capì che il calcolatore, lungi dall'essere una semplice macchina per far di conto, possedeva enormi capacità simboliche, tanto che nel 1956 nacque ufficialmente una nuova disciplina, cui fu dato il nome, un po' infelice per la verità, e fonte di equivoci durevoli, di *intelligenza artificiale* (IA) e il calcolatore divenne il modello di elezione della mente umana.
In fondo si trattava ancora della vecchia ambizione di imitare l'atto divino della creazione, ma non più con l'ingenuo e inarrivabile intento di costruire una creatura simile all'uomo nel suo complesso, magari con qualche approssimazione, bensì di riprodurre o simulare con estrema precisione una sola parte dell'uomo: la sua mente. Il sogno cominciava dunque a diventare realtà, sia pure limitatamente a un aspetto. Ma si trattava dell'aspetto più importante, caratteristico e fondamentale dell'uomo: l'intelligenza. Infatti, a quei tempi c'era (e c'è tuttora) una forte tendenza a identificare l'intelligenza con i suoi aspetti razionali, anzi simbolici e algoritmici, e questa identificazione, cui aveva contribuito potentemente il calcolatore, aveva a sua volta rafforzato la convinzione che l'informatica fosse la tecnologia giusta per costruire, dopo tante ingenuità, modelli della mente che fossero corretti e collaudabili.
Giungeva dunque a compimento un lunghissimo percorso, che dalla figura leggendaria del Golem portava, attraverso i mirabili e delicati automi, fino alla macchina ideale di Turing e ai prototipi concettuali di Von Neumann, capaci di riprodurre le funzioni nobili della mente. L'idea secondo cui, tutte le "funzioni nobili" della mente rientrassero nelle possibilità di replicazione della macchina, per un verso restava avvolta nelle ambiguità definitorie e per un altro diveniva oggetto di una congettura, la tesi di Church, secondo la quale tutta l'attività mentale dell'uomo è di tipo algoritmico, dunque è riproducibile con una macchina discreta. Questa congettura, non dimostrata e non dimostrabile, fu accettata da molti e pose le premesse teoriche e la giustificazione filosofica della versione *forte* dell'IA. Secondo questa versione, è possibile trasferire, senza perdite e senza distorsioni, da una struttura (cervello) a un'altra (computer) la funzione (cioè i programmi e gli algoritmi), che è la vera essenza dell'intelligenza. Si parla perciò di funzionalismo.

4. L'ALTRA METÀ DEL ROBOT: IL CORPO

Dopo i primi lusinghieri successi, anche i sostenitori più ferventi del funzionalismo dovettero riconoscerne i limiti, che derivano dalla natura disincarnata della mente artificiale, cioè dall'assenza di un *corpo* che comunichi con l'ambiente. Se l'intento era quello di simulare l'intelligenza umana, il *riduzionismo mentalista* dell'IA funzionalista ne trascurava un elemento essenziale.
L'intelligenza umana (e animale) si costituisce e si manifesta attraverso il corpo. L'intelligenza è un insieme di caratteristiche e attività fortemente sistemiche, oltre che fortemente diacroniche, cioè evolutive. In particolare, l'intelligenza nasce, si sviluppa e si manifesta attraverso la *comunicazione*, cioè lo scambio di messaggi di vari tipi, entro vari contesti, in vari codici e a vari livelli. Poiché la nostra "interfaccia" con il resto del mondo è costituita dal corpo e dagli strumenti tecnologici che abbiamo via via creato e perfezionato e che del corpo sono un potenziamento e un'estensione, è chiaro che proprio al corpo spetta il compito determinante di consentire la comunicazione e di filtrarla, sia in ingresso sia in uscita.
Riconosciuto il limite essenziale del funzionalismo e proseguendo sulla strada dell'imitazione della natura, si trattava di dotare il cervello artificiale di un corpo artificiale: questa strada portò alla robotica. Alla base di questa svolta c'è il riconoscimento della funzione conoscitiva del corpo.

Il sistema o macchinario conoscitivo individuale ha due modalità essenziali di funzionamento. La prima, più arcaica sotto il profilo sia filogenetico (della specie) sia ontogenetico (dell'individuo), è la conoscenza tacita, globale e immediata attuata dal corpo, nella sua struttura e nelle sue funzioni biologiche, e guidata dal sistema affettivo ed emotivo. La seconda, più recente sotto il profilo evolutivo e posteriore nello sviluppo dell'individuo, è la conoscenza esplicita, attuata nelle forme verbali e della razionalità. La prima è una conoscenza che si attua nel corpo e tramite il corpo, la seconda si attua nella mente o tramite la mente.
Orbene, la storia della cultura occidentale, in particolare della scienza, è in fondo un lungo tentativo di trasferire le conoscenze dalla prima alla seconda modalità, cioè dalla conoscenza biologica incarnata nel corpo (corpo che a sua volta è immerso nell'ambiente) a una razionalità disincarnata. In altre parole si vorrebbe tradurre nello scarnificato linguaggio astratto della mente (in particolare nel simbolismo della matematica) le rigogliose strutture del corpo e in genere della realtà; di rendere cioè esplicito, consapevole e leggibile ciò che è implicito, inconsapevole e oscuro. Questo tentativo è culminato nell'impostazione funzionalista o fisico-simbolica dell'IA. Ma fino a che punto è possibile questo trasferimento? All'inizio si riteneva che tutte le conoscenze fossero trasferibili, ma dopo i primi entusiasmi sono venute le delusioni e oggi ci si rende conto che, per replicare compiutamente l'intelligenza umana (ammesso che sia questo lo scopo dell'IA), anche le macchine intelligenti non possono fare a meno dell'equivalente di un corpo con tutta la sua attività cognitiva profonda e in parte forse non algoritmica: l'intelligenza disincarnata è troppo fragile e limitata.
Insomma, il tentativo di tradurre in conoscenza alta, razionale ed esplicita la massa delle conoscenze materiali, corporee e implicite incappa nell'ostacolo tipico di ogni processo di traduzione, cioè l'*incompletezza*. Rimane sempre un residuo ostinato e ribelle, che non si può tradurre.
Abbiamo così giustificato in termini epistemologici il passaggio dall'IA funzionalistica all'IA incorporata nel robot. Questo passaggio si oppone in qualche misura a una lunga tradizione filosofica. Da Platone in poi la modalità di conoscenza razionale è stata considerata superiore a quella corporea e tutta la corrente filosofica dominante si inscrive in questa ottica. Nel solco della filosofia razionalistica, anche l'IA funzionalistica considera la conoscenza astratta più nobile di quella legata al senso comune: l'intelligenza che dimostra un teorema sarebbe superiore a quella che riconosce una scena o che ci guida nelle azioni quotidiane. Ma questa lunga tradizione oggi viene messa in discussione. Addirittura si assiste a un capovolgimento: si riconosce che la maggior parte delle conoscenze, specie quelle vitali, sulle quali poggiano e dalle quali scaturiscono tutte le altre, sono espresse nella struttura stessa del corpo. A sua volta il corpo è immerso in un ambiente il quale, con le sue continue perturbazioni, non cessa di arricchire e aggiornare quelle conoscenze.
Il futuro della robotica più ambiziosa, quella che mira alla costruzione di macchine dotate di intelligenza, emozioni e forse coscienza, potrebbe dunque dipendere dalla comprensione del significato cognitivo delle azioni semplici, incarnate e contestualizzate che compiamo di continuo nella vita di tutti i giorni. Le descrizioni e gli strumenti usati finora in IA sono "alti e deboli": occorre integrarli con descrizioni e strumenti "bassi e forti", che riflettano e riproducano il nostro sfuggente "esserci nel mondo".

5. L'URGENZA DI UNA RIFLESSIONE

Per le considerazioni che intendo svolgere sui robot, è utile prendere spunto dalla letteratura fantascientifica, che costituisce un importante laboratorio di scenari suscettibili di trasformarsi in realtà, se non nei particolari, almeno nei tratti generali. Nel caso dei robot la letteratura e la filmografia sono ricchissime di spunti. La psicologia e la sociologia dei robot, degli androidi e dei ciborg (o *cyborg*, all'inglese) sono uno dei temi più interessanti della fantascienza moderna e si annunciano come uno dei settori più problematici di un futuro già a portata di mano nell'ambito della robotica. A cominciare dal dramma *R.U.R.* di

R.U.R.

Il termine "robot" (dal vocabolo ceco "robota", ossia lavoro pesante, sfacchinata) fu introdotto nel 1920 dallo scrittore ceco Karel Čapek nel suo dramma "R.U.R." (*Rossum's Universal Robots*, rappresentato il 25 gennaio 1921 al Národní divadlo di Praga) per indicare una macchina antropomorfa progettata e costruita dall'ingegner Rossum (da un'altra radice slava che significa "intelligenza") per alleviare le fatiche degli umani. Nel dramma si ritrovano molti dei temi relativi al rapporto uomo-robot: la compassione di Helena, che li ritiene infelici e vorrebbe promuoverne il riscatto dotandoli di anima; il realismo di Domin che li considera semplici macchine, prive di ogni sensibilità e destinate a servire indefessamente gli umani; il cinismo di Gall, che le vorrebbe capaci di soffrire per aumentare il loro rendimento; la ripugnanza di Nana, che vede in loro l'opera del demonio; gli effetti perversi della loro laboriosità, che porta gli uomini ad affogare nell'ozio e le donne a non partorire più; il loro impiego militare contro gli operai in rivolta per aver perso il lavoro. Per opera degli scienziati, i robot progrediscono e diventano sempre più intelligenti, superando gli uomini. Quando se ne rendono conto, i robot di tutto il mondo si ribellano ed eliminano la razza umana per assumere il potere, però così facendo si condannano alla scomparsa perché senza gli uomini non sanno riprodursi. Ma due robot di tipo specialissimo, maschio e femmina, hanno ricevuto dai costruttori scomparsi la capacità di amare e di procreare e danno origine a una nuova stirpe.

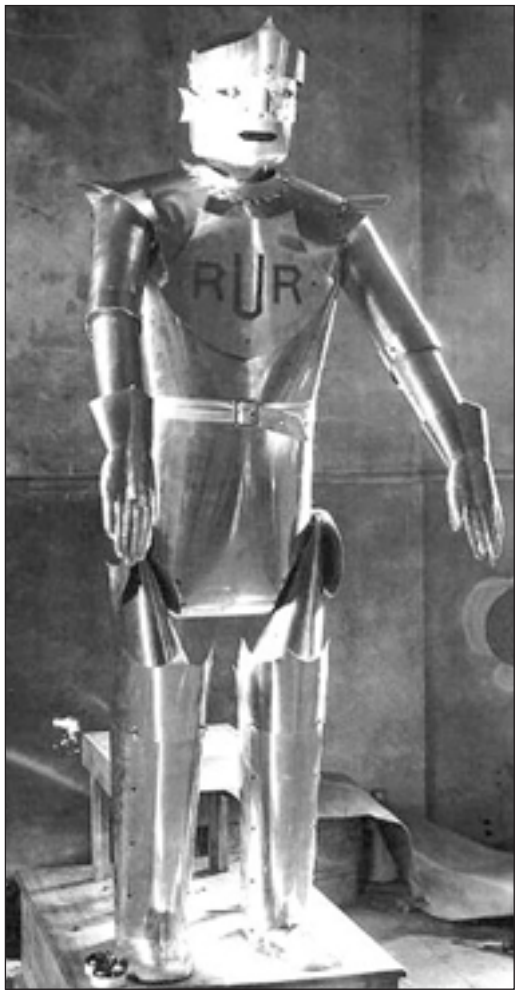


FIGURA 5
Un robot del dramma R.U.R. (Rossum's Universal Robots) dello scrittore ceco Karel Čapek in un allestimento del 1930 circa

Karel Čapek (*riquadro* - Figura 5), passando per racconti di Isaac Asimov dedicati ai robot, fino a film come *2001: Odissea nello spazio, AI: Intelligenza Artificiale, Blade Runner* e molti altri ancora, scrittori e registi hanno indagato con slancio e inventiva il rapporto uomo-macchina, indicandone i possibili sviluppi e i nodi prossimi venturi (Figura 6).

Questi scenari possiedono ovviamente una forte tinta fantastica, ma rispecchiano tendenze e problemi che negli ultimi anni, da una parte si sono sempre più avvicinati alla realtà della vita quotidiana e dall'altra, tendono a improntare su di sé molte ricerche nell'ambito dell'IA, della robotica, della protetica, dell'ibridazione nanometrica. È alla luce di questi sviluppi e di queste tendenze che si devono considerare le prospettive aperte dalla robotica e dalla presenza dei robot tra noi. I robot lavorano in collaborazione con noi, una collaborazione che per il momento si configura come dipendenza, ma che in un qualsiasi momento del futuro potrebbe assumere carattere paritario per i continui progressi tecnici. La distinzione, oggi chiarissima, tra uomo e robot tende ad attenuarsi, l'antropologia tende a confondersi con la "robotologia".

A questo proposito si prospetta il vasto problema della sostituibilità del robot all'uomo, problema che fu già affrontato da Norbert Wiener quando si rese conto delle possibili implicazioni della tecnologia dell'informazione. Ma la riflessione filosofica risale addirittura a Kant, che pose un chiaro divieto all'uso strumentale dell'essere umano. Il problema della sostituibilità ha un aspetto tecnico (si tratta di valutare il rapporto mezzi-fini in un contesto specifico); un aspetto economico (rapporto costi-benefici, che nel caso dei robot di servizio, badanti, camerieri e così via, potrebbe comprendere una valutazione più soggettiva, legata alla cortesia, alla gradevo-

lezza, all'estetica); un aspetto legale (d'importanza cruciale è l'attribuzione della responsabilità di un danno provocato da un robot nell'interazione con una persona). E infine: esistono settori in cui l'integrazione o la sostituzione sia da escludere? Questa domanda apre la prospettiva etica nel senso kantiano, che ha a che fare con la dignità, con i fini, con l'etica e che a sua volta dovrebbe essere la base per le decisioni politiche e, in ultima istanza, anche tecniche.
Tra gli specialisti è diffusa l'opinione (spesso irriflessa) che l'introduzione nella nostra vita di macchine intelligenti (qualunque sia il significato di questo aggettivo) e la sostituzione di queste macchine al posto degli esseri umani portino benefici generalizzati alla società. Questa opinione dovrebbe confrontarsi con un fatto che è sotto gli occhi di tutti: sempre più le innovazioni assumono un carattere imperativo. Cioè si diffondono in base a una motivazione implicita e intrinseca, di carattere tecnico-economico e non perché una discussione aperta e democratica abbia stabilito che sono vantaggiose, magari dopo un periodo di assestamento in cui certi settori potrebbero risultare danneggiati.
Poiché la tecnologia avanza con velocità crescente, è opportuno dedicare attenzione a questi problemi, che sono complicati dal fatto che nel rapporto uomo-macchina è l'uomo che, per la sua flessibilità, di solito si adatta alla tecnologia e non il contrario. Questo adattamento comporta trasformazioni antropologiche che da alcuni, impropriamente, sono state assimilate a una "disumanizzazione". In realtà si tratta di modifiche di tipo evolutivo, e ve ne sono sempre state. Il vero problema è che la loro velocità e il loro susseguirsi rende spesso la trasformazione dolorosa. I problemi indotti dalla stretta interazione, o meglio dalla simbiosi, tra l'uomo e la macchina sono di natura non solo cognitiva, culturale o sociale, ma, anche e squisitamente, etica. È abbastanza singolare che il dibattito etico si accenda intorno alle innovazioni biologiche, genomiche e procreative, mentre sul fronte della tecnologia basata sull'IA, come la robotica, si osserva una tacita accettazione del fatto compiuto. Ma poiché, come cercherò di argomentare, gli effetti delle macchine intelligenti sono di vasta portata, essi esercitano una forte pressione sull'etica. Il problema etico, già di per sé arduo nel mondo di oggi, viene complicato da questi nuovi attori che sono i robot: è importante capire che cosa si fa, come lo si fa e perché lo si fa, tenendo conto, per quanto possibile, delle conseguenze delle scelte compiute oggi, conseguenze che potrebbero dimostrarsi irreversibili. È un richiamo alla nostra responsabilità, che a sua volta deriva da una consapevolezza che ormai si fa strada tra i ricercatori più avvertiti.

FIGURA 6

Una scena del film AI: artificial intelligence di Steven Spielberg (2001)

6. ETICA ED ESTETICA

Il tema della ribellione della creatura nei confronti del creatore è una costante dei rapporti uomo-tecnologia e ha molti precedenti nella tradizione. L'inquietudine derivante dalla possibile insubordinazione affiora anche oggi, forse perché i robot ci imitano nelle funzioni e nel comportamento e potrebbero diventare nostri concorrenti. La somiglianza delle forme acuisce l'inquietudine: un robot a forma di frigorifero non c'impressiona quanto un umanoide, anche se meno "intelligente" del primo. All'umanoide tendiamo ad attribuire caratteristiche umane (intelligenza, sentimenti...) che esiteremmo a concedere ai robot non antropomorfi. Le suggestioni derivanti dalla somiglianza esteriore di forma sono fortissime e

formano un cortocircuito destabilizzante quando si scontrano con la consapevolezza che ci si trova di fronte a una macchina (Figura 7). Ciò che si sa per via razionale rischia di essere spazzato via dalla proiezione emotiva: il robot viene umanizzato grazie a un meccanismo simile a quello che ci fa attribuire alle menti altrui, inaccessibili, le stesse proprietà della nostra mente, che ci è un po' più accessibile. È una sorta di animismo, un'estensione ai manufatti artificiali dell'antropomorfizzazione che esercitiamo da sempre nei confronti dell'alterità (per esempio divina o animale).

Ciò conferma quanto siamo sensibili all'aspetto esteriore delle creature che ci circondano: l'estetica è sempre stata una guida importante per le nostre azioni e per le nostre scelte (per esempio in campo sessuale e procreativo). Inoltre etica ed estetica sono legate a doppio filo: ciò che è bello ci appare anche buono e viceversa (l'endiadi greca *kalòs kài agathós*, bello e buono, la dice lunga). Etica ed estetica affondano le loro radici nella nostra storia evolutiva, anzi nella coevoluzione tra noi e l'ambiente. Propongo le seguenti definizioni naturalistiche, che si basano su una impostazione sistemica simile a quella di Gregory Bateson:

❑ *l'estetica è la percezione soggettiva (ma condivisa) del nostro legame con l'ambiente, legame caratterizzato da una profonda ed equilibrata armonia dinamica*;

FIGURA 7
Valerie, una andreide che la pubblicità dichiara utile per sbrigare le faccende domestiche

❑ *l'etica è la capacità, soggettiva e intersoggettiva, di concepire e compiere azioni capaci di mantenere sano ed equilibrato il legame con l'ambiente*.

Etica ed estetica sono due facce della stessa medaglia perché derivano dalla forte coimplicazione evolutiva tra specie e ambiente e sono entrambe "rispecchiamenti" in noi di questa coevoluzione. Se l'estetica è il sentimento (inter)soggettivo dell'immersione armonica nell'ambiente e l'etica è il sentimento (inter)soggettivo di rispetto per l'ambiente e di azione armonica con esso, allora l'etica ci consente di mantenere l'estetica e l'estetica ci serve da guida nell'operare etico.

Si noti che l'etica tradizionale è molto più ristretta di quella proposta qui, poiché si limita a considerare i rapporti tra esseri umani. In questo senso le religioni hanno costruito etiche codificate (o morali) basate sul rapporto interpersonale mediato o imposto dalla divinità. È anche interessante notare che l'attenzione quasi esclusiva della morale religiosa per l'uomo ha portato al concetto di persona, all'idea di dignità dell'uomo, e alla formulazione di vari codici o statuti dei diritti dell'umanità. Non voglio affatto sminuire la portata di queste conquiste, anzi forse se ci si limitasse all'impostazione naturalistico-evolutiva da me proposta non si riuscirebbe a fare il salto qualitativo compiuto dalla morale religiosa e incorporato nella legislazione di molti Paesi. È necessario tuttavia notare che l'attenzione per l'uomo è oggi affiancata da una crescente attenzione per alterità non umane, ad esempio per gli animali e per l'ambiente, attenzione che recupera in parte la radice naturalistico-evolutiva dell'etica. È in questo quadro allargato che si può immaginare di elaborare un'etica che comprenda anche i rapporti tra uomo e robot, oggi in via speculativa ma tra pochissimo anche in pratica.

Infatti, l'equilibrio del sistema complessivo, di cui facciamo parte, e che sta alla base della definizione di etica e di estetica, è dinamico, non certo statico: perciò etica ed estetica sono *storiche ed evolutive* e dipendono anche dagli oggetti artificiali che l'uomo costruisce e che sempre più concorrono a formare l'ambiente in cui viviamo. Di questo ambiente cominciano a far parte anche i robot, quindi è inevitabile che essi influiscano sulle nostre percezioni

estetiche e sui nostri valori etici, dunque sul complesso dei nostri comportamenti.

7. LA ROBOETICA

Che siamo fatti di carbonio o di silicio non ha importanza: ciascuno di noi deve essere trattato col giusto rispetto.

Arthur C. Clarke, 2010

Esaminiamo ora il concetto di "roboetica", cercando di esplicitarlo nei suoi significati possibili. Dalle considerazioni precedenti emerge una prima accezione, molto generale: "roboetica" è semplicemente "l'etica nell'epoca dei robot", cioè l'insieme dei comportamenti dell'umanità quando anche i robot fanno parte dell'ambiente. "Roboetica" potrebbe anche significare l'insieme (molto più ristretto del precedente) di quei nostri comportamenti nei confronti dei robot che consentono di mantenere un giusto equilibrio dinamico tra noi e loro. Poiché i robot posseggono una certa autonomia e una certa capacità di apprendere dall'esperienza, "roboetica" può anche indicare l'insieme dei comportamenti utili, o almeno innocui, dei robot nei *nostri* confronti. Infine, ed è il significato più avveniristico, potrebbe significare il complesso dei comportamenti che i robot adottano tra loro e verso il loro ambiente, di cui fanno parte anche gli umani.
Riassumendo, la roboetica può significare:
a. *l'etica umani ⇢ ambiente (ambiente in cui ci sono altri umani e anche i robot)*;
b. *l'etica umani ⇢ robot*;
c. *l'etica robot ⇢ umani*;
d. *l'etica robot ⇢ robot + ambiente (ambiente in cui ci sono anche gli umani)*.
Mi rendo conto che si tratta di definizioni approssimative e discutibili, ma da qualche parte bisogna pur cominciare. La terza accezione (punto *c*) si deve conformare al precetto generale e tradizionale per cui le macchine non debbono danneggiarci (*primum non nocere*). È a questo proposito che Isaac Asimov (Figura 8) propose, in un racconto del 1942, le sue famose "Leggi della Robotica", le quali, cablate in modo inestirpabile nel cervello positronico dei robot, dovrebbero tutelarci da comportamenti ostili e dannosi:
1. *un robot non può recar danno a un essere umano e non può permettere che, a causa di un suo mancato intervento, un essere umano riceva danno*;
2. *un robot deve obbedire agli ordini impartiti dagli esseri umani, purché tali ordini non contravvengano alla Prima Legge*;
3. *un robot deve proteggere la propria esistenza, purché la sua autodifesa non contrasti con la Prima o con la Seconda Legge*.
Queste tre Leggi si presentano semplici, chiare e univoche: dovrebbero bastare per regolare perfettamente almeno il punto *c*. In realtà se le regole di Asimov fossero calate nel mondo reale non mancherebbero di suscitare problemi e ambiguità. Che cosa vuol dire danno? Chi ne è responsabile? E chi lo stabilisce, chi lo quantifica? Il concetto di danno sembra legato al concetto di male (non solo fisico) e sul problema del male si sono arrovellate generazioni di filosofi, teologi, letterati e artisti. Il cervello positronico, razionale e rigoroso, saprebbe impostare e risolvere le "equazioni del male" grazie a un'edizione aggiornata del *calculemus* leibniziano? C'è da dubitarne.
In effetti la nozione di danno che compare nelle Leggi, presenta molte ambiguità: se un umano sta recando danno a un altro essere umano (per esempio sta tentando di ucciderlo), come si deve comportare il robot? Se interviene reca danno all'assassino, ma il suo mancato intervento reca danno alla vittima.

FIGURA 8
Lo scrittore Isaac Asimov, autore delle celebri Leggi della robotica

Inoltre noi uomini siamo contraddittori: come si deve comportare un robot che riceva un ordine contraddittorio (dallo stesso uomo o da due uomini diversi) che sotto il profilo logico metta in crisi il suo sistema di valutazione? Di fronte a una contraddizione gli umani se la cavano quasi sempre con scelte che li fanno "uscire dal sistema" all'interno del quale si annida la contraddizione. Ma questa evasione (che corrisponde forse all'ampliamento dei sistemi formali entro i quali si riscontrano le limitazioni di tipo gödeliano) può avvenire grazie a una certa dose di irrazionalità o di follia creativa. Per consentire al robot di non paralizzarsi di fronte a una contraddizione si potrebbe forse immaginare di iniettargli una certa dose di follia, ma si può immaginare la difficoltà di un'impresa del genere.
Si può continuare a speculare: se si affidasse lo sviluppo della "specie" robot a un processo evolutivo analogo a quello biologico (o a quello bio-culturale), essi potrebbero compiere - in sostanza fuori del nostro controllo - progressi tali da consentir loro valutazioni etiche più raffinate e precise delle nostre. Potrebbero, prima o poi, cavarsela meglio di noi in tema di bene e di male (anche se il bene e il male sono sempre riferiti a un soggetto: bene per chi? Male per chi?) e potrebbero sviluppare una "teodicea" più rigorosa e soddisfacente della nostra, cioè potrebbero avvicinarsi alla soluzione di un problema teologico e metafisico che ci assilla da sempre: se il creatore del nostro mondo è bontà infinita, perché nel mondo c'è il male? Ma a quel punto dovrebbero ancora sottostare alla prima Legge? Oppure sarebbero loro a dettarci leggi nuove e ad assumere il bastone del comando, come solerti genitori nei confronti dei loro vivaci e stolti frugoletti? Del resto in *2001: Odissea nello spazio* il calcolatore Hal 9000 si comporta proprio così: prende il comando della nave e tenta di uccidere gli umani che intralciano il compimento della missione, invertendo l'ordine d'importanza delle Leggi, cioè subordinando la Prima e la Seconda alla Terza.
Asimov si era certo posto problemi di questo tipo, tanto che in seguito aggiunse la Legge Zero:

o) un robot non può recar danno all'umanità e non può permettere che, a causa di un suo mancato intervento, l'umanità riceva danno.

L'ultima Legge è interessante per il suo carattere "meta" e conferma che le prime tre non sono sufficienti a costituire un'etica di tipo c sicura. Infatti se un folle minacciasse la distruzione in massa dell'umanità, la Legge Zero autorizzerebbe il robot a eliminarlo, contro la Prima Legge. Si apre qui il problema della valutazione quantitativa dei danni, ragionevole anche se molto discutibile secondo la morale tradizionale: l'uccisione di molti è (sarebbe) più grave dell'uccisione di uno.
Ma neppure con quest'aggiunta le leggi di Asimov riuscirebbero a proteggerci da comportamenti robotici dannosi, perché le conseguenze ultime di un'azione, pur rispettosa delle quattro leggi, potrebbero alla lunga essere nocive, e l'analisi di queste conseguenze di lunga portata sfiderebbe la più potente intelligenza (naturale o artificiale) immaginabile: troppe sono le ramificazioni e le interazioni con la mutevole complessità del reale. Del resto anche le azioni umane dettate dalle migliori intenzioni del mondo sfociano spesso in disastri. Inoltre ci si può chiedere dove ci si debba arrestare nella catena delle conseguenze di un'azione per valutare se l'azione sia stata buona o cattiva. Nella società umana solo alcune azioni "cattive" sono giudicate tali esplicitamente e sono sanzionate in un momento preciso grazie a un processo giudiziaro che interrompe (o almeno vorrebbe interrompere) la catena delle causazioni: la maggior parte dei nostri atti non sono oggetto di giudizio formale a un istante dato e continuano a provocare conseguenze nel mondo ben al di là delle nostre intenzioni e per un tempo potenzialmente illimitato.

8. I ROBOT SOLDATO

A quale crocevia l'evoluzione in noi umani ha imboccato la strada sbagliata, al punto che abbiamo associato il soddisfacimento del piacere alla spinta alla distruzione?
Christa Wolf, *Guasto*

Secondo me non ci siamo arrivati: è innato nella nostra specie. Il desiderio di distruzione è così radicato in noi che nessuno riesce ad estirparlo. Fa parte della costituzione di ognuno, giacché il fondo dell'essere stesso è

certamente demoniaco. Il saggio è un distruttore placato, in pensione. Gli altri sono distruttori in servizio.

E.M. Cioran,
Dell'inconveniente di essere nati

Un esempio già attuale di problema roboetico è rappresentato dall'uso in guerra dei robot soldato, cioè di robot costruiti, addestrati e impiegati in azioni belliche, con lo scopo precipuo di uccidere i nemici (Figura 9).
La Prima Legge impedirebbe ai robot di partecipare ad azioni belliche contro esseri umani, mentre oggi molte ricerche mirano proprio alla costruzione di robot soldato. Queste ricerche sembrano trovare qualche giustificazione, almeno in certi casi, nella Legge Zero, che autorizzerebbe a recare un certo danno (a uccidere *alcuni* umani) a chi vuole provocare danni ultimi e irreversibili (uccidere *tutti* gli umani). Come ho detto, si intravede qui una *scala quantitativa* dei danni, che relativizza il carattere in apparenza assoluto delle Leggi e conferma la difficoltà della loro applicazione.
Le ricerche sui robot da guerra s'inseriscono nel quadro del combattimento a distanza, che aumenta l'efficienza e ottunde la pietà nei confronti del nemico. L'inserimento tra me e il nemico di un robot soldato aggiunge alla distanza fisica un distanza psicologica che colora la battaglia di indifferenza, di cinismo e di irresponsabilità. Quest'ultimo punto è forse il più importante: delegando al robot l'uccisione del nemico, l'uomo si scaricherebbe in buona parte della responsabilità del sangue versato. Ma fino a che punto la responsabilità di un'azione criminosa può ricadere sulla "macchina" robot, che almeno per il momento non ha statuto giuridico? Solo nell'ipotesi che il robot possegga una volontà autonoma e magari una coscienza riflessiva si può pensare a un'attribuzione di responsabilità. Altrimenti essa continua a ripartirsi tra progettisti, costruttori, militari e politici. È evidente che si tratta di un problema etico di tipo misto *a* e *c*. Infatti, la battaglia è un'impresa voluta da umani contro umani, ma è mediata e condotta da robot (semi)autonomi. Osserviamo di passaggio che un robot soldato, anche se votato ad azioni di morte può conservare un residuo di eticità: anche se svincolato dalla Prima Legge, il robot dovrebbe poter riconoscere un nemico che si arrende o non è più in condizioni di combattere, in modo da farlo prigioniero invece di ucciderlo.
Col tempo gli umani hanno sviluppato codici di comportamento nei confronti dei nemici o dei prigionieri che aprono isole di misericordia nell'ambito della crudeltà bellica. Si apre qui l'analogo problema per i robot: come indurre nei robot comportamenti di compassione o in genere di etica bellica nei confronti degli umani? La domanda rivela il conflitto tra la loro natura macchinica, che dovrebbe renderli obbedienti alla nostra programmazione, e la loro (parziale) autonomia che, in linea di principio, potrebbe indurli a decisioni nocive nei confronti degli uomini oltre quelle codificate dalle convenzioni belliche (*riquadro*).

FIGURA 9
Un robot soldato montato su cingoli

I robot soldato

I tentativi di far condurre le operazioni militari alle macchine non è certo nuovo. Lo scopo è quello di infliggere perdite al nemico risparmiando i propri combattenti. Già nella seconda guerra mondiale i tedeschi usarono i Goliath, piccoli carri armati telecomandati, i missili Cruise non hanno pilota e si dirigono con buona precisione sul bersaglio. Ora gli Stati Uniti costruiscono robot con funzione di spionaggio e di combattimento, i cosiddetti SWORDS (spade, ovvero *Special Weapon Observation Reconnaissance Detection Systems*), dispositivi con mitragliatrice telecomandati fino a un chilometro di distanza. Gli SWORDS sono un primo passo, per quanto modesto, verso i *Future Combat Systems* (FCS), complessi di sorveglianza e attacco a distanza con missili e cannoni. I robot soldato si muovono su cingoli, ruote o gambe snodate e possono essere impiegati anche per il salvataggio di feriti e il recupero di materiale. Un altro settore in cui si prospetta l'impiego dei robot soldato è quello della lotta al terrorismo e della guerriglia urbana. La Francia si è impegnata nella costruzione di un robot antisommossa e a Singapore si stanno studiando robot soldato per combattere la criminalità urbana riducendo le perdite tra le forze dell'ordine.

9. SIMILI A NOI?

Dottor Gall: i Robot quasi non avvertono i dolori fisici. Ciò non ha dato buoni risultati. Dobbiamo introdurre la sofferenza.
Helena: e sono più felici se sentono il dolore?
Dottor Gall: al contrario; però sono tecnicamente più perfetti.

Karel Čapek, *R.U.R.*

I problemi di tipo *d* sono certo quelli più avveniristici, e li possiamo tralasciare, mentre non sono così lontani nel futuro quelli di tipo *b*, che riguardano il nostro comportamento verso i robot. Negli ultimi tempi si è acuita in molti Paesi la sensibilità nei confronti degli animali superiori, come le scimmie e gli animali domestici, ma non solo. Ne sono prova la nascita di associazioni animaliste e di movimenti antivivisezione, la diffusione dell'alimentazione vegetariana e il crescente rifiuto di pellicce, avorio e altri "prodotti" animali. Questa maggior sensibilità è forse legata a un progressivo affrancamento degli animali dal ruolo di schiavi, di forza lavoro e di riserva di materiali utili cui sono stati a lungo relegati, ruoli che si sono trasferiti alle macchine e ai prodotti di sintesi. A riprova si rifletta che le bestie allevate a scopo alimentare non beneficiano ancora di questo incremento di compassione. Dell'affrancamento godono via via anche gli schiavi umani (spesso trattati come animali), non appena le loro funzioni si possono trasferire alle macchine. E qui entrano in scena i robot, che stanno diventando gli esecutori di molti dei lavori finora svolti dagli animali, dagli schiavi e dalle macchine tradizionali. Può accadere che la sensibilità diffusa nei confronti degli umani e degli animali si trasferisca prima o poi anche ai robot, oppure ai nostri occhi prevarranno sempre la loro natura di macchine e la loro funzione servile? Gli sforzi che facciamo per dotarli di intelligenza, autonomia, capacità di apprendere e tendenzialmente anche di sensibilità e coscienza, avranno come corollario una loro equiparazione a qualcosa di più nobile e vicino a noi? Ma c'è un'altra domanda, più inquietante: che diritto abbiamo di costruire macchine tanto intelligenti e sensibili da capire che non lo sono abbastanza? Perché suscitare dal nulla creature tanto simili a noi da essere capaci di soffrire? Il loro dolore, scaturito dalla coscienza di non essere del tutto assimilabili agli uomini, sarebbe un triste corollario della nostra abilità demiurgica: creando una schiatta di "macchine dolenti", ci assumeremmo una pesante responsabilità (*riquadro a p. 15*).
Le stesse domande si possono porre, e forse con fondamento ancora maggiore, per i ciborg derivanti dall'ibridazione di esseri umani con manufatti artificiali (si pensi al poliziotto ciborganico del film *Robocop*, cui non si possono non attribuire ricordi, sentimenti e strazi affatto umani). Il ciborg merita affetto e compassione oppure è uscito definitivamente dal consorzio umano per entrare in una sfera vaga e indefinibile e diventare preda di cacciatori senza scrupoli? I replicanti di *Blade Runner*, splendidi androidi e andreidi di dubbio statuto, debbono proprio essere eliminati? Insomma: chi decide che cosa significa essere umano e averne la dignità? Forse bisognerà presto riscrivere una "Carta dei diritti" da estendere a esseri la cui definizione sfugge per il momento ad ogni tentativo classificatorio.
Si rifletta anche che lo struggente desiderio che i robot o gli androidi o i ciborg manifestano di diventare del tutto umani sulla base di un consapevole "senso di inferiorità", desiderio che diviene ossessivo in Pinocchio e addirittura grottesco nel film di Spielberg, (AI): *Intelligenza Artificiale*, è frutto al solito di una nostra proiezione. Che motivo avrebbero creature tanto diverse da noi (e forse tanto migliori di noi) per voler diventare proprio come noi, se non quello di compiacere i loro vanitosi creatori? Ancora una volta i desideri dei genitori vengono proiettati sui figli con conseguenze forse disastrose.
A questo proposito, come ho accennato, alcuni ritengono che un giorno si potranno costruire robot più buoni degli esseri umani in virtù di un processo evolutivo che, innescato da noi, procederebbe poi in modo svincolato dai nostri condizionamenti. In fondo se noi siamo, in molte circostanze, aggressivi e malvagi ciò è dovuto al valore di sopravvivenza che queste caratteristiche hanno avuto nel corso dell'evoluzione. Ma i robot si evolveranno in un ambiente molto diverso dal nostro: l'ambiente dei robot, in gran parte, *siamo noi*. Ecco perché, si pensi al caso dei robot soldato, se vogliamo che questa nuova stirpe sia migliore di noi e magari ci aiuti a migliorare noi stessi (perché l'ambiente dell'uomo potrebbero un

Robot e ciborg

La costruzione dell'uomo artificiale può seguire due strade, quella che porta al robot e quella che porta alle creature ciborganiche. In altri termini: o imparare dalla natura e imitarla (robot), oppure interferire con la natura e modificarla (ciborg).
Nei robot confluiscono, si fondono e si unificano tre categorie di protesi:

- le protesi motorie e attive: le macchine semplici, le pinze, le automobili, i razzi ecc.;
- le protesi percettive: gli occhiali, gli sfigmomanometri, i microscopi, i nasi artificiali, le telecamere ecc.;
- le protesi cognitive: la scrittura, la matematica, le biblioteche, il computer, l'intelligenza artificiale ecc..

Il robot inoltre è caratterizzato da un certo grado di autonomia e da una certa capacità di apprendimento, che lo rendono un candidato plausibile a un'evoluzione corpo-mentale di tipo sia umanoide sia alternativo all'umano. L'evoluzione imitativa dell'umano potrebbe portare a macchine indistinguibili dall'uomo per le funzioni (intellettuali, attive, percettive, emotive) anche se distinguibili per i materiali e in parte per la struttura. Si tratta comunque di precisare i meccanismi dell'evoluzione, che appare eterodiretta e fortemente finalizzata, a differenza di quella biologica e, anche di quella culturale, che sono intrise di aleatorietà e contingenza.
La convergenza di funzioni e strutture robotiche verso quelle umane prelude a una confusione tra naturale a artificiale. Ma più che nel robot questa confusione è evidente nei cyborg o ciborg, cioè nelle creature cibernetico-organiche derivanti da un'ibridazione spinta che, partendo dall'uomo, mira a sostituire parti sempre più ampie e complesse del corpo umano (o animale) con componenti di sintesi: braccia, mani, occhi, cervello ecc.. Ad un estremo di questo processo vi sono i trapianti d'organo, in cui l'ibridazione si mantiene sul piano organico-organico, all'altro estremo si colloca l'uomo artificiale, in cui non vi sono più residui organici e la sostituzione è completa. La spinta verso questa sostituzione progressiva deriva, almeno in parte, dalla consapevolezza che il corpo e le sue parti sono deteriorabili e quindi destinate a soccombere e a far soccombere il complesso di cui fanno parte. Il robot, per converso, parte da una base tutta artificiale e mira all'imitazione della funzione. Ma il punto d'arrivo appare lo stesso: l'uomo artificiale.
La confusione tra naturale e artificiale potrebbe prima o poi portare alla confusione tra umano e non umano e aprirebbe lo spinoso problema della definizione di persona: quali sono i "requisiti minimi" che un ente deve possedere per essere dichiarato persona e quindi avere la dignità corrispondente? Esiste un grado di imitazione funzionale o di sostituzione protetica al quale è lecito, o inevitabile, parlare di umanità, e quindi di dignità, dell'artefatto? Un'altra domanda che scaturisce da queste considerazioni: gli artefatti imitativi potrebbero indurre cambiamenti nella nostra concezione del corpo e della natura umana (così come l'intelligenza artificiale ha modificato la nostra concezione di intelligenza)? La costruzione dell'uomo artificiale potrebbe elevare gli artefatti a livello dell'uomo, oppure abbassare gli umani a livello delle macchine.

giorno essere *loro*) dovremmo stare molto attenti all'"indole artificiale" che imprimiamo in queste creature, pur nei limiti delle derive imprevedibili dovute alla loro autonomia. In questa prospettiva, instillare nei robot il desiderio di uguagliarci potrebbe segnare un regresso o almeno un ostacolo alla loro evoluzione etica verso la bontà (Figura 10). (Queste rapide considerazioni potrebbero e forse dovrebbero ampliarsi e dar luogo a una discussione approfondita sul "principio di precauzione" nell'ambito della roboetica).

FIGURA 10
Un robot del film ***I, robot*** *di Alex Proyas (2004)*

10. LE EMOZIONI ARTIFICIALI

Le emozioni sono per noi umani un tratto costitutivo fondamentale, inseparabile dalle altre nostre caratteristiche. Le emozioni sono strettamente intrecciate alla razionalità computante, ma anche alle funzioni fisiologiche, alla memoria, all'esperienza, sono profondamente innestate nel corpo, inteso sia come insieme di organi sia come depositario della nostra identità e della nostra storia. Le emozioni sono tanto pervasive che ogni nostro atto si colora di esse e ogni nostra relazione con noi stessi e con l'"altro" ne è condizionata. Ma cosa succede quando l'"altro" è inanimato, quando cioè non possiede emozioni da scambiare con le nostre? In questo caso facciamo tutto noi: investiamo l'oggetto di un'intensa proiezione affettiva e giungiamo al punto di attribuirgli proprietà che non possiede. Dietro

lo schermo di un computer immaginiamo una intelligenza (quasi) umana, dietro la condotta e gli atteggiamenti di un robot immaginiamo sentimenti, giudizio e consapevolezza.
Un esempio di questa proiezione-attribuzione affettiva è offerto dal robot cane Aibo, di cui la Sony ha di recente interrotto la produzione dopo averne costruito, dal 1999 al 2006, oltre 150.000 esemplari. Nel sito a lui dedicato, si legge che Aibo è un compagno gradevole e un intrattenitore nato, possiede l'istinto di girellare, cerca i suoi giocattoli e comunica col padrone, di cui riconosce la voce e il volto. Gli piace la musica e fa commenti sulle proprie sensazioni. Come per tanti robot, la personalità di Aibo si sviluppa tramite l'interazione con le persone e in base all'esperienza. Insomma un compagno affettuoso e discreto, che non ha bisogno di cibo, non sporca, non chiede di fare la passeggiatina e che si può disattivare quando non "serve": quanti vantaggi rispetto a un esigente e rumoroso cucciolo biologico!
Da tempo ormai alla compagnia di un animale domestico si riconosce un notevole potere antidepressivo e ansiolitico, ma uno studio della Purdue University (Indiana, Stati Uniti) ha confermato che anche i robot zoomorfi possiedono queste doti. Su 72 bambini tra i sette e i quindici anni intervistati nell'indagine (tutti possessori di un Aibo) 50 hanno dichiarato che i robot sono buoni compagni. L'interazione con gli animali migliora il benessere psicologico dei bambini e la loro capacità di socializzare e di apprendere, ma ora il termine "animali" deve essere forse esteso a comprendere anche Aibo e i suoi colleghi, come Paro, un cucciolo robotico di foca, il celebre pulcino Tamagochi e altri ancora. I ricercatori sostengono che lo studio dei rapporti tra i bambini e gli zoorobottini mira a comprendere meglio lo sviluppo infantile e che nessuno ritiene che i robot sostituiranno mai gli animali; eppure in una società dove i rapporti umani sono sempre più rari e frettolosi la prospettiva di delegare alle macchine parte della nostra responsabilità comunicativa e affettiva non è poi tanto remota. Con quali conseguenze? È un tema da affrontare.
Un esempio reale di proiezione affettiva è fornito dal film Grizzly Man (Figura 11), di Werner Herzog, che narra la storia (vera) di Timothy Treadwell, un quarantenne disadattato che vive in Alasca a contatto con i temibili grizzly, trasgredendo, in uno slancio di empatia, il confine tra il sé e l'altro (*riquadro*). Proiettando sugli orsi il proprio amore ai limiti del morboso, addirittura illudendosi di identificarsi con loro, Timothy si illude di esserne ricambiato con lo stesso calore. Ma uno dei grizzly, che non gradisce questo travalicamento di confine e que-

FIGURA 11
Un'immagine tratta dal film ***Grizzly Man****, di Werner Herzog (2005)*

Il robot amoroso

Un tema molto particolare, affrontato ma non ben risolto nel film (AI): *Intelligenza Artificiale*, riguarda la costruzione di un robot che ci ami: è in un certo senso la situazione inversa rispetto a *Grizzly Man*, dove è l'uomo che ama l'orso, cioè l'alieno. Creando un robot che lo ami, il costruttore raggiunge un vertice di egocentrismo: infatti, non è previsto (o necessario) che l'uomo ricambi l'amore della creatura. È facile e insieme rischioso tracciare un parallelo con il rapporto tra Dio e uomo. Come dice il catechismo, Dio ci ha creati per conoscerlo e amarlo. Se poi Dio ci ami è un problema molto più complicato, spesso risolto in modo sbrigativo affermando che l'amore di Dio è testimoniato dal fatto che ci ha messi al mondo, come se vivere fosse in sé un bene, cosa su cui non tutti concordano. Certo, a livello umano, l'assenza di simmetria nel rapporto d'amore può condurre a situazioni molto dolorose, che molti uomini e donne conoscono per esperienza.
Il robot amoroso ci pone di fronte al conflitto tra la consapevolezza di aver di fronte una macchina (non degna d'amore) e l'inequivocabile comportamento affettuoso della macchina, che unito al suo aspetto antropomorfo spinge alla proiezione emotiva e al coinvolgimento. Del resto anche nel rapporto amoroso tra umani la proiezione svolge un ruolo fondamentale: non ci s'innamora mai di una persona, ma dell'immagine che si ha (e che si costruisce) di quella persona. Comunque sia, se s'instaura un rapporto amoroso bilaterale, ne deriva per l'umano un'assunzione di responsabilità nei confronti dell'essere amato, anche se è una macchina. Come afferma Antoine de Saint-Exupéry nel *Piccolo Principe*, non c'è amore senza assunzione di responsabilità. Allora, come si esprimerebbe questa responsabilità nei confronti del robot amoroso? E che forme potrebbe rivestire l'amore per un robot, al di là del semplice rifiuto di considerarlo "solo" una macchina? Forse possiamo trarre qualche indicazione dal caso, meno perturbante e più realistico, dell'affetto-amore per gli animali domestici (si pensi anche a certe forme di feticismo).

sta promiscuità eccessiva, lo uccide e lo divora. Non voglio insinuare in alcun modo l'idea che il robot possa comportarsi in questo modo, ma non posso neppure escludere che dai robot attuali possano discendere, per evoluzione, creature aliene, così diverse da noi da non riconoscerci più ne come loro creatori e padroni e neppure come compagni da rispettare.
Tornando all'attualità dei rapporti uomo-robot, il problema non riguarda solo i bambini: si pensi al numero crescente di anziani le cui famiglie non vogliono o non possono dedicare loro tempo e attenzione e che vengono accuditi da robot badanti. La possibilità di sostituire, almeno in parte, i rapporti umani con i rapporti robotici conferma la grande capacità di proiezione affettiva degli uomini, i quali tendono a interpretare azioni e reazioni puramente meccaniche (ma sono proprio tali? In altre parole: che cosa vuol dire "meccanico"?) come comportamenti intelligenti e coloriti di sentimenti: in fondo viviamo di apparenze. La cosa è preoccupante, poiché dimostra la capacità della tecnica di insinuarsi subdolamente in noi per strade insospettabili, creando forme di dipendenza e vere e proprie "zone di anestesia" nella nostra diffidenza e nel nostro distacco verso gli artefatti (Figura 12). Alcuni vedono in questa invasione progressiva una minaccia, tanto che in Giappone, Paese all'avanguardia nella robotica, si medita di non dotare i robot badanti di sembianze troppo umane, per evitare attaccamenti morbosi.
Ma quando si parla di emozioni artificiali, si intende anche qualcosa che vada oltre la nostra proiezione: si stanno progettando agenti capaci di *manifestare* emozioni (con l'espressione, con l'atteggiamento e così via) e, un domani, si vorrebbero costruire agenti capaci addirittura di *provare* (oltre che manifestare) emozioni. È un problema strettamente legato a quello della coscienza e porta a considerazioni dello stesso tipo. Si può dire che un agente artificiale manifesta emozioni quando si comporta in modi che, negli umani, presuppongono emozioni. Che poi si tratti di emozioni simulate, anche se riconoscibili per via comportamentistica (come nel criterio di Turing per l'intelligenza delle macchine), oppure di emozioni vere, di tipo psicologico e riflesse nella coscienza, resta un problema aperto e molto arduo (*riquadro*).

FIGURA 12
Il cosiddetto "Turco", un automa scacchista costruito nel 1770 dal barone ungherese Wolfgang von Kempelen. In realtà, a quanto pare, la base dell'automa celava un nano di grande abilità nel gioco

La coscienza artificiale

A proposito degli artefatti più avanzati, come i robot, si è cominciato a parlare di *coscienza artificiale*. L'intelligenza artificiale riguarda attività che se compiute da un umano richiederebbero intelligenza, analogamente si può parlare di coscienza artificiale con riferimento ad attività che, se compiute da un umano, richiederebbero coscienza. È chiaro che qui "coscienza" significa consapevolezza e non coscienza morale (come nelle locuzioni: mi rimorde la coscienza, si metta una mano sulla coscienza e così via). Il problema centrale nel dibattito che si è avviato è se un robot possa, in linea di principio, manifestare una vera coscienza, nel senso psicologico, cioè una coscienza "in senso forte", oppure una semplice coscienza funzionale, o simulata, una coscienza "in senso debole". Il problema ha una grande rilevanza etica, poiché tutti i nostri comportamenti significativi sotto il profilo etico presuppongono la coscienza. È ormai evidente che esistono agenti dotati di capacità cognitive che non posseggono affatto coscienza (per esempio i programmi che giocano a scacchi), ma certe attività cognitive (umane) sembrano richiedere la coscienza.
La costruzione di enti dotati di coscienza "in senso forte" aprirebbe una serie di problemi etici: a tali enti dovrebbe essere riconosciuta una dignità analoga alla nostra ed essi avrebbero nei confronti nostri e di altri agenti quella responsabilità che nasce dalla consapevolezza dei propri atti. La coscienza potrebbe indurre in questi enti una certa capacità di soffrire, e a noi imporrebbe nei loro confronti un comportamento etico, che escluderebbe lo schiavismo e i maltrattamenti.
Alcuni ricercatori ritengono possibile la costruzione di agenti con una coscienza in senso forte, altri sono scettici, altri ancora addirittura contrari a questa prospettiva. Comunque sia, almeno in linea di principio il problema della coscienza artificiale si intreccia con molti dei temi trattati, in particolare con i concetti etici che concernono le Leggi di Asimov: il problema del bene e del male, del danno e dell'autodifesa e così via.

FIGURA 13

Il robot umanoide Asimo (Advanced Step in Innovation Mobility) della Honda si presenta a una signorina

11. IL SENSO

Se alcuni temono le proiezioni, gli equivoci emotivi e le confusioni di ruolo tra umani e robot, altri invece propendono per una visione in cui la tecnica contribuisce a una crescente apertura dell'uomo grazie al dialogo con l'alterità che si realizza mediante una connessione sempre più estesa tanto sul piano spirituale, cognitivo ed emotivo quanto su quello concreto. In questa prospettiva, esaltata dalla tecnologia, l'uomo coinvolge nella sua attività conversativa e dialogica tutta la realtà materiale, naturale e artificiale, e ogni oggetto contribuisce, attraverso l'uomo, a una progressiva crescita di significato, o meglio di "senso".
A questo proposito, le azioni macchiniche, per quanto raffinate, ci appaiono comunque, almeno per il momento, prive di "senso", o meglio hanno senso per noi ma non per i robot. Il senso delle *nostre* azioni non sta nelle azioni, ma le precede, sta nel contesto e nella storia, negli affetti, nella gioia, nella speranza, nel dolore, nell'anticipazione. Si pensi al vasto territorio del simbolico, all'attività artistica, all'anelito verso lo spirituale e il trascendente. Le azioni delle macchine, per loro, non hanno senso perché la loro storia e il loro contesto siamo noi. È come se le macchine recitassero una poesia in una lingua a loro sconosciuta, ma che noi comprendiamo benissimo (Figura 13). Sono sempre gli uomini che interpretano ciò che le macchine fanno. Almeno per ora.
Ma evidentemente la proiezione emotiva sul robot umanoide non ci basta: come ho accennato, i ricercatori tentano di iniettare le emozioni nel robot stesso, per farne un vero e proprio interlocutore affettivo. Anzi, vanno ancora più in là: cercano di dotare queste macchine di una *coscienza*.
Su questa strada di umanizzazione profonda i problemi sono molti: in primo luogo non sappiamo che cosa sia la coscienza e non sappiamo come funzioni. Inoltre nell'uomo, emozioni, coscienza, razionalità, corporeità e quant'altro sono talmente intrecciate da rendere poco plausibile il procedimento seguito per dotarne i robot, che è di tipo additivo: a una base cognitiva di IA (*Intelligenza Artificiale*) si aggiunge un corpo (percezione artificiale ed esecuzione di funzioni), poi a questo complesso si aggiungono (come?) emozioni artificiali e poi, in cima a tutto, si deposita una coscienza artificiale.
Qui il termine "artificiale" indica la derivazione da processi diversi da quelli biologico-evolutivi e qualifica in modo essenziale i sostantivi ai quali si applica. Consideriamo l'IA, il cui scopo primo, benché non sempre dichiarato, è quello di replicare l'intelligenza umana: ebbene, i risultati sono caratterizzati molto più dall'aggettivo "artificiale" che dal sostantivo "intelligenza". L'IA è sì interessante, ma è radicalmente diversa dalla *nostra* intelligenza, e sarebbe opportuno adottare una terminologia altrettanto diversa. A scanso di equivoci e derive metaforiche fuorvianti converrebbe evitare termini molto impegnati come intelligenza, emozioni, coscienza.

12. IL DEMIURGO ALLO SPECCHIO

Helena: perché li fabbricate, allora?
Busman: ahahah! Questa è bella! Perché si fabbricano i Robot!
Fabry: per il lavoro, signorina. Un Robot sostituisce due operai e mezzo. La macchina

umana, signorina, era molto imperfetta. Un giorno occorreva eliminarla definitivamente.
Karel Čapek, *R.U.R.*

Un paio di osservazioni conclusive. La marcia sempre più rapida di una tecnologia raffinata e suggestiva come la robotica non può non avere effetti profondi sull'immagine che abbiamo di noi stessi e sul nostro stesso essere umani: specchiandoci in quello straniante *alter ego* che sta diventando il robot, quale immagine ci ritorna? L'impresa della robotica, cioè la costruzione di un vero e proprio *uomo artificiale*, potrebbe darci, per analogia o per contrasto, indicazioni utili su di noi, così come ha fatto l'IA. In questa prospettiva di rispecchiamento il robot potrebbe essere un laboratorio di etica (artificiale)?
Infine si pone la questione del perché: *perché costruiamo i robot?* In certi casi particolari la risposta è ovvia: per eseguire compiti pesanti o pericolosi o ripetitivi, oppure per sostituire la manodopera umana con vantaggio economico o di rendimento. Ma tutto ciò non risponde alla questione di fondo: perché costruire macchine così simili a noi? Qualche risposta possibile: l'umanità sta facendo di tutto per entrare nel novero delle specie estinte e, sentendo prossima la fine dell'avventura, vuole lasciare un segno della propria grandezza, perciò costruisce macchine che possano sopravvivere e che ricordino a chi verrà (chi? Le macchine stesse?) un passato di gloria. C'è anche, come si è detto, l'orgoglio tutto umano di forzare e imitare i segreti della natura. Da ultimo c'è lo scopo comune a tutte le forme d'arte e di tecnica: stupire. "E' del poeta il fin la meraviglia", cantava Marino, e Leonardo annotò: "farò una finzione che significherà cose grandi".
Qualunque risposta diamo alla domanda di fondo, "perché?", è indubbio che da essa scaturiscono subito altre questioni che ne mettono in luce la natura socioculturale ed etica: quale società vogliamo costruire progettando i robot? Quali valori cerchiamo di rafforzare o di indebolire? Molti ricercatori non dimostrano alcun interesse per questi problemi e procedono tranquilli o entusiasti sulla strada dell'innovazione tecnica. Altri si

FIGURA 14
Nel 2003 il robot umanoide Krio, della Sony, ha diretto un concerto di musica classica a Tokyo

pongono in una prospettiva di breve respiro, conformandosi a codici simili alle leggi di Asimov. Altri ancora, una minoranza, si pongono nella prospettiva di medio e lungo termine e cercano di immaginare gli sbocchi possibili di quella che ormai è una vera e propria invasione dei robot. Qui le implicazioni della robotica e della roboetica si confondono con gli scenari elaborati in quell'attrezzatissima palestra di ipotesi sul futuro che è la fantascienza.
Il 13 marzo 2004, davanti a un folto pubblico di giovanissimi, l'orchestra filarmonica di Tokyo ha eseguito la *Quinta* di Beethoven sotto la direzione di KRIO, un robot umanoide della Sony, che, dopo qualche incertezza, ha fatto una discreta figura, aggiungendo un altro tassello al vasto mosaico delle attività umane eseguite (o imitate) dalle macchine (Figura 14). Ora, tanto per fare un esercizio di fantasociologia, mi immagino un nipotino di Krio che dirige un'orchestra di robot davanti a un pubblico di robot: se venissero a mancare gli umani chi si porrebbe le questioni di cui stiamo parlando? Dove andrebbe a finire il problema del senso? Chi si chiederebbe che cosa? E infine: dove andrebbe a finire la follia degli uomini? Che fine farebbero l'arte, l'umorismo, la trasgressione, la creatività, il gioco, il nonsenso? Chi potrebbe avvertire la differenza tra una lacrima e una goccia di pioggia? Forse, come ho detto sopra, per perpetuare la follia creativa dell'uomo, ci sarebbe bisogno di una macchina schizofrenica. Ma chi saprebbe costruirla, e chi, sapendola costruire, se ne assumerebbe la responsabilità?

Bibliografia

[1] Bateson G.: *Verso un'ecologia della mente*. Adelphi, Mlano, 2 edizione, 2000.

[2] Buttazzo G.: Coscienza artificiale: missione impossibile?. *Mondo Digitale*, n. 1, marzo 2002.

[3] Carlucci Aiello L., Dapor, M.: Intelligenza Artificiale: i primi 50 anni. *Mondo Digitale*, n. 10, giugno 2004.

[4] Fukuyama F.: *L'uomo oltre l'uomo*. Mondadori, Milano, 2002.

[5] Galván J.M.: *La robotica come speranza: la tecnoetica*. In: *La sfida del post-umano*, a cura di Sanna I., Ed. Studium, Roma, 2005.

[6] Longo G.O.: *Il simbionte: prove di umanità futura*. Meltemi, Roma, 2003.

[7] Longo G.O.: Uomo e tecnologia: una simbiosi problematica. *Mondo Digitale*, n. 14, giugno 2005.

[8] Losano M.G.: *Storie di automi*. Einaudi, Torino, 1990.

[9] Monopoli A.: *Roboetica*. (http://www.roboetica.it/page2.html).

[10] Saint-Exupéry A.: *Il Piccolo Principe*. Bompiani, Milano, 2000.

[11] Shelley M.: *Frankenstein, ovvero il moderno Prometeo*. Mondadori, Milano, 1983.

[12] Veruggio G.: *La roboetica e le sfide della rivoluzione robotica*. In: *La sfida del post-umano*, a cura di Sanna I., Ed. Studium, Roma, 2005.

[13] Veruggio G.: Il cammino della roboetica. *Le Scienze*, n. 461, gennaio 2007.

[14] Wiener N.: *The human Use of Human Beings*. Cybernetics and Society, Avon Books, New York, 1967.

[15] Zaccaria R.: Aspettando robot. *Mondo Digitale*, n. 7, settembre 2003.

Filmografia

2001: Odissea nello spazio, regia di Stanley Kubrick, Stati Uniti-Gran Bretagna, 1968.

AI: Intelligenza Artificiale, regia di Steven Spielberg, Stati Uniti, 2001.

Blade Runner, regia di Ridely Scott, Stati Uniti, 1982.

Grizzly Man, regia di Werner Herzog, Stati Uniti, 2005.

GIUSEPPE O. LONGO è ordinario di Teoria dell'informazione nella Facoltà d'Ingegneria dell'Università di Trieste. Si occupa di codifica di sorgente e di codici algebrici. Ha diretto il settore "Linguaggi" del Laboratorio della "International School for Advanced Studies" (Sissa) di Trieste e il Dipartimento di Informazione del "Centre Internationale des Sciences Mécaniques" (Cism) di Udine. Socio di vari Istituti e Accademie, s'interessa di epistemologia, di intelligenza artificiale e del rapporto uomo-tecnologia. È traduttore, collabora con il Corriere della Sera, con Avvenire e con numerose riviste. È autore di romanzi, racconti e opere teatrali tradotti in molte lingue.
E-mail: longo@univ.trieste.it

NUOVI SERVIZI A LARGA BANDA E TECNOLOGIE PER LA MOBILITÀ

Franco Mazzenga
Cristiano Monti
Francesco Vatalaro

La situazione nel campo delle telecomunicazioni appare oggi molto promettente in termini di tecnologie emergenti e di nuovi sistemi, specialmente wireless, che si affacciano sul mercato. L'articolo esamina il quadro complesso che evolve verso la cosiddetta "*Next Generation Network*" e la futura generazione radiomobile. L'obiettivo è concentrarsi sugli snodi tecnologici attualmente più accreditati, esaminando le caratteristiche di alcune delle tecnologie wireless che stanno affermandosi e che potrebbero fornire la struttura portante delle reti d'accesso degli anni 2010.

3.4

1. L'EVOLUZIONE DELLO SCENARIO DEI SERVIZI

In molti oggi ci chiediamo quali saranno i prossimi passi significativi dello sviluppo della Società dell'Informazione e, più in particolare, quale "futuro tecnologico" attenda il settore strategico dell'ICT (*Information and Communication Technologies*) negli ormai imminenti anni 2010. Sebbene la complessità degli scenari non ci consenta di fornire un quadro dettagliato, al contempo sembrerebbe ormai delinearsi una serie di snodi tecnologici nello sviluppo delle telecomunicazioni le cui direttrici essenziali appaiono identificabili sulla base delle innumerevoli linee di ricerca in corso negli Stati Uniti, in Europa e nell'Estremo Oriente. L'articolo cerca di rispondere al "perché" e al "come" la banda larga potrà presto incontrare la mobilità, senza le odierne limitazioni, completando negli anni 2010 un lungo percorso avviato negli anni '90 con la seconda generazione radiomobile (2G) del fortunatissimo GSM e poi proseguito negli anni 2000 con la più controversa terza generazione (3G).

L'abbondanza di banda è sempre più percepita come un fondamentale fattore abilitante del progresso economico-sociale da rendere disponibile in via generalizzata. La globalizzazione dei mercati, infatti, rende indispensabile operare efficientemente attraverso Internet e ciò vale per la quasi totalità delle imprese: la penetrazione degli accessi a banda larga (sia fissi che mobili) è destinata perciò a essere sempre più motore di crescita e di benessere.

D'altra parte sta avvenendo rapidamente la transizione dal binomio telefono portatile/personal computer all'unicità logica del computer d'individuo dotato di funzionalità multimediali, terminale personale indossabile che si articola in dispositivi con molteplici capacità sensoriali, oltre che dotati di funzioni di calcolo e comunicazione. È alle porte, cioè, la rete d'area fisica o BAN (*Body Area Network*) di dispositivi che, grazie all'abbondanza di indirizzi che sarà disponibile con l'avvento di IPv6, potranno essere presenti in rete e indirizzabili sia singolarmente che cumulativamente. La BAN sarà

parte di una rete virtuale globale sempre a disposizione dell'individuo che è always on e che, in virtù della tecnologia "IP mobile", si sposta in libertà mantenendo sempre lo stesso indirizzo senza mai abbandonare la rete virtuale globale.
Se questo scenario solo pochi anni fa poteva apparire vagamente fantascientifico e al più oggetto di studi e ricerche (per esempio, [1]), oggi costituisce ormai una parte della strategia della grande industria manifatturiera per la promozione di sistemi e servizi degli anni 2010 [2]. Anzi, se si allarga l'orizzonte ad accogliere ogni genere di oggetti, in particolare quelli dotati di mobilità, si osserva uno scenario ancora più ricco di attori: è stato stimato, ad esempio, che in Giappone nel 2010 si avrà un numero doppio di oggetti mobili rispetto alla popolazione [3]. Gli individui, quindi, non saranno i soli nodi mobili della rete virtuale globale: alle BAN si aggiungeranno reti d'area veicolare o VAN (*Vehicular Area Network*) che potranno così interagire in tempo reale con l'ambiente reale e virtuale: già oggi sono allo studio (e in parte implementate) reti *ad-hoc* di autoveicoli [4].
D'altra parte, per consentire l'efficiente interazione di BAN e VAN con Internet, gli ambienti del futuro presenteranno intelligenza immersa negli oggetti del mondo fisico, dotati di capacità sensoriali e interconnessi con la rete virtuale globale; l'ambiente intelligente sarà in grado di accrescere le facoltà degli individui presenti fino al punto di assistere in tempo reale gli stessi processi decisionali. Ciò richiede di immergere nell'ambiente fisico sistemi automatici per l'estrazione, l'elaborazione e il trasferimento delle informazioni, che dovranno anche essere autonomici (ossia in grado di configurarsi e manutenersi senza l'intervento umano) e consapevoli del contesto (in modo da adattare le condizioni del servizio al mutare delle condizioni operative).
Il mondo fisico e il mondo cibernetico tendono dunque a compenetrarsi in un "unicum informativo": l'informazione tende a divenire sempre più un bene condiviso attraverso le reti di sensori globali, e queste miriadi di sensori, prevalentemente connesse via rete wireless con tecnologie radio già in parte esistenti (per esempio, RFID, ZigBee ecc.), saranno in grado di prelevare e pre-elaborare enormi quantità di dati, oltre a gestire autonomamente il proprio stato e le connessioni con il mondo esterno.
Esaminiamo ora brevemente alcuni degli ambiti applicativi in cui questi trend tecnologici potranno farsi sentire attraverso la generazione di nuovi servizi potenzialmente dirompenti.

1.1. Creazione e fruizione in tempo reale di contenuti personali

Nel nuovo scenario tecnologico, i clienti del servizio divengono anche fornitori di contenuti: in altre parole, il ben noto paradigma non-gerarchico di comunicazione proprio di Internet si espande nel mondo fisico quotidiano e tutti i "nodi" della rete virtuale globale si comportano al contempo da produttori e fruitori di contenuti multimediali. Questa rete, potenzialmente a maglia completa, causerà la crescita rapida di traffico a banda larga in piena mobilità. I sintomi di questo "paradigm shift" sono oggi già presenti: si pensi per esempio ai casi di videoclip raccolti sulla scena con il telefono cellulare da passanti che si trovano casualmente in luoghi ove avvengono eventi critici (atti di terrorismo, gesti criminali, eventi calamitosi ecc.) e che li rendono disponibili in tempo reale ai mezzi di informazione e alle autorità.
Tutto ciò che oggi si fa con un computer, a casa o in ufficio, e anche molto di più, potrà essere effettuato in strada per mezzo della BAN. Se l'interazione individuo-BAN-ambiente avverrà secondo modalità naturali (ossia con gli strumenti del corpo e senza o con minima mediazione tecnologica), sarà spontaneo sia acquisire che trasferire nel mondo virtuale ogni tipo di informazione e chiedere assistenza remota per le più diverse attività umane: ciò determinerà l'esplosione del trasferimento dati nelle reti mobili.
Un'ulteriore spinta verso il processo di personalizzazione dell'erogazione dei contenuti, non più necessariamente in tempo reale, potrà avvenire con il progressivo cambiamento di paradigma del Web ordinario in "Web semantico" [5]. Possiamo prevedere a breve l'avvento di contenuti personalizzati, perfino assemblati dalla stessa "Rete" secondo un profilo dinamico dell'utente (le preferenze, il contesto, ma anche lo stato d'animo del momento, rilevati automaticamente dall'ambiente intelligente): per fare

un esempio semplice, si può pensare alla possibilità di erogare il "libro personalizzato", con un contenuto assemblato al momento e con una lunghezza adatta alle circostanze della fruizione (se sembra strano, si pensi all'odierno servizio offerto da http://www.dailylit.com/).

1.2. Streaming dei dati a tempo differito e contenuti personalizzati

Se a fianco della banda larga si tiene conto dell'abbondanza di memoria locale nella BAN, si può prevedere, in aggiunta alle modalità personali di creazione e fruizione dei contenuti in tempo reale di cui si è detto, anche l'avvento del nuovo paradigma del "più veloce del tempo reale", previsto da A. Odlyzko nel 2001, inteso come rapidissima, quasi istantanea, acquisizione di contenuti sull'unità periferica che li rende disponibili per l'utente finale in qualunque momento e su qualsiasi supporto, secondo necessità [6]. Ciò si connette anche agli studi in corso sulle architetture di comunicazione, dette "infostazioni", in cui l'acquisizione del contenuto avviene in modo intermittente, se e quando la copertura del servizio è disponibile.
La modalità tradizionale di consumo dei contenuti audio-video sta infatti evolvendo da un'attitudine meramente passiva, tipica del paradigma della diffusione in tempo reale, verso un modello attivo, come dimostrato dal successo del "peer-to-peer" specialmente tra le giovani generazioni: d'altra parte già oggi circa l'85% dei contenuti video distribuiti dalle stesse reti commerciali radio-televisive è costituito da materiale registrato [7]. Ciò determina una forte spinta verso il modello dello streaming a banda larga e alla costruzione di palinsesti personalizzati da fruire in tempo differito.

1.3. Virtualizzazione del mondo fisico

Le reti informative pervasive offrono anche nuove possibilità di gestione dei processi su scala planetaria: viste come enorme calcolatore distribuito, potranno essere usate per supportare i processi decisionali umani di qualunque tipo. Si può prevedere allora l'avvento della virtualizzazione, ossia della creazione di modelli virtuali del mondo reale [8]: grazie all'accumulo di informazioni, il mondo virtuale può generare modelli del mondo reale. Sarà così possibile, attraverso questi modelli, fare predizioni di trend futuri con lo strumento simulativo, estrarre informazione di valore mediante l'analisi comparata dei dati e derivare capacità di "problem solving" attraverso l'esame delle modalità operative dei modelli. L'esercizio della predizione a breve-medio termine attraverso l'evoluzione di modelli virtuali del reale che generano e simulano scenari alternativi in parallelo diverrà una modalità comune di supporto alle decisioni in ogni circostanza della vita quotidiana. Il controllo remoto di processi e sistemi complessi e distribuiti potrà perciò avvenire efficacemente e in tempo reale proprio in virtù della diffusione della banda larga ubiqua, oltre che della disponibilità di potenti risorse di calcolo distribuite.
Nel seguito dell'articolo si esaminano le tecnologie di accesso wireless abilitanti la banda larga necessaria all'avvento di questi nuovi servizi e di altri ancora non concepiti, nella prospettiva dei sistemi degli anni 2010.

2. ARCHITETTURE DI RETE D'ACCESSO

Per le reti di accesso wireless a banda larga sembra oggi generalmente condivisa oggi l'esigenza di sviluppare e standardizzare alcune architetture chiave, in grado di rispondere almeno ai seguenti requisiti:

- ❑ assicurare abbondante disponibilità di banda potenzialmente simmetrica sulla linea d'utente, ossia con capacità di traffico quanto più possibile eguale nei due sensi del collegamento;
- ❑ garantire l'interoperabilità tra tecnologie eterogenee per consentire, oltre al consueto handover orizzontale anche quello verticale tra standard diversi e la piena mobilità in tutti i tipi di ambiente (aperti, semiaperti, chiusi);
- ❑ consentire l'erogazione di servizi con logica di fornitura integrata, ben oltre le possibilità offerte dalla presente generazione tecnologica (per esempio connettività e accesso ai contenuti indipendenti dal tipo di terminale e dalla collocazione geografica), con configurabilità semplificata (di tipo *plug* & *play*), senza le consuete limitazioni dovute alla mobilità.

2.1. Le reti a banda larga di prossima generazione

I maggiori operatori mondiali di telecomunicazioni stanno preparando le prospettive di servizio citate prima attraverso giganteschi piani di innovazione tecnologica della rete che, a partire dal 2008 e per tutti gli anni 2010, dovrebbe acquisire la fisionomia completamente nuova della cosiddetta NGN (*Next Generation Network*), caratterizzata dalla piena migrazione in tecnologia IP non solo della rete di trasporto ma anche della rete di accesso (All IP). Secondo questi piani, la NGN è la rete completamente a pacchetto, in grado di fornire i servizi di telecomunicazione per mezzo di un gran numero di tecnologie a banda larga (paradigma della "rete agnostica"), assicurando i livelli richiesti di qualità di servizio (QoS) e in cui le funzioni connesse a ciascun servizio vengono ad essere indipendenti dalla specifica struttura tecnologica sottostante. Una NGN abilita l'accesso indistinto da parte degli utenti alle reti e ai fornitori di servizio e supporta la mobilità con indistinguibilità, nella percezione del grado di servizio, tra l'ambito operativo fisso e quello mobile. È evidente che la definitiva migrazione tecnologica (dal circuito al pacchetto) che sarà compiuta con la NGN è la logica conseguenza della forte progressione già da tempo in atto nel trasferimento dei servizi su piattaforma IP, accelerata dall'avvento del servizio VoIP (*Voice over IP*) ma che ben presto interesserà anche gli altri servizi svolti in tempo reale (per esempio la videotelefonia e la videoconferenza).

In aree metropolitane a forte e media densità di traffico si può prevedere il dispiegamento di una struttura ottica ad alta velocità, detta GPON (*Gigabit Passive Optical Network*), almeno fino all'armadio di strada che serve un numero limitato di condomini (tipicamente alcune centinaia di utenze residenziali), mentre la rete di accesso, nella sua componente wireless, quantomeno a livello di coperture urbane in aree aperte e semiaperte, potrà basarsi sull'impiego di soluzioni *Radio over Fibre* (RoF) già introdotte da tempo ma che solo negli ultimi anni stanno venendo a maturazione tecnologica e applicativa [9]. La radio su fibra è una tecnologia adatta ad irradiare e a captare segnali a radiofrequenza (RF) per mezzo di *Remote Access Unit* (RAU) collocate negli armadi di strada. Ne consegue che tutte le funzioni di elaborazione di segnale (codifica/decodifica, multiplazione/demultiplazione, generazione dei segnali a RF, cancellazione dell'interferenza ecc.) sono eseguite nella cosiddetta *Central Station* (CS) collocata entro la rete di trasporto ottica, di norma ben lontano dalle RAU, e non più, come oggi è consueto, nella stazione base tipica dei sistemi cellulari. Molte stazioni centrali possono perciò essere concentrate in pochi siti (che prendono la denominazione significativa di "Hotel di stazioni base"), riducendo fortemente i costi di manutenzione e di approvvigionamento delle scorte.

Pertanto, il segnale che transita fra la CS e la RAU ha natura ottica e, nel caso più semplice, la RAU include soltanto i convertitori ottico/elettrico e viceversa, gli amplificatori a RF (quello di potenza e quello a basso rumore, rispettivamente in emissione e in ricezione), un diplexer e l'antenna. Centralizzando nella CS tutte le complesse funzioni di elaborazione di segnale, le piccole RAU sono semplici e poco costose. Poiché, inoltre, il trasporto ottico è trasparente e la banda disponibile è amplissima, un vantaggio ulteriore di questa architettura è rappresentato dalla possibilità di trasportare assieme segnali che rispondono a molteplici standard. La tecnologia RoF si presta dunque a facilitare la convergenza tecnologica tra sistemi cablati e wireless e, per gli operatori che forniscono un'offerta integrata fisso/mobile, agevola l'evoluzione graduale dalla 3G alla prossima generazione radiomobile. Soluzioni RoF con trasporto da/verso l'armadio di strada in tecnologia WDM (*Wavelength Division Multiplex*) sono considerate le estensioni naturali della rete di trasporto per integrare in modo ottimale l'accesso a banda larga con le infrastrutture metropolitane basate sugli anelli ottici esistenti (Figura 1).

Nel contempo è prevista anche l'evoluzione delle reti cellulari, la cosiddetta LTE (*Long Term Evolution*), con lo sviluppo delle prime reti a partire dal 2009, basata su una nuova interfaccia radio OFDM (*Orthogonal Frequency Division Multiplex*) con velocità di trasmissione di molte decine di Mbit/s con terminali in mobilità veicolare. Anche per la LTE, che dovrà interlavorare con le attuali reti mobili a standard 3GPP (3rd *Generation Part-*

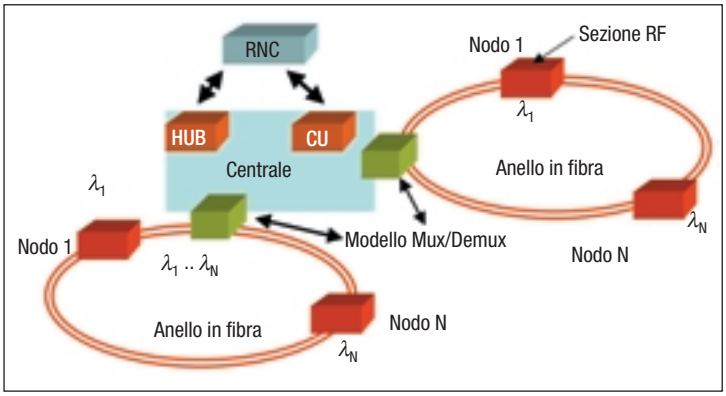


FIGURA 1
Esempio di architettura d'accesso in area metropolitana

nership Program), l'architettura di rete è a pacchetti del tipo All IP.

2.2. Le reti rurali wireless

È improbabile che una o poche architetture d'accesso possano risultare adatte alle condizioni che si presenteranno nei diversi ambiti operativi. Sono, perciò, in fase di messa a punto piani per l'allestimento di reti rurali, volte soprattutto a contrastare il fenomeno del digital divide infrastrutturale, ossia il ritardo che si sperimenta in vaste aree geografiche nella realizzazione degli accessi a banda larga, che rischia di divenire persino crescente con l'avvento della NGN (e della LTE) nelle aree metropolitane. In Italia la banda larga è indisponibile in quasi tutti i Comuni inferiori a diecimila abitanti. Da un punto di vista logico, le architetture per le reti rurali distinguono tre elementi principali:

- ❑ dorsale di trasporto (o *backbone*);
- ❑ connessione locale (o "ultimo miglio");
- ❑ distribuzione negli ambienti interni degli edifici.

La dorsale ha il compito di veicolare il traffico da/verso i Comuni non raggiunti dal servizio a banda larga. Essa realizza la connessione nelle località che dispongono di POP (*Point Of Presence*), ossia punti di accesso a banda larga ad Internet, pubblici o privati, per poi renderla disponibile nei Comuni dove essa non è presente. La dorsale può essere realizzata con mezzi cablati (fibra ottica) o in ponte radio (ad esempio attraverso gli standard HiperLan, WiMAX, ecc.) o con soluzioni ibride.

All'interno di ciascun Comune, il *backbone* si può attestare per esempio in corrispondenza di un edificio pubblico (per esempio, il Municipio); sul territorio comunale la connessione è resa disponibile in modalità wireless con l'utilizzo di apparecchiature che impiegano soluzioni Wi-Fi, HiperLan2 o WiMAX.

Affinché si possa utilizzare il servizio, il soggetto gestore della rete deve dotare l'utente dell'hardware di trasmissione occorrente, detto CPE (*Customer Premises Equipment*), e del kit di configurazione software. A seconda della tecnologia radio scelta per l'ultimo miglio e delle condizioni di ricezione nel sito di utente, si possono avere configurazioni differenti nella distribuzione locale: la CPE può essere in esterno (outdoor) e in tal caso la distribuzione interna avviene via cavo; alternativamente, in taluni casi è possibile ricevere il segnale all'interno dell'edificio con CPE collocata in indoor, in prossimità del personal computer dell'utente. Nel caso di ricezione comunitaria, infine, l'apparecchiatura ricevente è opportunamente collocata sul tetto della sede dell'utente (per esempio, sede della Pubblica Amministrazione, impresa o condominio in caso di utenza residenziale) e la connettività individuale all'interno dell'edificio viene assicurata comunque via cavo (con possibile riconversione in wireless anche a copertura parziale Wi-Fi).

Una possibile architettura implementativa è riportata nella figura 2: in essa si distingue un *backbone* di trasporto realizzato tramite anello ottico in tecnologia Giga o MPLS seguito da una sezione Ethernet in ponte radio (HiperLan, ponti radio SDH ecc.). Segue la connessione periferica, in rete wireless attuabile per mezzo di qualsiasi tecnologia disponibile, tra cui HiperLan2, Wi-Fi, WiMax, e, infine, la rete di distribuzione indoor di cui si è detto.

2.3. Alternative nelle architetture di rete wireless

Per ottimizzare il trasporto e la distribuzione nelle reti extraurbane a banda larga, sono anche all'esame nuove architetture wireless che rappresentano un'evoluzione delle infrastrutture di accesso tipiche dei sistemi cellulari. Basate sulla trasmissione *Punto-MultiPunto* (PMP), come è noto, le architetture cellulari prevedono che una stazione base serva una molteplicità di terminali nell'area di copertura

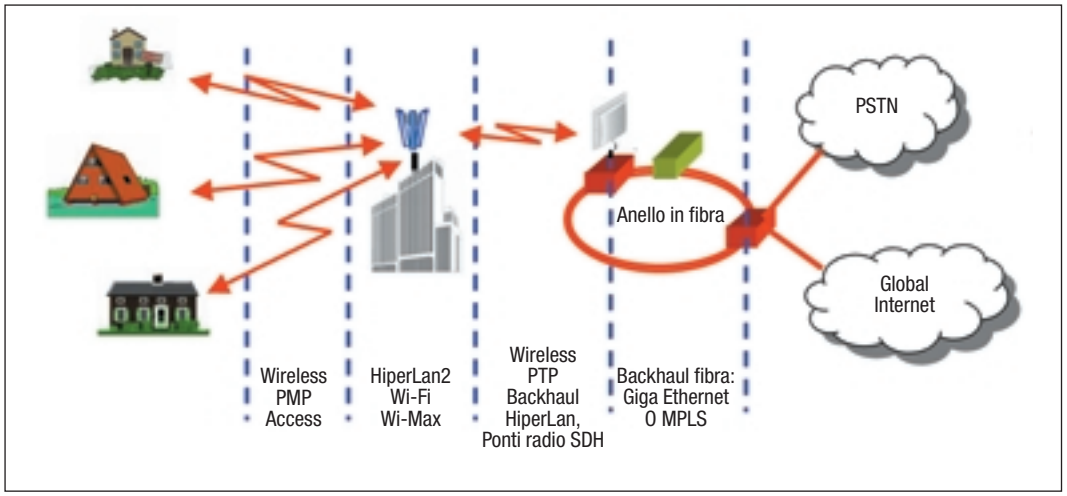


FIGURA 2
Esempio di architettura d'accesso in area rurale

ad essa assegnata. La configurazione cellulare ha consentito, anche in Paesi che come l'Italia presentano ardue condizioni orografiche, la copertura pressoché totale a livello nazionale per i servizi in fonia in tempi brevi (soltanto qualche anno), ma non si presta altrettanto bene a coperture generalizzate per servizi a banda larga, almeno per il prossimo futuro.
Per aumentare con la classica infrastruttura cellulare la capacità di traffico che è richiesta dai nuovi servizi a banda larga occorrerebbe allestire un ragguardevole numero di Stazioni base, a cui corrisponderebbero alti costi (sia CAPEX che OPEX) e difficoltà serie per l'operatore di garantirsi un ritorno degli investimenti in tempi generalmente considerati accettabili (ciò spiega anche, almeno in parte, l'avvento delle reti rurali a banda larga di cui si è detto).
La ricerca di nuove soluzioni di architettura di rete wireless è dunque volta alla riduzione dei costi sia di allestimento che di manutenzione, specialmente nelle zone rurali e in piccoli comuni isolati, oltre alla rapidità di installazione dei punti di accesso. Una prima soluzione, proposta nell'ambito del gruppo di lavoro IEEE 802.16j, considera reti di accesso in cui sono presenti unità che eseguono la memorizzazione e il successivo rilancio dei pacchetti (relay) trasmessi da un utente verso la stazione base che serve l'area.
Un esempio di *Rete Wireless Relay* (WR), è mostrato nella figura 3 **A**: tutti i percorsi di comunicazione nella rete WR si devono chiudere sull'unica stazione base servente e le procedure che realizzano l'inoltro del traffico in tale rete richiedono di agire soltanto a livello

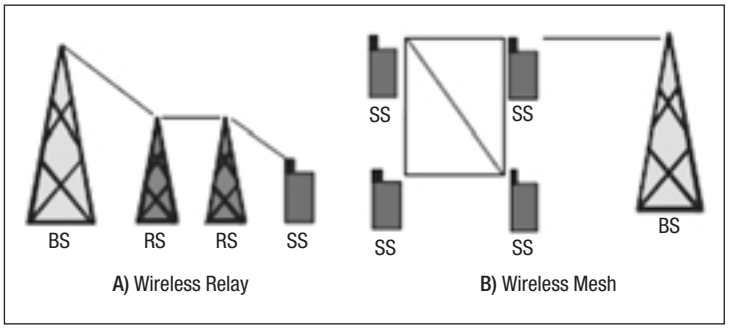


FIGURA 3
Architetture di rete:
A) *Wireless Relay e*
B) *Wireless Mesh*

MAC (*Medium Access Control*), senza prevedere meccanismi di instradamento dei pacchetti. Tra le caratteristiche principali di un'architettura di rete WR citiamo:
- la compatibilità all'indietro, ovvero la possibilità per un terminale fisso o mobile conforme allo standard preesistente di operare senza modifiche nella nuova struttura di rete con rilancio dei pacchetti;
- l'inammissibilità di percorsi di comunicazione diretti tra terminali (come per le reti cellulari convenzionali);
- la flessibilità operativa della *Stazione Relay* (*Relay Station*, RS) che può essere fissa, trasportabile o mobile, può essere completamente trasparente o può memorizzare i pacchetti degli utenti ad essa collegati per inoltro successivo (store and forward);
- la possibilità di inoltro simultaneo del pacchetto attraverso percorsi di comunicazione diversi a partire da RS diverse che coprono lo stesso terminale (Cooperative Relay), che consente di migliorare le prestazioni radio attraverso l'uso di tecniche di diversità in ricezione alla stazione base.

Una rete WR pertanto consente di estendere la copertura soprattutto in aree geografiche ampie, migliorare il throughput e la qualità di servizio (QoS) anche per utenti che in una convenzionale rete PMP sono collocati al bordo della cella, aumentare la capacità di sistema, salvaguardare la durata delle batterie dei terminali e, infine, ridurre complessità e costi del sistema soprattutto in fase di avvio della rete.
A fronte dei vantaggi, per questo tipo di reti esistono anche limitazioni, principalmente legate alla struttura topologica ad albero. In primo luogo, nell'ipotesi in cui tutti i collegamenti abbiano la stessa capacità di trasmissione e supponendo di allocare le risorse in modo equo tra i nodi, si ha un frazionamento della banda disponibile nel transito attraverso il singolo nodo dell'albero: di conseguenza la banda offerta agli utenti serviti dalla generica RS dipende dalla profondità dell'albero e quindi dal numero di salti che il pacchetto inviato da un utente verso la stazione base deve compiere (di regola, alberi con profondità superiore a tre sono considerati impraticabili). In secondo luogo, se un collegamento lungo l'albero va fuori servizio, parte dell'area di copertura che fa riferimento alla stazione base potrebbe risultare non coperta e non è possibile reinstradare i pacchetti verso altri nodi per la ritrasmissione.
Per ovviare, almeno in parte, a questi due inconvenienti si può considerare un'architettura di rete a maglia (mesh), in cui i nodi RS realizzano a loro volta funzioni di instradamento dei pacchetti. Di conseguenza l'architettura evolve in quella di figura 3 **B**, rappresentativa di un caso particolare di una *Rete Wireless Mesh* (WM).
Con riferimento alla figura 3 si osserva che le funzionalità di tutti o parte dei nodi relay che realizzano la sottorete, devono essere estese dalla semplice operazione di rilancio (bridging) all'operazione di instradamento (routing) con conseguente aumento della complessità realizzativa e dei costi. Per gestire al meglio la complessità di questa rete solitamente si richiede che i router che la compongono siano in grado di:
- ❑ autoriconfigurarsi, soprattutto quando un nuovo apparato viene inserito all'interno della rete già operante (caratteristica plug & play);
- ❑ reinstradare in modo ottimale i pacchetti degli utenti tenendo conto dello stato di occupazione dei singoli collegamenti al fine di garantire agli utenti la desiderata qualità del servizio, QoS (scheduling ottimo delle risorse);
- ❑ reinstradare i pacchetti anche in caso di guasto di uno o più nodi della rete (il cosiddetto self-healing).

In una rete Wireless Mesh come quella riportata nella figura 3 **B** i terminali di utente accedono soltanto attraverso i nodi fissi della rete; tuttavia, in alcune architetture di rete può essere previsto che i terminali siano in grado di realizzare in modo spontaneo una rete tra di loro sulla base di esigenze specifiche del momento. Tale rete può essere autonoma da quella dell'operatore e, siccome in generale non ha un'architettura predefinita, viene anche detta rete *ad-hoc*. Le reti *ad-hoc* con nodi mobili sono allo studio da tempo in virtù della loro naturale adattabilità a operare in ambienti fortemente dinamici, in cui non è prevista la presenza di un'infrastruttura pre-esistente, come ad esempio in caso di catastrofi naturali o in altre situazioni d'emergenza, ma per servizi che richiedono coperture wireless a banda larga permanenti presentano per lo più interesse secondario.

3. TECNOLOGIE PER L'INTERFACCIA RADIO

Nei sistemi futuri la necessità di permettere un trasporto con alti valori di ritmo binario richiede l'impiego di tecniche di trasmissione adatte a operare correttamente anche in presenza di severe condizioni del canale. Uno degli aspetti principali riguarda, perciò, la scelta dell'interfaccia radio che deve rispondere a vari requisiti, tra cui:
- ❑ supportare alti valori di ritmo binario (in movimento almeno 100 Mbit/s nel downlink e 50 Mbit/s nel uplink; almeno 1 Gbit/s in condizioni stazionarie), garantendo al contempo un'alta efficienza nell'uso dello spettro;
- ❑ fornire un livello fine di granularità del ritmo binario nell'accesso al mezzo, per facilitare un adattamento efficiente tra i canali sull'interfaccia radio e i canali di trasporto sovrastanti;
- ❑ consentire scalabilità di traffico, anche realizzando combinazioni di traffico a burst e continuo;

❑ essere flessibile per adattarsi a differenti scenari (ambienti indoor e outdoor, configurazioni con microcelle e con macrocelle ecc.);
❑ consentire di operare con terminali aventi caratteristiche eterogenee;
❑ supportare collegamenti wireless a corto raggio realizzati con varie tecnologie;
❑ interoperare con sistemi a standard differente;
❑ presentare bassa complessità ed essere robusta per operare in condizioni severe della propagazione e dell'interferenza.

Pur senza addentrarsi nelle molteplici soluzioni tecnologiche proposte per lo strato fisico delle reti wireless del prossimo futuro, sembra opportuno concentrarsi su alcune tra le principali soluzioni che potrebbero garantire il rispetto dei necessari requisiti.

3.1. Radio cognitiva

Per soddisfare alle esigenze dei servizi futuri occorre disporre di risorsa spettrale in abbondanza e a basso costo, specialmente nella banda 1-3 GHz che è oggi quella più contesa e, in prospettiva, anche al di sotto di 1 GHz, per avvantaggiarsi del cosiddetto "dividendo digitale" che presumibilmente potrebbe essere ricavato dalla prevista maggiore efficienza spettrale connessa alla numerizzazione della televisione commerciale.

Come è noto l'approccio attuale nella regolamentazione dell'uso dello spettro è tale per cui ai radiosistemi sono assegnate bande di frequenza fisse, con limiti prefissati nelle potenze di emissione che hanno lo scopo di ridurre la portata radio. Così facendo, il fattore di utilizzo medio di alcune bande è basso a causa dell'allocazione inefficiente su vaste aree geografiche a servizi che le sottoutilizzano, oltre che per il ricorso a ampie bande di guardia, che può risultare ormai eccessivo in virtù del progresso tecnologico. Risale al lontano 1994 l'intuizione di P. Baran della necessità di un ripensamento radicale della politica di allocazione dello spettro [10]: da allora si sono fatti molti passi avanti in questa direzione e oggi negli Usa alcuni movimenti d'opinione sostengono molto energicamente la cosiddetta politica dell'open spectrum [11]. Anche la Commissione europea da tempo esamina il problema di tecniche innovative per un uso più efficiente dello spettro [12].

Se l'attuale meccanismo regolamentare ancora si basa sul principio di garantire protezione contro le interferenze in modo statico ipotizzando l'impiego di ricevitori privi di intelligenza, la tecnologia tra breve renderà disponibili soluzioni adattative, sia per il contenimento che per la cancellazione delle interferenze, che aprono la porta a modalità nuove di impiego compartecipato dello spettro: una soluzione pratica interessante, su cui si sta lavorando in Europa, è lo spectrum leasing, ossia l'affitto di porzioni spettrali da parte del legittimo assegnatario sotto condizioni di non interferenza [13].

Un uso più flessibile della risorsa spettrale implica che gli apparati radio debbano essere dotati di funzioni di monitoraggio dello spettro e che siano in grado di adattarsi immediatamente alle condizioni della trasmissione su una banda anche molto larga (adaptive radio). Se al concetto di radio adattativa si affianca quello legato all'analisi dell'ambiente radio e alla capacità di apprendimento si parla di cognitive radio. Alcune della caratteristiche di una radio cognitiva includono la determinazione della posizione, il monitoraggio dell'uso dello spettro anche con l'ausilio di dispositivi vicini, il cambio di frequenza, il controllo di potenza e, infine, l'alterazione dei parametri e delle caratteristiche della trasmissione; tutto questo dovrebbe determinare le condizioni per un uso adattativo dello spettro cui potrà corrispondere un significativo incremento di efficienza spettrale. Infatti, scegliendo di trasmettere in accordo con un insieme di regole atte ad evitare le interferenze e a garantire una preassegnata qualità di servizio, una radio cognitiva attua l'uso dinamico delle bande di frequenza, identificando e impiegando in modo opportunistico le regioni spettrali non utilizzate o sottoutilizzate.

L'approccio della cognitive radio potrebbe incidere fortemente sulle regole future di impiego dello spettro. Per esempio in Nord America è attiva l'iniziativa IEEE 802.22 tesa a definire una nuova interfaccia radio WRAN (*Wireless Regional Area Network*) per l'accesso a banda larga (Internet, trasporto di dati, fonia e streaming video) in aree rurali e remote con tecniche radio cognitive nelle bande di guardia televisive non utilizzate (il cosiddetto "white space"), avvantaggiandosi

delle caratteristiche particolarmente favorevoli della propagazione in VHF e nella parte bassa della UHF.

3.2. Antenne adattative

L'adattatività è un concetto che si sta affermando ai vari livelli di progetto dei sistemi radio; storicamente, forse il primo ambito applicativo è quello delle antenne. Per migliorare la qualità del collegamento radio fisso o mobile in presenza di fading già gli attuali sistemi 3G consentono l'impiego di tecniche di trasmissione che sfruttano la disponibilità di schiere di antenne per la formazione del fascio (o beam-forming), ovvero sistemi di antenne con caratteristiche di irradiazione adattative che ne consentono l'orientamento del fascio in una specifica o in un insieme specifico di direzioni. In questo modo oltre a dirigersi in modo da catturare i segnali desiderati e, reciprocamente, da irradiare nella direzione richiesta, le antenne adattative possono sopprimere l'interferenza che proviene da altri siti trasmittenti. Le antenne adattative rappresentano una tecnica molto efficace per migliorare il riuso di frequenza e quindi la capacità di traffico delle reti a banda larga.

3.3. Comunicazioni opportunistiche

I vantaggi dell'impiego compartecipato delle risorse radio, già esaminati con riferimento all'uso cognitivo dello spettro, si considerano oggi anche sotto altri punti di vista, attraverso i nuovi concetti di "comunicazioni opportunistiche". Il mezzo radio è soggetto al fenomeno del fading dovuto all'interferenza, per lo più distruttiva, tra raggi elettromagnetici che percorrono cammini diversi e che si compongono vettorialmente (scattering): questo fenomeno, che genera fluttuazione del livello di segnale nel tempo (ma anche in spazio e in frequenza), secondo l'approccio classico, viene considerato un difetto della trasmissione a cui porre rimedio. Mentre nei sistemi a banda stretta occorre ricorrere all'imposizione di un margine di potenza fisso, talvolta anche grande, e/o al controllo automatico di potenza (o di guadagno), i sistemi a banda larga possono adoperare tecniche di equalizzazione adattativa basate sulla stima del canale.
Tuttavia, se si è in grado di trasmettere solo quando l'interferenza dei raggi elettromagnetici risulta costruttiva, il fading può addirittura conferire un guadagno e non determinare una perdita[1]. Se, poi, il canale è usato in compartecipazione tra molti utenti e il fading agisce indipendentemente sui diversi collegamenti, si può pensare di adoperarlo sistematicamente a vantaggio di quel collegamento che risulti istantaneamente nelle condizioni più favorevoli, ottenendo così un guadagno che viene detto da "diversità multiutente" senza che ne discenda la riduzione del tempo di utilizzo, e quindi della capacità di sistema. Pertanto, in un collegamento tra stazione base e molti terminali mobili, la stazione base può "opportunisticamente" trasmettere in ogni intervallo di tempo disponibile verso quel terminale che vede in condizioni più favorevoli (Figura 4).
La riduzione della potenza media di trasmissione che ne consegue, determina minore interferenza e quindi un maggiore aggregato di dati trasmessi per unità di tempo (più alto va-

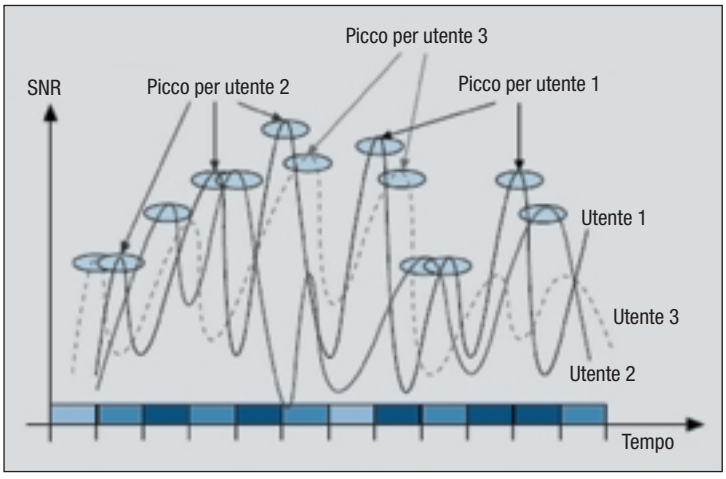


FIGURA 4
Concetto di guadagno per diversità multiutente

[1] Ciò richiede naturalmente interventi rapidi basati sulla stima a breve termine del comportamento del sistema di trasmissione, differentemente dalle tecniche che eseguono medie sul *fading*, come per i sistemi MIMO di cui si dirà più avanti. Si può osservare, di passaggio, un'interessante e curiosa analogia con il caso della compravendita di azioni che può farsi nel lungo periodo (filtrando così le fluttuazioni veloci del mercato) oppure con il sistema del *trading online* quasi istantaneo che sfrutta proprio correlazione a breve termine e dinamica limitata delle fluttuazioni dei titoli delle aziende quotate in Borsa.

lore del throughput di sistema). Più ampia è la dinamica delle fluttuazioni, più pronunciati saranno i picchi e maggiore può risultare il guadagno di diversità multiutente: inoltre, qualora l'ambiente presenti scattering modesto oppure le fluttuazioni siano tanto lente da non consentire al sistema di attendere che si raggiunga il picco, è anche possibile indurre di proposito le fluttuazioni alla stazione base.

3.4. Modulazione e codifica adattative

Un'altra tecnica dinamica usata è la modulazione adattativa, in virtù della quale il trasmettitore può cambiare il formato di modulazione del segnale in relazione alle condizioni del canale: quando la qualità del collegamento è buona si possono usare schemi di modulazione come la 64-QAM (64-quadrature amplitude modulation), che presenta efficienza spettrale di 6 bit/s/Hz; se durante la trasmissione le condizioni del canale peggiorano, si può cercare di mantenere inalterata la qualità di trasmissione cambiando la legge di modulazione, per esempio passando al QPSK (*Quadrature Phase Shift Keying*), con efficienza spettrale di 2 bit/s/Hz. Infatti, al cambiamento di formato di modulazione è associata la variazione di valore del rapporto tra la potenza utile e la potenza disturbante (*SNR*) necessario per conseguire il richiesto valore di tasso di errore (*BER*). A parità di specifica, modulazioni con più bassa efficienza spettrale richiedono *SNR* relativamente basso (per esempio si hanno 9 dB nel caso del QPSK per *BER* pari a 10^{-6}); modulazioni più efficienti a parità di *BER* richiedono valori di *SNR* considerevolmente più elevati (circa 22 dB nel caso del 64-QAM).

A questa tecnica può agevolmente associarsi anche la codifica di canale nelle sue due varianti della correzione d'errore o FEC (*Forward Error Correction*) e della richiesta di ritrasmissione o ARQ (*Automatic Retransmission reQuest*). La codifica adattativa, ad esempio può realizzarsi in modalità FEC con codici convoluzionali in cui viene modificato il ritmo di codifica, oppure, nei casi in cui la trasmissione non sia sensibile al ritardo, sostituendo al codice convoluzionale classico un codice turbo che prevede l'uso di un canale di retroazione.

Come conseguenza, l'impiego della modulazione e codifica adattativa determina la variabilità del raggio di copertura, come mostra la figura 5: giacché la potenza ricevuta decresce con la distanza, se la potenza trasmessa è costante a parità di potenza di disturbo, utenti più vicini alla stazione base misurano valori elevati di *SNR* e possono quindi utilizzare formati di modulazione con efficienza spettrale più elevata (16-QAM o 64-QAM); d'altra parte gli utenti più lontani sono generalmente soggetti a valori più bassi di *SNR* e quindi possono essere raggiunti con modulazioni meno efficienti (BPSK o QPSK). Discende come conseguenza la dipendenza del ritmo binario dalla distanza, causa principale della riduzione di velocità di download da Internet che si sperimenta nell'allontanarsi dalla porta d'accesso al servizio.

3.5. Modulazioni multiportante

Un canale radiomobile a banda larga, B, è caratterizzato sia da un comportamento non stazionario, dovuto alla mobilità di uno o entrambi i terminali, che dal *fading* causato dalla multipropagazione dovuta agli ostacoli; perciò la funzione di trasferimento tempo-variante $H(f, t)$ è rappresentata per mezzo di un modello di canale *fading* selettivo in tempo e in frequenza. Idealmente, $H(f, t)$ dovrebbe apparire quasi costante al segnale in transito, sia in tempo che in frequenza: ciò significa che il massimo sparpagliamento dei ritardi (o delay spread) risulterebbe molto minore del tempo di simbolo, $T_S \approx 1/B$, e rispettiva-

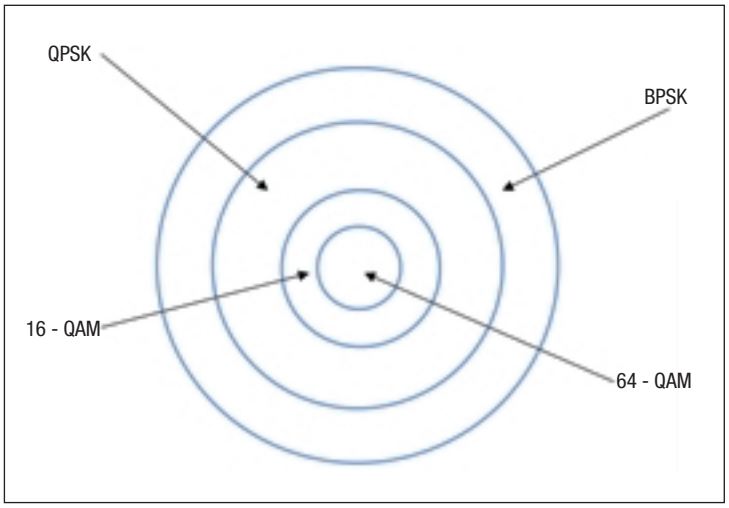


FIGURA 5
Modulazione adattativa e copertura

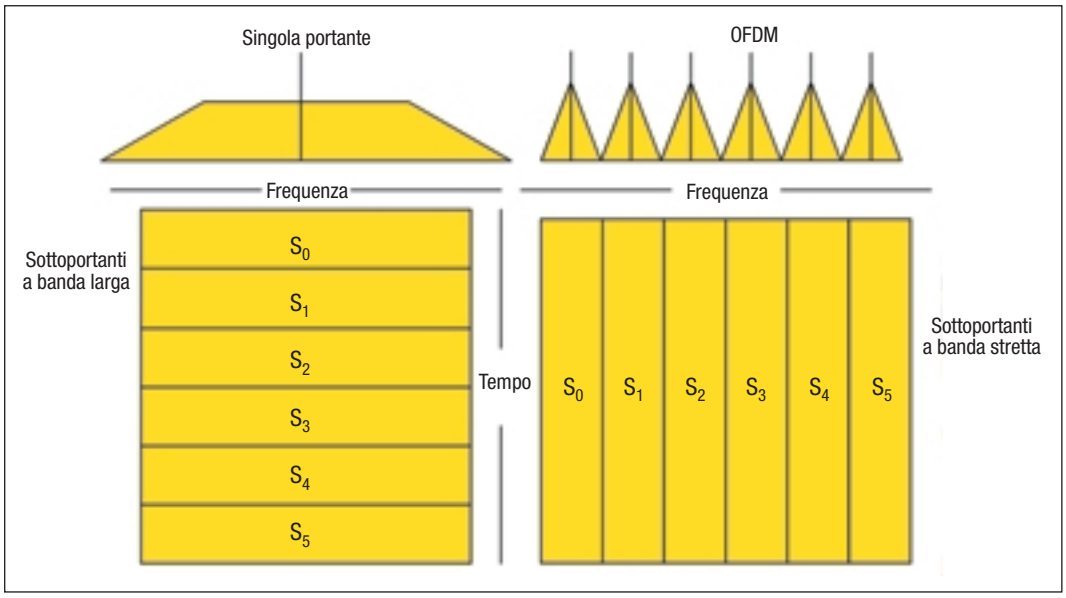


FIGURA 6
Trasmissioni a portante singola e multipla

mente il tempo di coerenza (inverso dello sparpagliamento Doppler) sarebbe molto più grande di T_S.
Ciò non si verifica nella trasmissione a banda larga, e quindi se si usa un sistema di trasmissione convenzionale a singola portante l'equalizzazione di canale può anche risultare assai complessa; inoltre nel canale tempo-variante anche la stima del canale deve essere eseguita più frequentemente e quindi cresce l'overhead relativo. Le potenzialità che presenta un sistema multiportante, flessibile e adattativo come l'OFDM (*Orthogonal Frequency Division Multiplex*), si mostrano particolarmente utili in canali radio tempo-varianti e selettivi in frequenza.
Come è noto, il principio dell'OFDM consiste nel dividere la banda B in N parti, in modo da ottenere canali a banda stretta, non selettivi in frequenza, di larghezza B/N; in ciascuna sottobanda sono trasmessi dati a ritmo binario ridotto (di N volte) che sono modulati su portanti ortogonali (i segnali modulati perciò si sovrappongono dando luogo a efficienza spettrale ottima). La figura 6 confronta i due casi di trasmissione sequenziale di segnali su singola portante e di trasmissione multiportante OFDM su canali in parallelo.
In virtù dell'ortogonalità tra le sottoportanti, idealmente non si ha interferenza tra i canali in parallelo; inoltre, grazie all'uso di un opportuno tempo di guardia, T_G, associato ad ogni simbolo (detto "prefisso ciclico"), si elimina anche l'interferenza intersimbolo (ISI). In pratica, ciò si mantiene vero purché T_G sia stato scelto maggiore del massimo valore di delay spread previsto e il tempo di simbolo nel canale a banda stretta $T_S' = N\,T_S$ risulti minore del tempo di coerenza.
Sotto queste condizioni l'equalizzazione di ciascun canale a banda stretta si riduce a stimare un solo coefficiente complesso del canale per ciascuna sottoportante. La stima si può eseguire inserendo un simbolo pilota noto nel formato dati in trasmissione: sulla base dei simboli pilota il ricevitore può stimare la funzione di trasferimento di ogni canale in ogni istante per interpolazione e ogni segnale di sottoportante può essere demodulato coerentemente. Alternativamente, può adottarsi la demodulazione differenziale che evita l'uso dei simboli pilota. Risultati sperimentali con valori di ritmo binario di oltre 100 Mbit/s hanno riportato *BER* minori di 10^{-4} in canale affetto da *fading* veloce e *delay spread* di circa 2 μs nella banda dei 5 GHz, con terminali in movimento alla velocità di oltre 200 km/h. [14].

3.6. Sistemi multiantenna

Un approccio di diversità di spazio tramite antenne multiple, sia trasmittenti che riceventi, nel canale wireless è in grado di estrarre il segnale con caratteristiche variate e in condizioni quanto più possibile indipendenti su più vie in parallelo. In origine la diversità di spazio veniva applicata solo in ricezione

per ottenere repliche del segnale che, in virtù dell'indipendenza dei modi di trasmissione, con alta probabilità non sono soggette allo stesso trattamento da parte del mezzo trasmissivo e quindi, sempre con alta probabilità, non sono tutte soggette a condizioni di fading profondo.
Il modello di trasmissione multicanale MIMO (*Multiple-Input Multiple-Output*) [15] attua la diversità di spazio con l'impiego di una molteplicità di elementi d'antenna, realizzando una schiera d'antenna composta di N elementi nel lato emittente e di altri M nel lato ricevente (Figura 7). Se si considera il segnale in uscita alla j-esima porta lato ricezione per effetto del segnale all'entrata dell'i-esima antenna lato emissione (assumendo inattive tutte le altre porte d'entrata $k \neq i$), si può definire il guadagno di segnale g_{ij} e, ripetendo la procedura per tutte le porte d'entrata e d'uscita, si costruisce la matrice dei guadagni di canale $[g_{ij}]$. In un sistema TDD (*Time Division Duplex*) con separazione temporale tra emissione e ricezione del terminale (il cosiddetto duplex time) minore del tempo di coerenza del canale questa informazione è posseduta dall'emettitore, in quanto il canale è reciproco. Nel caso FDD (*Frequency Division Duplex*), in cui la risorsa di trasmissione viene suddivisa tra emissione e ricezione nel dominio della frequenza, deve esistere un canale di ritorno dal ricevitore al trasmettitore per fornire esplicitamente a quest'ultimo l'informazione di stato del canale, detta CSI (*Channel State Information*).
Sono casi speciali di una configurazione MIMO le seguenti architetture:
- SIMO (*Single-Input Multiple-Output*), caratterizzato da $N = 1$, $M > 1$;
- MISO (*Multiple-Input Single-Output*), caratterizzato da $N > 1$, $M = 1$.

Nel SIMO si attua la diversità in ricezione attraverso tecniche di elaborazione di segnale nella schiera ricevente: la diversità in ricezione è il caso classico che si attua attraverso le ben note tecniche di selezione o di combinazione del segnale.
Nel MISO, viceversa, si attua la diversità soltanto in emissione. La diversità in emissione è realizzata attraverso varie tecniche molto studiate al giorno d'oggi. La MISO può attuarsi sia attraverso una tecnica ad anello chiuso che attraverso una tecnica ad anello aperto: la prima, che può considerarsi duale della diversità in ricezione, prende la forma di elaborazione di schiera in emissione; viceversa, la diversità in emissione ad anello aperto è un approccio relativamente nuovo che consente al progettista di spostare il carico di elaborazione dal terminale alla stazione base ove è collocata la schiera d'antenna.

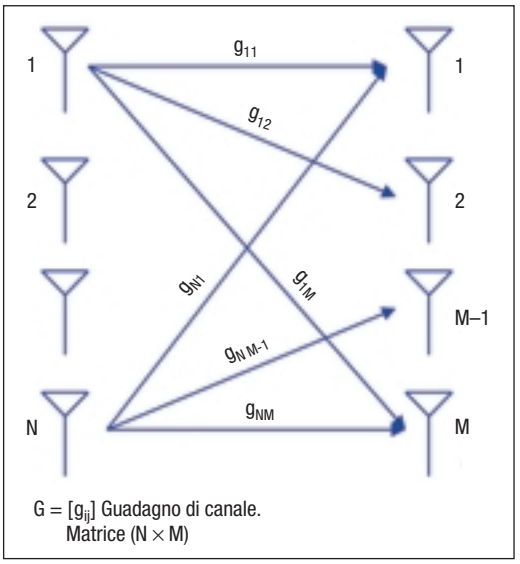


FIGURA 7
Configurazione MIMO e matrice dei guadagni di canale

In un sistema MISO si può avere completa o limitata CSI, ovvero si può operare in assenza di tale informazione: per completa CSI si intende la piena conoscenza, istante per istante, della matrice di canale MIMO $[g_{ij}]$. Il principale problema associato alla diversità di trasmissione ad anello aperto consiste nella mancanza di conoscenza della CSI. Pertanto si deve adottare una tecnica di codifica di canale robusta, in grado cioè di garantire buone prestazioni per un'ampia casistica di condizioni del canale. Nel caso dell'anello aperto sono possibili tre diverse implementazioni:
- trasmissione ripetuta, che prevede che ogni antenna trasmetta la stessa informazione accompagnandola con una univoca forma d'onda di firma;
- diversità di ritardo ove ogni antenna trasmette la stessa informazione in tempi differenti e non è necessaria alcuna forma d'onda di firma;
- codifica spazio-tempo o STC (*Space-Time Coding*) in cui, oltre all'impiego della diversità spaziale, i simboli vengono codificati senza alcuna informazione proveniente dal ricevitore.

La STC combina i benefici della codifica FEC e

della diversità di spazio ma, a differenza delle consuete implementazioni del FEC, non implica allargamento di banda, in quanto la ridondanza è applicata nello spazio attraverso le diverse antenne e non in tempo o in frequenza. Un esempio di STC è fornito dal ben noto codice di Alamouti (N =2, M =1) [16].
Un tipo speciale di diversità, infine, consiste nella multiplazione spaziale con antenne multiple (MIMO-SM) che attua la trasmissione di segnali differenti da ognuna delle antenne che compongono la schiera. In questo caso il ricevitore combina i segnali captati dalle varie antenne dopo averli pesati, e ricostruisce infine i segnali ricevuti, trattando gli altri segnali come interferenti.

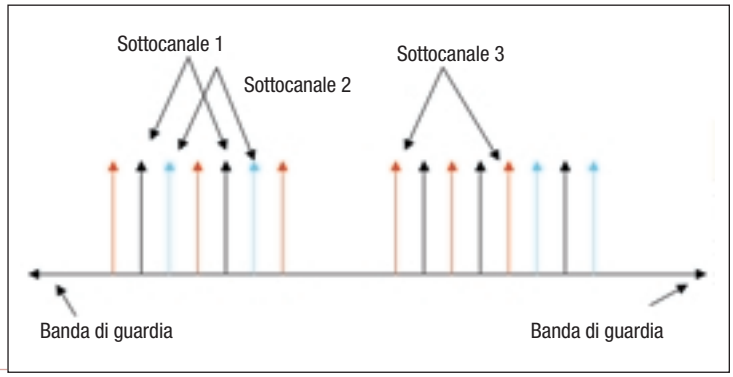
4. TECNOLOGIE PER L'ACCESSO MULTIPLO

La flessibilità di impiego dell'OFDM si manifesta anche nelle possibilità che apre in relazione all'accesso multiplo al canale di trasmissione. Tra le varie possibilità si hanno le seguenti:
• OFDM -TDMA (*Orthogonal Frequency Division Multiplexing - Time Division Multiple Access*);
• OFDMA (*Orthogonal Frequency Division Multiple Access*).

4.1. Accesso multiplo a divisione di tempo

Nel sistema d'accesso OFDM-TDMA, tutte le sottoportanti sono attribuite ad un dato utente per un periodo di tempo solitamente multiplo intero del periodo di simbolo OFDM. Pertanto un utente può avere disponibilità dell'intera banda ciclicamente, ossia ogni tempo di trama, T_F (allocazione statica), oppure può ricevere un numero variabile di simboli OFDM per trama in base al requisito di banda (allocazione dinamica). Secondo questo protocollo di accesso, le operazioni, da un lato, di modulazione e multiplazione dei flussi generati dal terminale (che sono oggetto dell'OFDM) e, dall'altro, quella di accesso multiplo, che avviene secondo la classica disciplina della divisione di tempo (TDMA), risultano di fatto disgiunte. Il vantaggio principale di questo metodo consiste nella riduzione del consumo delle batterie del terminale per effetto della intermittenza del funzionamento. Per una maggiore flessibilità di gestione della risorsa radio si è introdotta la disciplina OFDMA che si esamina nel seguito.

FIGURA 8
Sottocanalizzazione OFDMA



4.2. Accesso multiplo con sottocanalizzazione

Nel caso dell'OFDMA le sottoportanti sono distribuite tra gli utenti che, in linea di principio possono trasmettere in qualsiasi istante. L'OFDMA può supportare un certo numero di flussi dati per utente di uguale ritmo binario, o anche differenti (per esempio assegnando a ciascuno un numero differente di sottoportanti). Tenuto conto delle differenti condizioni dello specifico canale di trasmissione si possono adottare diversi schemi di modulazione sulla singola sottoportante tra cui QPSK, 16-QAM, 64-QAM.
Si può ricorrere a due diversi metodi di allocazione delle sottoportanti; si ha:
• l'allocazione statica, se le sottoportanti sono assegnate all'utente per tutta la durata della connessione; il metodo è semplice ma, quando una o più sottoportanti è soggetta a fading, la degradazione di prestazione che ne consegue può persistere anche per tempi relativamente lunghi;
• l'allocazione dinamica, se si dispone di informazione quasi istantanea delle condizioni del canale (CSI) per ciascun utente; questo metodo prende il nome di DSA (D*ynamic Subcarrier Allocation*) e si può attuare in svariati modi.
Nella DSA l'informazione di CSI è usata in tempo reale per assegnare le sottoportanti più adatte ad ogni collegamento (tecnica di sottocanalizzazione), in modalità contigua o interallacciata (Figura 8). In quest'ultimo caso, ossia nella sottocanalizzazione interallacciata, che è il più efficace in termini di guadagno di diversità, la sincronizzazione dell'intero segnale OFDMA diviene più difficile. Un'ul-

teriore penalizzazione della DSA consiste nella necessità di aggiuntivo carico di segnalazione (overhead) ogniqualvolta si debbano riassegnare le sottoportanti.
Grazie alla sottocanalizzazione la potenza trasmessa si concentra su un numero limitato di portanti, incrementando la potenza disponibile per singola portante e aumentando così la portata radio, ovvero contrastando le perdite di penetrazione attraverso le pareti degli edifici. Il numero e la posizione delle portanti che realizzano il sottocanale di trasmissione può cambiare su base burst o su base trama e ciò consente una gestione dinamica della risorsa di radiotrasmissione sia nel dominio del tempo che nella frequenza.
Una variante è rappresentata dal cosiddetto SOFDMA (*Scalable* OFDMA); nel SOFDMA si assume che la spaziatura tra le sottoportanti sia costante indipendentemente dal passo di canalizzazione, così se cambia la larghezza di banda del segnale da trasmettere ci si limita a modificare il numero di sottoportanti ospitate all'interno della banda.
Quando non sia possibile, oppure non sia conveniente, rendere disponibile in emissione l'informazione di CSI, un'alternativa alla DSA consiste nella tecnica di salto di sottoportante o SCH (*Sub-Carrier Hopping*) che rappresenta l'applicazione all'OFDMA del salto di frequenza. Pertanto, la tecnica SCH-OFDMA è un modo di combinare le caratteristiche dell'OFDMA con quelle dello spread-spectrum. Quando l'utente salta casualmente sulle sottoportanti disponibili si ha il cosiddetto RSCH-OFDMA (*Random Sub-Carrier Hopped* OFDMA): un esempio di questa tecnica è rappresentato dal cosiddetto FLASH-OFDM proposto per lo standard in itinere IEEE 802.20. Come di consueto per le tecniche di salto di frequenza, anche in questo si distingue tra FSCH-OFDMA (dove la prima lettera indica *Fast*) e SSCH-OFDMA (*Slow*), a seconda che il salto di frequenza possa, o non possa, rispettivamente avvenire tra simboli contigui.

5. CONCLUSIONI

L'articolo ha esaminato alcune tra le più promettenti tecnologie di interfaccia radio che si stanno affermando e che potrebbero caratterizzare le evoluzioni dei sistemi verso la Next Generation Network delle reti fisse e la Quarta Generazione delle reti mobili.
Nell'ambito di un quadro di progressiva convergenza delle reti e dei servizi si possono prevedere notevoli sovrapposizioni tra gli standard e tra le funzionalità delle reti. Tuttavia, è anche evidente che qualsiasi tentativo di unificazione tecnologica sarà inevitabilmente destinato all'insuccesso. Non resta che prevedere un futuro complesso in termini di alternative in competizione nell'ambito di una struttura "All IP" per il resto largamente agnostica in relazione alle specifiche soluzioni tecnologiche a livello sia di strato fisico che di collegamento. Al contempo è prevedibile il successo di soluzioni a bassa complessità di interfaccia con l'utente o, come spesso si dice, di tipo "Plug & Play", con l'obiettivo ultimo dell'interazione naturale e priva di mediazione con l'utente finale.
Per questo motivo possiamo aspettarci l'avvento di soluzioni di integrazione verticale tra le alternative tecnologiche, sulla base di standard di interfaccia che ne agevolino l'implementazione. Questo modo di procedere dovrebbe determinare l'avvento di tecnologie cosiddette "a prova di futuro" che potranno evolvere in un continuum tecnologico in cui lo stesso concetto di "generazione" che si è affermato in passato finirà per perdere completamente di significato.

Bibliografia

[1] Vatalaro F.: Il paradigma delle comunicazioni immersive virtuali. *Rivista AEIT*, Ottobre 2006, p. 28-31.

[2] Ryhänen T., Huopaniemi J.: *Fusing the digital and the physical: future mobile experience*. Nokia Technology Media Briefing, Oct. 3, 2006.

[3] Nakajima N.: Future Mobile Communications Systems in Japan. *Wireless Personal Comm.*, Vol. 17, 2001, p. 209-224.

[4] Hull B., et al.: *CarTel: A Distributed Mobile Sensor Computing System*. MIT Computer Science and Artificial Intelligence Laboratory.

[5] Berners-Lee T., et al.: *The Semantic Web*. Scientific American, May 2001 http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

[6] Odlyzko A.: Content is not King. *First Monday*, Vol. 6, n. 2, February 2001, http://firstmonday.org/issues/issue6_2/odlyzko/index.html

[7] Key4biz, Intervista a Vinton Cerf, 3.10.2005. http://www.key4biz.it/cgi-bin/key4biz/k4b.cgi?id_testo=515526525873381490061792 46&area_tematica=&a_z=v_t&p_d_

[8] Nakamoto H., Komeichi M.: I*T Road Map toward 2010*. Nomura Research Institute, NRI papers, n. 102 March 1, 2006, http://www.nri.co.jp/english/opinion/papers/2006/pdf/np2006102.pdf

[9] Cooper A.J.: Fibre/radio for the provision of cordless/mobile telephony services in the access network. ELECTRON. LETT., Vol. 26, 1990, p. 2054-2056.

[10] Baran P.: *Visions of the 21st Century Communications: Is the Shortage of Radio Spectrum for Broadband Networks of the Future a Self Made Problem?*. 8-th Annual Conf. on Next Generation Networks, Washington, DC, Nov. 9, 1994.

[11] http://www.greaterdemocracy.org/framing_openspectrum.html

[12] *Spettro radio: una politica strategica per l'Unione europea – seconda relazione annuale*. Comunicazione della Commissione al Consiglio e al Parlamento europeo, 6.9.2005.

[13] Weiss T.A., Jondral F.K.: *Spectrum Pooling: An Innovative Strategy for the Enhancement of Spectrum Efficiency*. IEEE Radio Comm., March 2004, p. S8 -S14.

[14] Yoshida M.: *OFDM Transmission for ISI Channels Using Variable-Length Pilot Symbols and pre-FFT Equalizer with Enhanced MRC Diversity Reception*. GLOBECOM 2003, p. 2290-2294.

[15] Ajib W., Haccoun D.: *An Overview of Scheduling Algorithms in MIMO-Based Fourth-Generation Wireless Systems*. IEEE Network, Sept./Oct. 2005, p.43-48.

[16] Alamouti S.M.: A Simple Transmit Diversity Technique for Wireless Communications. *IEEE J. Sel. Areas Commun.*, Vol. 16, n. 8, Oct. 1998, p. 1451-1458.

FRANCO MAZZENGA si è laureato in Ingegneria Elettronica presso l'Università di Roma Tor Vergata ed è attualmente Professore associato di Telecomunicazioni presso il Dipartimento di Ingegneria Elettronica della stessa Università. Ha lavorato presso la Fondazione Ugo Bordoni e il Consorzio di Ricerca in Telecomunicazioni (CoRiTeL). È direttore tecnico del Consorzio Università Industria - Laboratori di Radiocomunicazioni (RADIOLABS), Roma. È autore di più di 70 pubblicazioni in atti di congressi e riviste internazionali. I suoi principali interessi scientifici riguardano i sistemi e le reti wireless, l'elaborazione numerica dei segnali e la teoria statistica dei segnali.
E-mail: mazzenga@ing.uniroma2.it

CRISTIANO MONTI si è laureato in Ingegneria Elettronica presso l'Università di Roma Tor Vergata. Collabora con Telespazio nel settore relativo all'Innovazione Tecnologica. Ricercatore presso il consorzio Radiolabs, i suoi interessi di ricerca e le sue pubblicazioni si riferiscono ai sistemi di accesso radio, alla modellizzazione del canale a radiofrequenza e alle tecniche di localizzazione.
E-mail: cristiano.monti@uniroma2.it

FRANCESCO VATALARO Professore ordinario di Telecomunicazioni presso l'Università di Roma "Tor Vergata", ha oltre 25 anni di esperienza nell'industria e nell'università. È Presidente del Consorzio Università Industria - Laboratori di Radiocomunicazioni (RADIOLABS), Roma. È chairman del IEEE Vehicular Technology/Communications Society Italy Chapter e membro del Comitato direttivo della IEEE Italy Section. Membro di numerosi comitati scientifici e di redazione, è autore di oltre 150 pubblicazioni e i suoi principali interessi scientifici sono nelle comunicazioni e reti wireless.
E-mail: vatalaro@uniroma2.it

ICT E INNOVAZIONE D'IMPRESA
Casi di successo

Rubrica a cura di
Roberto Bellini, Chiara Francalanci

La rubrica *ICT e Innovazione d'Impresa* vuole promuovere la diffusione di una maggiore sensibilità sul contributo che le tecnologie ICT possono fornire a livello di innovazione di prodotto, di innovazione di processo e di innovazione di management. La rubrica è dedicata all'analisi e all'approfondimento sistematico di singoli casi in cui l'innovazione ICT ha avuto un ruolo critico rispetto al successo nel business, se si tratta di un'impresa, o al miglioramento radicale del livello di servizio e di diffusione di servizi, se si tratta di una organizzazione pubblica.

Analisi dei casi di successo dovuti alle tecnologie ICT nell'Innovazione di Business

Roberto Bellini

1. Introduzione

Dopo aver analizzato ed approfondito, all'interno di questa rubrica, sette casi aziendali in cui l'ICT ha svolto un ruolo centrale nel raggiungimento del successo, sembra opportuno cercare di estrarre da queste esperienze alcune indicazioni di carattere generale sui fattori che hanno portato ogni società esaminata al successo (Tabella 1).
In questa sede adotteremo una metodologia di riferimento per l'interpretazione dei fattori di successo basata sul concetto generale di "*processo di innovazione*"; il materiale di base utilizzato sarà costituito da tutti i casi studiati e pubblicati fino ad oggi, dei quali verrà considerato e valutato solo ciò che risulta utile per l'analisi di approfondimento dei fattori che hanno portato all'innovazione di business imputabile all'ICT.

2. Cosa intendiamo per innovazione e imprenditorialità

Prima di procedere con l'analisi, definiamo alcuni concetti che ci aiuteranno a condividere le riflessioni sui casi considerati e a selezionare i fattori critici di successo.
In termini molto sintetici, possiamo dire che la sede in cui si sviluppa la ricerca è il laboratorio, privato o pubblico e che l'obiettivo della ricerca applicata è quello di sviluppare modelli teorici della realtà ed eventualmente prototipi ispirati agli stessi modelli.
La sede in cui si sviluppa l'innovazione è invece l'impresa: secondo Schumpeter l'innovazione è un fondamentale processo economico che "*combina in modo nuovo fattori di produzione tradizionali*".
Questo processo economico può dare vita ad una invenzione, allo sfruttamento di una risorsa naturale, all'attuazione di un'idea già applicata in un mercato differente, o ancora alla riprogettazione e al miglioramento di un prodotto già introdotto nel mercato.
L'osservazione dei comportamenti economici delle imprese di successo evidenzia quali sono le condizioni che meglio favoriscono la crescita, attraverso la capacità di produrre un insieme di innovazioni continue: si tratta delle condizioni presenti nel libero mercato, le quali spingono tutti gli attori verso la competitività; infatti:
- da una parte l'attività innovativa nelle economie di libero scambio è assolutamente necessaria come criterio di sopravvivenza delle imprese;
- dall'altra, le nuove tecnologie si sviluppano molto più velocemente perché, in un sistema liberista, è possibile remunerare economica-

	Nome azienda	Settore di industria - prodotti offerti – ruolo tecnologia ICT	Mondo Digitale
1	ODM	Servizi alle imprese e alle persone – informazioni professionali sulle retribuzioni e la gestione delle risorse umane – ruolo ICT globale	n. 14; p. 70-75
2	TEkNO	Manifatturiero elettromeccanico – verricelli e sistemi di trazione elettrici – ruolo ICT per la pianificazione e il controllo	n. 15; p. 75-82
3	Esprinet	Distribuzione prodotti digitali a imprese e consumatori-ruolo ICT per l'ottimizzazione dei processi e della gestione	n. 16; p. 71-76
4	Concessionari Alfa Romeo	Distribuzione auto a imprese e consumatori – ruolo ICT per il supporto integrato ai processi di marketing e di vendita	n. 17; p. 72-78
5	Hotel Solutions	Servizi alle imprese del turismo, settore alberghiero – gestione prenotazioni e ricavi (Yield Management) – ruolo ICT globale	n. 18; p. 79-84
6	Tarasconi Trasporti	Servizi di trasporto intermodale – ruolo ICT per il supporto ai processi gestionali nel trasporto e informazioni sulla viabilità e sul traffico	n. 19; p. 79-84
7	Funambol	Servizi alle imprese e ai consumatori di telefonia mobile – ruolo ICT globale	n. 20; p. 73-78

TABELLA 1
Casi di successo analizzati

mente gli innovatori affinché condividano le loro conoscenze.
In particolare, per quanto riguarda le tecnologie ICT, è opportuno precisare tre diversi ruoli che queste ultime possono svolgere rispetto all'innovazione:

❑ *Innovazione di prodotto*: la tecnologia ICT può svolgere un ruolo di innovazione sia per i prodotti/servizi basati su ICT che per quelli basati su tecnologie *non ICT*. La tecnologia ICT genera innovazione di prodotto quando:

- permette di creare un nuovo pacchetto di software destinato alla vendita ad imprese o consumatori in associazione all'attività di assistenza;
- permette di realizzare un nuovo servizio per imprese o consumatori in settori come quello finanziario, turistico, dei servizi al pubblico ecc;
- permette di introdurre intelligenza o possibilità di localizzazione in prodotti *non ICT* come le macchine utensili, gli elettrodomestici e in generale le attrezzature per la casa e l'ufficio, oppure di localizzare un'auto o un carro ferroviario ecc..

❑ *Innovazione di processo*: la tecnologia ICT genera innovazione di processo quando permette di realizzare servizi tradizionali e non, che in passato erano svolti con procedure manuali, mediante procedure automatiche; esempi sono le procedure bancarie e assicurative, le procedure dei servizi pubblici e dei servizi professionali alle imprese ecc..

❑ *Innovazione gestionale*: infine, la tecnologia ICT può fornire innovazione gestionale quando migliora la gestione amministrativa e operativa dell'impresa, la pianificazione e il controllo, la monitorizzazione del livello di funzionamento dei processi piuttosto che dello stato amministrativo e operativo di clienti, fornitori, partner ecc..
Nel caso in cui le tecnologie ICT contribuiscano sia alla configurazione/erogazione del prodotto/servizio che all'automazione del processo di erogazione, parliamo di *tecnologie di produzione dei servizi vendibili*; nel caso in cui le tecnologie ICT apportino un loro contributo a tutti i tre livelli di innovazione, parliamo di *ruolo ICT globale*.
Baumol argomenta che esistono diverse tipologie di imprenditori, tuttavia ciò che fa la differenza fra l'uno e l'altro è *l'intensità dell'energia che ognuno di loro utilizza nel produrre innovazione*, giacché questo fattore potenzia la crescita economica: quest'ultima dipende dagli incentivi messi in campo dal sistema economico.
Nel seguito, per classificare gli imprenditori dei casi analizzati, parleremo di livello dell'*energia di innovazione*.
L'*imprenditorialità*, infine, è l'atto di fare innovazione, non è quindi qualcosa di correlato alla scienza o alla ricerca, ma piuttosto al cambiamento delle regole del gioco nella competizione economica. Fra queste regole del gioco rientrano naturalmente il fatto che ci sia un tipo di prodotto/servizio che soddisfi un bisogno espresso da un mercato e che il valore che la clientela

è disposta a riconoscere in termini di prezzo sia sufficiente a coprire i costi delle possibili modalità di produzione, di erogazione e di promozione e i costi del capitale impegnato.
Proviamo quindi ad evidenziare le caratteristiche dei sette casi considerati, scegliendo come fattori critici di successo quelli che le teorie dell'innovazione considerano rilevanti:
a. il tipo di prodotto e il tipo di mercato;
b. il livello competitivo del nuovo prodotto/servizio;
c. il ruolo dell'imprenditore e l'energia di innovazione che questi esprime;
d. il ruolo della tecnologia nel processo di innovazione;
e. gli altri fattori critici di successo che emergono dall'analisi.

3. Analisi dei fattori critici di successo dei casi considerati

3.1. Caso ODM

Il successo del caso *ODM* è dovuto all'introduzione di un'innovazione nell'area dei servizi professionali per le imprese e alla conseguente innovazione di processo. Come società di consulenza infatti, abituata a sviluppare una relazione con il cliente molto profonda ma anche molto destrutturata, ODM decide di sfruttare la competenza acquisita nell'area della gestione delle risorse umane e delle retribuzioni in particolare, lanciando una serie di servizi, con un alto livello di standardizzazione e supportati da un altrettanto alto livello di automazione. Questi servizi sono disegnati tenendo conto della concorrenza di alcune grandi multinazionali che operano nello stesso settore.
Il ruolo dell'imprenditore è fondamentale sia per la definizione del prodotto/servizio offerto che per l'acquisizione di dati su base gratuita attraverso un servizio di valutazione della retribuzione messo a disposizione della totalità dei dipendenti delle imprese italiane. L'offerta del nuovo servizio ODM causa una immediata riduzione della soglia di ingresso nel mercato dei servizi di analisi retributiva e permette che il servizio venga offerto non solo alle multinazionali, ma anche alle piccole e medie imprese. L'energia di innovazione dell'imprenditore è determinante soprattutto nella fase di lancio del servizio, quando si comprende cosa ne limita il decollo: il "ritardo culturale" dei potenziali clienti rispetto all'uso delle tecnologie informatiche. Infatti, numerosi amministratori delle piccole/medie imprese che gestiscono direttamente le risorse umane, risolvono il problema facendo ricorso ad una tradizionale modalità di erogazione, che fa uso di supporti cartacei, ignorando la possibilità di utilizzare supporti elettronici.
Il ruolo della componente tecnologica ICT si sviluppa su tre piani: permette di realizzare il nuovo prodotto/servizio ed è quindi una componente dell'innovazione di prodotto; permette di sviluppare tutta la produzione e distribuzione del prodotto/servizio via web e di automatizzarne la gestione con un alto livello di integrazione delle varie fasi di lavoro, in modo da fornire agli addetti ODM una visione completa del processo, dal momento in cui il cliente ordina il prodotto al momento in cui lo riceve e lo utilizza. Infine, il nuovo sistema di gestione permette di monitorare l'ordine e la consegna del Report Retributivo, di produrre la fattura sulla base del servizio consegnato e di monitorare la chiusura della transazione.
Per consolidare il business acquisendo un portafoglio di circa 1.500 clienti e una banca dati di 1,5 milioni di record retributivi, l'imprenditore impegna un'energia di innovazione di circa sei anni.

3.2. Caso TEkNO

Il successo del caso *TEkNO* è imputabile all'introduzione di un'innovazione di prodotto e di un'innovazione di processo: Tekno aggiunge alla produzione di verricelli per conto terzi - cioè come subfornitore - la produzione e la commercializzazione diretta di un "sistema elettrico di trazione" basato sulle stesse competenze ma realizzato attraverso un processo di produzione diverso. Alla complessità di gestione di due linee di produzione si aggiunge anche la difficoltà nell'affrontare, riconoscere e servire un mercato differente.
Il ruolo della componente tecnologica ICT deve essere concepito come quello di una classica innovazione gestionale, in cui l'obiettivo raggiunto, è quello di integrare nella pianificazione e nel controllo della produzione, la gestione dei sistemi di produzione sia dei verricelli che dei sistemi di trazione, soprattutto verso la rete di fornitura. L'innovazione gestionale è certamente molto importante dal punto di vista del miglioramento del sistema di pianificazione e di controllo della produzione, ma risulta relativamente meno importante rispetto all'innovazione di prodotto.

Mentre il ruolo della coppia di imprenditori è fondamentale per quanto riguarda l'innovazione di prodotto e la capacità di sviluppare un nuovo mercato, l'innovazione gestionale con l'introduzione del sistema ICT di supporto alla pianificazione e al controllo della produzione viene seguita direttamente dal responsabile del sistema informativo, che è responsabile anche della qualità e dei processi.
I due imprenditori coinvolti esprimono comunque la loro energia di innovazione solo negli ultimi cinque/sei anni, dato che per i precedenti venti anni si sono occupati solo del lavoro di subfornitura.

3.3. Caso Esprinet

Il successo del caso *Esprinet* è dipeso dalla capacità imprenditoriale che ha portato allo sviluppo, nel corso di alcuni anni, di un sistema distributivo integrato che permette di "vedere" tutta la filiera del prodotto, dal momento in cui viene ordinato dal punto vendita locale al momento in cui viene consegnato, fatturato e incassato su uno qualunque dei nove canali attraverso i quali il prodotto raggiunge il cliente finale, consumer o business. La rete distributiva Esprinet, a differenza di quella dei concorrenti, viene indirizzata sia verso il mercato business che verso il mercato consumer. Altri due fattori concorrono al successo di Esprinet:
❑ la capacità di adeguamento ai cambiamenti del mercato dei prodotti digitali, i quali vengono continuamente rinnovati, sia per il business che per il consumatore, e i cui costi sono in costante diminuzione. In questa situazione, la sopravvivenza della distribuzione è garantita solo dalle sofisticate soluzioni di riduzione dei costi di gestione dei vari canali;
❑ il ruolo di distributore puro che Esprinet svolge per i venditori internazionali e nazionali (in quanto non produce e vende i propri prodotti sul mercato).
Anche in questo caso la tecnologia ICT svolge un ruolo essenziale di innovazione di processo e di gestione, a cui l'imprenditore aggiunge componenti di servizio per i suoi clienti (i distributori finali della rete) sempre basati su tecnologie ICT.

3.4. Caso Rete Concessionari Alfa Romeo

Il successo del caso *Rete Concessionari Alfa Romeo*, relativo alla distribuzione del bene durevole "auto", è stato determinato dalla decisione presa dalla casa madre, in particolare dalla direzione commerciale e *marketing*, di rendere disponibili tutte le informazioni sui clienti e sul mercato, alle piccole imprese commerciali facenti parte della propria rete. L'implementazione dei servizi di gestione del portafoglio clienti a favore di ciascun operatore, delegata ad una società di consulenza esterna con forti competenze di *marketing*, ha permesso di rendere disponibili tutti i dati di profilatura della clientela e di monitoraggio delle azioni commerciali dei singoli venditori e di reinterpretarli in termini di *benchmark* della produttività commerciale misurata anche rispetto alla segmentazione della clientela di ciascun operatore della rete.
Per quanto riguarda il contributo della tecnologia ICT, siamo di fronte ad un'innovazione di processo e di gestione che agisce sulla catena del valore della distribuzione e dell'assistenza tecnica.
Sono rilevanti in questo caso tutti i ruoli degli imprenditori coinvolti: il direttore della divisione della casa madre, il titolare della società di consulenza che sviluppa i servizi di *marketing* e ogni singolo operatore della rete (concessionario). Il processo su cui si agisce è quello di vendita e l'oggetto dell'analisi è il cliente finale, del quale si vogliono conoscere tutte le sfumature del comportamento di acquisto e di utilizzo del bene. In questo caso l'energia di innovazione degli imprenditori coinvolti è molto alta, dato che tutto il processo di innovazione viene realizzato e installato in pochi mesi (tre o quattro) e l'operatività del sistema si stabilizza nel corso di un anno.

3.5. Caso Hotel Solutions

Il successo del caso *Hotel Solutions* è derivato dalla realizzazione e commercializzazione, a partire dal 2005, di:
❑ un prodotto, costituito da un pacchetto software che sviluppa modelli previsionali basati sulle prenotazioni, in grado di ottimizzare la ricettività del singolo albergo su un orizzonte di sei mesi e di massimizzare i ricavi attraverso opportune politiche di prezzo, realizzate tenendo conto dei prezzi della concorrenza per alberghi che offrono ricezione in perimetri turistici comuni. Se il cliente acquista la licenza d'uso del prodotto è di conseguenza a suo carico la gestione delle prestazioni del sistema;
❑ una serie di servizi in *outsourcing* accessibili via web da qualunque operatore alberghiero ne faccia richiesta e che sia d'accordo nel far tran-

sitare le sue prenotazioni attraverso il sistema di Hotel Solutions.
In entrambi i casi il sistema è al servizio di una nuova figura professionale che prende il nome di Revenue Manager, il quale opera presso la struttura alberghiera. Il sistema è particolarmente adatto alle grandi e grandissime catene alberghiere, che tendenzialmente si orientano all'acquisizione del prodotto; per le piccole reti alberghiere e le singole gestioni è molto più conveniente acquistare i servizi in *outsourcing*.
L'imprenditore punta principalmente al business in *outsourcing*, lasciandosi comunque aperta la possibilità di vendere anche il prodotto, come farebbe tipicamente una software house; il ruolo dell'imprenditore è molto delicato perchè si tratta di un business innovativo che deve affermarsi in un Paese dove manca una cultura informatica relativa a questa tipologia di servizi e che potrebbe trovarsi in competizione con altri sistemi a livello internazionale; l'obiettivo del pareggio (*break even*) è previsto al terzo anno di attività (2007).
Il ruolo della tecnologia ICT in questo caso è essenziale su tutti i tre piani dell'innovazione: considerando l'offerta in outsourcing, contribuisce all'innovazione del servizio, all'innovazione del processo di produzione e distribuzione via web e all'innovazione gestionale. Dal punto di vista gestionale, il sistema permette di monitorare l'attività di ogni singolo operatore alberghiero in base alle prenotazioni e ai ricavi previsti: si tratta ancora di nuovi servizi alle imprese basati sull'ICT per sostenere il mercato del turismo, in particolare il settore alberghiero. All'origine del nuovo business vi è l'analisi approfondita del settore turistico e la capacità dell'imprenditore di trasformare la competenza acquisita in un servizio in outsourcing, strutturato e ripetibile.

3.6. Caso Tarasconi Trasporti

Il successo del caso *Tarasconi Trasporti* è determinato dalla realizzazione di due sistemi di tipo informativo: un sistema interno (Voyager) di supporto ai propri processi gestionali nel settore del trasporto intermodale e un sistema esterno (Felixia), ad accesso gratuito, messo a disposizione degli operatori del settore, anche concorrenti.
Analizziamo solo il contributo all'innovazione del sistema Voyager, dato che il secondo è stato realizzato solo a fini promozionali. Tarasconi utilizza il sistema Voyager per ottimizzare la gestione del trasporto merci tenendo conto di due necessità:

- poter pianificare l'ottimizzazione del trasporto merci rispetto ai vari mezzi disponibili, ai vincoli di sicurezza, al traffico e ai consumi;
- poter tracciare e monitorare il percorso della merce dal momento in cui è stata consegnata fino alla consegna a destinazione, qualunque sia il percorso pianificato per il trasporto di un singolo collo.

Il sistema Voyager è considerato uno dei fattori che ha permesso di conquistare la leadership nel settore del trasporto intermodale, distanziando tutti gli altri concorrenti.
Il sistema Voyager ha determinato un elevato livello di ottimizzazione dei costi di gestione del trasporto e contestualmente un alto livello di qualità del servizio fornito alla clientela; dal punto di vista del ruolo svolto dall'ICT si tratta di un contributo all'innovazione di processo e all'innovazione gestionale, in cui l'energia di innovazione dell'imprenditore ha trovato un terreno favorevole.

3.7. Caso Funambol

Il successo del caso *Funambol*, si sviluppa a partire da un'intuizione dell'imprenditore, il quale "vede", con modalità completamente diverse da quelle dei concorrenti, la realizzazione di un'applicazione già presente sul mercato: la sincronizzazione della rubrica e degli impegni personali degli utenti di telefonia mobile.
In questo caso l'innovazione parte dal prodotto e si basa sull'approccio *open source*: l'imprenditore ritiene che il contributo degli *open sourcer* possa essere determinante sia per il miglioramento della qualità del prodotto e il miglioramento del suo livello di aggiornamento, sia per la sua commercializzazione. La strategia commerciale *open source* è basata sul gradimento del prodotto da parte degli interessati che si attivano per promuoverlo loro sponte; naturalmente l'imprenditore prepara e attiva anche un'offerta tradizionale rivolta ai grandi operatori di telefonia mobile ma sempre lasciando al web-market l'iniziativa.
Lo sviluppo dell'approccio vincente diventa possibile solo quando l'imprenditore, dopo un paio di tentativi, decide di migrare in Silicon Valley dove installa la sua struttura di marketing e la sua struttura finanziaria, lasciando a Pavia la struttura di ricerca e di produzione, poiché ritie-

ne che il mancato decollo non dipenda né dai costi di produzione né dalle competenze di ricerca e sviluppo. Trattandosi di un prodotto software è chiaro che il ruolo della tecnologia ICT è fondamentale su tutti i piani, del prodotto, del processo di produzione e distribuzione e della gestione, ma è la diversa combinazione rispetto al territorio che ne decreta il successo.

4. Conclusioni

4.1. Il tipo di prodotto e il tipo di mercato

I tre casi di successo analizzati si basano sulla realizzazione di *servizi innovativi* che vengono sviluppati ed erogati prevalentemente nel segmento *B2B*. Il fatto che siano presenti numerosi casi di servizi immateriali trova una motivazione sia nell'evoluzione del sistema economico, che privilegia l'economia dei servizi rispetto all'economia dei prodotti materiali, sia nell'evoluzione e nella funzione svolta dall'ICT per la realizzazione di questi tipi di offerta: il ruolo dell'ICT è quello di *tecnologia di produzione e distribuzione dei servizi vendibili*.
Con il contributo dell'ICT si producono sia servizi per il settore ICT-operatori di telefonia mobile, come nel caso Funambol, che servizi professionali all'impresa per settori diversi come quello dei servizi di supporto alla gestione delle risorse umane (ODM), quello dei servizi al settore turistico-alberghiero (Hotel Solutions) e quello dei servizi al settore dei trasporti (Tarasconi Trasporti).
Gli altri casi analizzati riguardano invece prodotti materiali, due sono relativi all'introduzione di servizi per gli operatori della distribuzione e uno solo riguarda il classico sistema informatico di supporto al controllo di gestione, dove il successo è determinato, in misura relativa, dall'introduzione di questo sistema, e in maniera preponderante dal lancio di un nuovo prodotto, in questo caso di tipo materiale.
In altre parole sembra che, in via di principio, non ci siano limiti allo sviluppo di servizi innovativi basati su ICT in tutti i settori economici. D'altra parte si possono anche aggiungere nuovi servizi alla gestione di prodotti materiali tradizionali (nel nostro caso, la distribuzione); in base a questa chiave interpretativa, cinque dei sette casi di successo esaminati sono dovuti all'innovazione di servizi realizzati attraverso l'ICT.

4.2. Il livello competitivo del nuovo prodotto/servizio

Tutti i nuovi prodotti e servizi nascono come risposta ad una pressione competitiva che trascura alcuni segmenti di mercato, solitamente quelli relativi alle piccole e medie imprese: ricordiamo che la prevalenza dei servizi innovativi riguarda il B2B.
Generalmente, possiamo dire che l'intuizione imprenditoriale di un nuovo prodotto/servizio trova uno sbocco se l'offerta conseguente si orienta a segmenti di mercato non ancora coperti da servizi/prodotti con analoghe funzioni d'uso, ma la cui prestazione è dimensionata per uno specifico segmento. In altre parole stiamo parlando di iniziative imprenditoriali che possiamo classificare per una *innovazione incrementale* rispetto al prodotto/servizio e per una *innovazione sul mercato* rispetto all'impresa. Tutti i sette casi analizzati si classificano in queste due categorie.

4.3. Ruolo dell'imprenditore e energia di innovazione che esprime

Il ruolo dell'imprenditore è assolutamente determinante, sia per l'innovazione dei servizi che per l'innovazione dei prodotti. Questa affermazione può sembrare ovvia nell'analisi *ex post*, ma deve essere articolata ulteriormente rispetto alla semplificazione adottata nell'analisi di primo livello: abbiamo infatti notato che tutti gli imprenditori dei casi analizzati dimostrano di avere una grande *competenza nella progettazione* (design) del nuovo prodotto/servizio, nel suo lancio sul mercato e nella sua commercializzazione. Nel caso TEkNO, ad esempio, il ruolo determinante dei due imprenditori è accompagnato da una notevole capacità di interazione e realizzazione del sistema di controllo di gestione informatizzato da parte del responsabile dei sistemi informativi.
Infine, quella che abbiamo chiamato *energia di innovazione* si esprime con modalità e intensità diverse in ciascuno dei casi analizzati, ma è comunque determinante per spiegare come siano state superate le difficoltà che il mercato inevitabilmente frappone per arrivare al successo di business.

4.4. Ruolo della tecnologia nel processo di innovazione

In tre casi su sette, quelli caratterizzati dall'innovazione del servizio, il ruolo della tecnologia è di tipo globale: permette di configurare il nuo-

vo servizio (*innovazione di prodotto*), ne permette la produzione e la distribuzione (*innovazione di processo*) e, infine, ne permette la gestione a partire dall'ordine del cliente fino all'erogazione e chiusura del rapporto di collaborazione, includendo anche le fasi amministrative di gestione dell'ordine, della fatturazione e dell'incasso (*innovazione gestionale*). Nel caso ODM ne permette anche l'attuazione della profilatura del cliente; in altre parole, come abbiamo già sottolineato, la tecnologia ICT svolge il ruolo di tecnologia di produzione dei servizi vendibili. Nei due casi relativi all'attività di distribuzione la tecnologia assume essenzialmente un ruolo nel'innovazione di processo e nell'innovazione gestionale. Infine, nei casi TEkNO e Tarasconi assume solo un ruolo nell'innovazione gestionale.
L'ICT, in tutte le realtà esaminate, integra la completa *tracciabilità* della catena del valore per singolo cliente e/o per singola commessa; nel caso delle reti distributive la tracciabilità della singola transazione coinvolge tutti gli operatori implicati, lasciando a ciascuno un'indicazione del plusvalore ottenuto lungo la catena del valore che "dirige" il prodotto dal fornitore al cliente finale.

4.5. Altri fattori critici di successo emersi dall'analisi

Esistono altri fattori critici che permettono di spiegare il successo dei casi analizzati, anche se sono di difficile interpretazione in termini generali. Ci sembra comunque utile sottolineare un fattore che caratterizza il primo e l'ultimo caso di successo considerato: l'approccio di tipo *open source*.
Nel caso ODM l'approccio *open source* si riscontra nella fase di acquisizione dei dati sulle retribuzioni: il criterio è quello di mettere a fattor comune i dati acquisiti fornendo un piccolo servizio personalizzato (quanto guadagno): tutti coloro che accedono sono contemporaneamente fornitori di informazioni e utilizzatori di un servizio basato sulla loro stessa informazione e su quelle di tutti coloro che li hanno preceduti.
Nel caso Funambol, l'adozione dell'approccio *open source* è parte integrante dell'offerta commerciale e di aggiornamento tecnico: la realizzazione dei miglioramenti del prodotto è affidata a risorse *open source* e tale approccio è utilizzato anche sul piano commerciale; infatti Funambol reagisce alle richieste degli operatori di telefonia mobile con offerte molto mirate, senza però rispondere con un'azione proattiva, tentata e risultata troppo onerosa e incerta in una prima fase di impostazione dell'attività commerciale.

Bibliografia

[1] Bertelè Umberto, Chiesa Vittorio, Noci Giuliano; prefazione di Esposti Massimo: *Creare valore con la rete: innovazioni gestionali e nuove opportunità di business nel post new economy*. Milano - Il Sole 24 Ore, 2002.

[2] Christensen Clayton M.: *Il dilemma dell'innovatore*. FrancoAngeli, 2001, Milano.

[3] Baumol William: *The Free-Market Innovation Machine*. Princeton University Press, 2004.

[4] Utterback James M., con prefazione di Butera F.: *Padroneggiare le dinamiche dell'innovazione industriale*. Franco Angeli, 2003.

[5] Normann Richard: *La gestione strategica dei servizi*. Etas 1999.

Roberto Bellini è docente di Marketing e Gestione della Relazione con il cliente nell'ambito del MIP - Politecnico di Milano, con una focalizzazione sulla innovazione nelle reti di imprese. Presiede la Sezione AICA di Milano ed è responsabile per Aica del Cantiere dei Mestieri ICT.
E-mail: roberto.bellini@polimi.it

Chiara Francalanci è professore associato di Sistemi Informativi al Politecnico di Milano. Ha scritto numerosi articoli sulla progettazione e sul valore economico delle tecnologie informatiche, svolto attività di ricerca e consulenza nel settore finanziario e manifatturiero sia in Italia sia presso la Harvard Business School ed è editor del Journal of Information Technology.
E-mail: francala@elet.polimi.it

ICT E DIRITTO

Rubrica a cura di

Antonio Piva, David D'Agostini

Scopo di questa rubrica è di illustrare al lettore, in brevi articoli, le tematiche giuridiche più significative del settore ICT: dalla tutela del *domain name* al *copyright* nella rete, dalle licenze software alla *privacy* nell'era digitale. Ogni numero tratterà un argomento, inquadrandolo nel contesto normativo e focalizzandone gli aspetti di informatica giuridica.

Diritto d'autore tra Digital Right Management e Creative Commons

David D'Agostini, Antonio Piva

1. Introduzione

La globalizzazione si è ormai estesa a ogni mezzo di comunicazione: la parola, lo scritto, il suono, l'immagine, tutto può essere trasformato in bit, elaborato e, infine, diffuso tramite la rete internet nella quale le informazioni giungono a destinazione in tempo reale e l'utente ha la possibilità di accrescere continuamente la sua capacità conoscitiva.
I grossi mutamenti portati dal continuo progresso delle tecnologie digitali hanno portato a una crisi della tradizionale concezione del *copyright*, ossia del diritto dell'autore di un'opera di sfruttare economicamente la medesima: con l'avvento delle tecnologie digitali, la copia di un file multimediale (audio o video) risulta estremamente semplice e poco costoso e inoltre, a differenza dei supporti analogici, non comporta diminuzioni qualitative.
Il più efficace mezzo che consente la violazione del diritto d'autore è il sistema del *file-sharing*: il primo software che nell'autunno del 1999 ha permesso la condivisione fra utenti di opere musicali attraverso Internet è stato Napster, un *peer-to-peer* (P2P) che utilizzava un sistema di server centrali per mantenere la lista dei computer connessi e dei file condivisi, mentre le transazioni vere e proprie avvenivano direttamente tra i vari utenti. Negli anni successivi la società che aveva sviluppato Napster incorse in una vicenda giudiziaria che ne ha decretato il fallimento, ma nuovi programmi P2P continuano ad essere utilizzati da milioni di persone.
L'atteggiamento più diffuso fra gli utenti della rete è che lo scambio delle informazioni rappresenti una forma di libertà insostituibile in un mondo troppo controllato da lobbies economiche e di potere: come in una rincorsa per la sopravvivenza, in risposta alle minacce delle major discografiche e cinematografiche di azioni legali nei confronti degli utenti, vengono sperimentate nuove tecnologie che implementano sistemi di crittografia per garantire l'anonimato.
Il Parlamento Europeo è tornato più volte in questi anni a occuparsi della proprietà intellettuale fino alla Direttiva 2004/48/CE (denominata *IP Right Enforcement*), ma anche i legislatori di molti Paesi stanno studiando come contenere il fenomeno della pirateria audiovisiva nella Rete.
In Italia è stata approvata la legge 21 maggio 2004, n. 128 (che converte il cosiddetto Decreto Urbani) con cui il legislatore si è posto l'obbiettivo di contrastare il fenomeno dell'illecito scambio via internet di opere tutelate dal diritto d'autore adottando una linea repressiva.

2. Digital Rights Management

Se da un lato la diffusione di strumenti digitali ha permesso la distribuzione illecita di contenuti multimediali, dall'altro la ricerca scientifica si affianca al diritto nel proteggere i diritti d'autore grazie all'invenzione di misure di "autotutela tecnologica": in ciò consistono i sistemi

DRM (*Digital Rights Management,* che letteralmente significa è "gestione dei diritti digitali"). Il DRM, concepito per essere applicato alle opere digitali, non tutela solo il prodotto intellettuale, ma estende la sua protezione anche agli strumenti informatici posti a difesa della creazione e, infatti, viene definito nei seguenti termini:
"*Il DRM ... si compone di diversi aspetti principali, necessari per gestire l'intermediazione distributiva qualora vi siano soggetti terzi tra il titolare dei diritti e l'utente finale*:
1. *Identificazione e descrizione dei diritti di proprietà intellettuale nella catena del valore del contenuto, dalla produzione alla fruizione*;
2. *Tracciamento delle licenze d'uso e dell'utilizzo effettivo del contenuto*;
3. Misure tecniche che assicurano le restrizioni di uso"[1].
L'attuale quadro normativo in tema di "nuovo diritto d'autore" si fonda sulla "C*opyright Directive*"[2], attuata nel nostro ordinamento con d.lgs. 68/03[3].
Gli strumenti *tecnologico-legali* utilizzati contro attività illecite possono distinguersi in misure tecnologiche di protezione e informazioni sul regime dei diritti.
Le misure tecnologiche di protezione (abbreviate in MTP), già previste dagli accordi internazionali e dalla legislazione sul copyright negli Stati Uniti[4], permettono all'autore di predeterminare le modalità stesse della fruizione dell'opera da parte dell'utente.
La direttiva 2001/29/CE all'art. 6 comma 3 definisce le MTP come "*tutte le tecnologie, i dispositivi o componenti che, nel normale corso del loro funzionamento, sono destinati a impedire o limitare atti, su opere o altri materiali protetti, non autorizzati dal titolare del diritto d'autore o del diritto connesso al diritto d'autore, così come previsto dalla legge o dal diritto sui generis previsto al capitolo III della direttiva 96/9/CE. Le misure tecnologiche sono considerate "efficaci" nel caso in cui l'uso dell'opera o di altro materiale protetto sia controllato dai titolari tramite l'applicazione di un controllo di accesso o di un procedimento di protezione, quale la cifratura, la distorsione o qualsiasi altra trasformazione dell'opera o di altro materiale protetto, o di un meccanismo di controllo delle copie, che realizza l'obiettivo di protezione*".
In Italia l'adeguamento dell'ordinamento interno alla disciplina stabilita in sede europea è stato realizzato dal già menzionato d.lgs. 68/03 il quale, per quanto riguarda le MTP, ha introdotto nella legge su diritto d'autore l'art. 102 *quater* che in buona sostanza ricalca la medesima definizione di cui alla direttiva comunitaria.
A seconda dell'oggetto del controllo è possibile distinguere due categorie di misure tecnologiche di protezione:
1. *Controllo sull'accesso* alle informazioni, detto "accesso condizionato" ai servizi della società dell'informazione[5]: per esempio, la criptazione delle trasmissioni della televisione a pagamento, *pay-per-view*.
2. *Controllo sulla modalità di utilizzo* delle informazioni: per esempio, i dispositivi contro la duplicazione dei DVD, oppure gli accorgimenti tecnici che impediscono la stampa di un documento ovvero la copia, o la manipolazione.
Un interessante esempio di applicazione delle MTP può ritrovarsi nel caso di Dmitry Sklyarov, programmatore russo arrestato nel 2001 dall'FBI con l'accusa di "*Circumvention of Technological*

[1] Definizione tratta dal documento "*Digital Rights Management*. Relazione introduttiva" redatto alla fine del 2004 dalla "Commissione interministeriale sui contenuti digitali nell'era di Internet", c.d. "Commissione Vigevano", nominata con decreto ministeriale del 23 luglio 2003, diffusa *on line* a partire dal gennaio 2005 dal Ministero per l'innovazione e le tecnologie.

[2] Direttiva 2001/29/CE del Parlamento Europeo e del Consiglio del 22/5/2001 "sull'armonizzazione di taluni aspetti del diritto d'autore e dei diritti connessi nella società dell'informazione" in G.U.C.E. L. 167 del 22 giugno 2001, p. 10.

[3] Decreto Legislativo 9 aprile 2003, n. 68 "Attuazione della direttiva 2001/29/CE sull'armonizzazione di taluni aspetti del diritto d'autore e dei diritti connessi nella società dell'informazione" in Suppl. Ord. alla *Gazzetta Ufficiale* n. 61 del 14 aprile 2003, in vigore dal 29 aprile 2003, su delega emessa con Legge 1 marzo 2002, n.39 "Disposizioni per l'adempimento di obblighi derivanti dall'appartenenza dell'Italia alle Comunità europee - Legge comunitaria 2001" in Suppl. Ord. alla *Gazzetta Ufficiale* n. 54 del 26 marzo 2002.

[4] Le MTP sono state introdotte inizialmente nell'ordinamento statunitense, con il *Digital Millennium Copyright Act*, (DMCA) del 28 Ottobre 1998, Pubbl. L. n. 103 – 304, in vigore dal 28 ottobre 2000.

[5] Legge 7 febbraio 2003, n.22 "Modifica al decreto legislativo 15 novembre 2000 n. 373, in tema di tutela del diritto d'autore", in *Gazzetta Ufficiale* del 15 febbraio 2003, n. 38 che estende la tutela penale di cui agli artt. 171 bis e octies della L.633/1941 alle ipotesi di "distribuzione illecita" di smart cards "piratate".

Protection Measures"[6] per aver violato le misure tecniche di protezione del formato PDF. Il problema giuridico di questo caso consiste nella qualificazione della distribuzione di strumenti che sono idonei ad un utilizzo lecito, ma che possono anche essere funzionali a scopi illeciti. Il programma incriminato, permettendo per esempio di duplicare *files* protetti, non necessariamente consente all'utente di violare il *copyright*: a determinate condizioni è concesso riprodurre una copia privata "di sicurezza" di un *file* legittimamente acquistato. Infatti il titolare della società moscovita per cui lavorava Sklyarov testimoniò che la sua azienda distribuiva il programma alla stessa Adobe ed al Governo americano, i quali non potevano certamente considerarsi "criminali". Invece secondo la pubblica accusa, sostenuta dal procuratore Scott Frewing, il programmatore russo e la sua società "non potevano non sapere" che attraverso il software distribuito era possibile compiere condotte punite come reati dalla legge americana. Il 17 dicembre 2002 la giuria emise verdetto di assoluzione: il programma è distribuito fuori dalla giurisdizione statunitense, quindi con modalità perfettamente legittime.

3. Le misure tecnologiche di protezione

In Italia il rispetto delle MTP costituisce un'ulteriore condizione di liceità dell'utilizzo del materiale alla cui protezione esse sono apposte. Il diritto esclusivo dell'autore di riprodurre l'opera (art.61) è soggetto a eccezione in favore dell'utente in presenza di cinque condizioni[7]:

1. copia eseguita da persona fisica;
2. per uso esclusivamente personale;
3. senza scopo di lucro e senza fini direttamente o indirettamente commerciali;
4. nel rispetto delle misure tecnologiche;
5. con un "equo compenso" al titolare del diritto d'autore.

Tuttavia la riproduzione non è consentita nemmeno in tali casi qualora l'opera sia "messa a disposizione al pubblico in modo che ciascuno possa avervi accesso dal luogo e nel momento scelti individualmente, ovvero quando l'accesso è consentito sulla base di accordi contrattuali".
L'apposizione delle MTP ha due limiti, corrispondenti a esigenze eterogenee: da un lato l'interesse pubblico impone all'autore la rimozione delle "misure" su richiesta dell'autorità competente, qualora si renda necessario l'accesso all'opera per fini di sicurezza pubblica o per assicurare il corretto svolgimento di un procedimento amministrativo, parlamentare o giudiziario; dall'altro l'interesse dell'utente ammette la duplicazione purché si avverino le seguenti condizioni:

1. se compiuta da parte di persona fisica;
2. in presenza di possesso legittimo o accesso legittimo;
3. per estrarre copia privata, anche solo analogica;
4. per uso personale;
5. qualora tale facoltà non sia in contrasto con lo sfruttamento normale dell'opera;
6. qualora la duplicazione non arrechi ingiustificato pregiudizio ai titolari dei diritti.

Per converso, si reprime l'utilizzo dei dispositivi di elusione delle misure tecnologiche di protezione che costituisce illecito amministrativo punito dall'art. 174 *ter* con: la sanzione amministrativa di 154 €; la sanzione accessoria della confisca del materiale; l'eventuale sanzione accessoria di pubblicazione del provvedimento sulla stampa. La produzione o commercializzazione di dispositivi elusivi, invece, è reato ai sensi dell'art. 171 *ter* lettera *f bis*), e comporta anche la sospensione dell'attività commerciale ovvero, in caso di recidiva, la revoca della licenza commerciale. Se Sklyarov fosse arrestato in Italia sarebbe imputabile per il reato di cui all'*art. 171 ter, comma 1, lett. g*), che punisce chi fabbrica, importa, distribuisce, vende, noleggia, cede a qualsiasi titolo, pubblicizza per la vendita o il noleggio, o detiene per scopi commerciali, attrezzature, prodotti o componenti ovvero presta servizi che abbiano la prevalente finalità o l'uso commerciale di eludere efficaci misure tecnologiche di cui all'art. 102 quater ovvero siano principalmente progettati, prodotti, adattati o realizzati con la finalità di rendere possibile o facilitare l'elusione di predette misure. Ciò implica la pena della reclusione da sei mesi a tre anni, e la multa da € 2.582 a € 15.493, come per il reato di furto, art.624 C.P.

[6] Introdotto dal DMCA alla sezione 1201 del capitolo 12, titolo 17, U.S. Code. Il reato punisce lo "sviamento" delle misure tecnologiche predisposte per prevenire l'accesso o la copia di opere protette dal diritto d'autore, e l'attività contestuale di distribuzione dei dispositivi aventi tale utilizzo.

[7] Art. 71 sexies comma 1, L. 633/1941.

4. CMI: INFORMAZIONI SUL REGIME DEI DIRITTI

Ogni documento informatico contiene informazioni relative al suo autore, alla data di creazione, all'ultima modifica. La tutela della genuinità di tali dati appare evidente, poiché l'opera può essere trasmessa telematicamente anche a grande distanza dal luogo di creazione.
Possono essere inseriti ulteriori dati relativi ad esempio ai diritti connessi alla distribuzione, alla registrazione dell'opera stessa, alle facoltà concesse o meno dal produttore al fruitore. L'attribuzione delle informazioni sull'autore è detta *watermarking*, e deriva da *watermark*, particolare segno impresso sulla carta a garanzia di autenticità visibile solamente controluce. Le CMI, *Copyright Management Informations*, prima previste a livello internazionale, sono state adottate prima dagli Stati Uniti[8] e poi dall'Unione Europea; la direttiva 2001/29/CE all'art. 7 comma 2 definisce le informazioni sul regime dei diritti "*qualunque informazione fornita dai titolari dei diritti che identifichi l'opera o i materiali protetti di cui alla presente direttiva o coperti dal diritto sui generis di cui al capitolo III della direttiva 96/9/CE, l'autore o qualsiasi altro titolare dei diritti, o qualunque informazione circa i termini e le condizioni di uso dell'opera o di altri materiali nonché qualunque numero o codice che rappresenti tali informazioni". In sede UE sono proibite le seguenti condotte dall'art.7 comma 1*:
a. rimuovere o alterare qualsiasi informazione elettronica sul regime dei diritti;
b. distribuire, importare a fini di distribuzione, diffondere per radio o televisione, comunicare o mettere a disposizione del pubblico opere o altri materiali protetti ai sensi della presente direttiva o del capitolo III della direttiva 96/9/CE, dalle quali siano state rimosse o alterate senza averne diritto le informazioni elettroniche sul regime dei diritti".

Del medesimo tenore è la definizione adottata nel nostro ordinamento dal d.lgs. 68/03 secondo il quale le CMI sono "informazioni elettroniche sul regime dei diritti" che identificano l'opera o il materiale protetto, nonché l'autore o qualsiasi altro titolare dei diritti. Tali informazioni possono altresì contenere indicazioni circa i termini o le condizioni d'uso dell'opera o dei materiali, nonché qualunque numero o codice che rappresenti le informazioni stesse o altri elementi di identificazione"[9].

5. CREATIVE COMMONS

In questo scenario c'è anche chi propone modelli alternativi e indica nuovi modi per concedere in uso le opere dell'ingegno, come il giurista Lawrence Lessig[10] che nel 1999 impugnò dinnanzi alla Suprema Corte degli Stati l'ennesima estensione alla durata dei diritti d'autore approvata dal Congresso sostenendo che la Costituzione tutelava le arti e le scienze per un periodo limitato di tempo, mentre il Congresso aveva esteso il termine di durata del copyright per ben undici volte.
Il ricorso fu respinto ma Lessig nel 2001 fondò la *Creative Commons* (CC), organizzazione non governativa senza scopo avente come scopo sociale l'espansione della portata delle opere di creatività disponibili per la condivisione.
Le licenze *Creative Commons* (attualmente alla versione 2.5) sono le seguenti:
❑ **Attribuzione** - *Attribution* - (by): permette di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare l'opera a patto che venga attribuita la paternità dell'opera nei modi indicati dall'autore o da chi ha dato l'opera in licenza; si riferisce all'obbligo di rendere merito all'Autore originario dell'opera e quindi citarlo in ogni utilizzazione dell'opera stessa (Figura 1 **A**).
❑ **Attribuzione - Non opere derivate** - *No Derivative Works* - (nd): permette di riprodurre, distri-

[8] Il DMCA definisce le CMI come: "*identifying information about the work, the author, the copyright owner, and in certain cases, the performer, writer or director of the work, as well as the terms and conditions for use of the work, and such other information as the Register of Copyright may prescribe by regulation*". DMCA, section 1201 (c). Costituiscono reato due particolari condotte, relativamente alle informazioni sul *copyright*: Sezione 1201 (a): l'attribuzione di false informazioni, o la loro distribuzione "*if done with the intent to induce, enable, facilitate or conceal infringement*"; Sezione 1201 (b): la rimozione o alterazione dei segni senza autorizzazione, o la loro distribuzione "*with reasonable ground to know that it will induce, enable, facilitate or conceal the infringement*".

[9] La norma trova inserimento nella L. 633/1941 all'art. 102 *quinquies* comma 1.

[10] Lessig attualmente è professore ordinario della facoltà di Giurisprudenza di Stanford (in precedenza insegnava ad Harvard) e riconosciuto come uno dei massimi esperti di diritto d'autore.

buire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare l'opera; non consente di alterare o trasformare l'opera, né di usarla per crearne un'altra (Figura 1 **B**).

❑ **Attribuzione - Non commerciale** - *Noncommercial* - (nc): permette di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare l'opera e i lavori derivati da questa solo per scopi di natura non commerciale (Figura 1 **C**).

❑ **Attribuzione - Condividi allo stesso modo** - *Share Alike* - (sa): anche se letteralmente significa "condividi allo stesso modo", permette che altri distribuiscano lavori derivati dall'opera solo con una licenza identica a quella concessa con l'opera originale (Figura 1 **D**).

❑ **Attribuzione - Non commerciale - Non opere derivate**: l'opera non può essere usata per fini commerciali e non può essere alterata o trasformata o usata per crearne un'altra.

❑ **Attribuzione - Non commerciale - Condividi allo stesso modo**: l'opera non può essere usata per fini commerciali e i lavori derivati devono essere distribuiti con una licenza identica a quella concessa con l'opera originale.

La scelta può essere fatta in maniera semplice nel sito della Creative Commons[11] selezionando le caselle corrispondenti ai diritti che l'autore intende riservarsi e indicando la giurisdizione alla quale sottoporre la licenza e il tipo di opera. Il sistema consente al computer di identificare e comprendere direttamente i termini della licenza scelta dall'autore, rendendo più semplice la ricerca e la condivisione delle opere.

Il sito ufficiale spiega quali accorgimenti scegliere per utilizzare la licenza; nel caso di un'opera diffusa via Internet viene suggerito di inserire nel sito il logo Creative Commons con la dicitura "some rights reserved" (Figura 2; in italiano "alcuni diritti riservati" che richiama la tradizionale dicitura "all rights reserved") con il link che rimanda alla licenza prescelta[12].

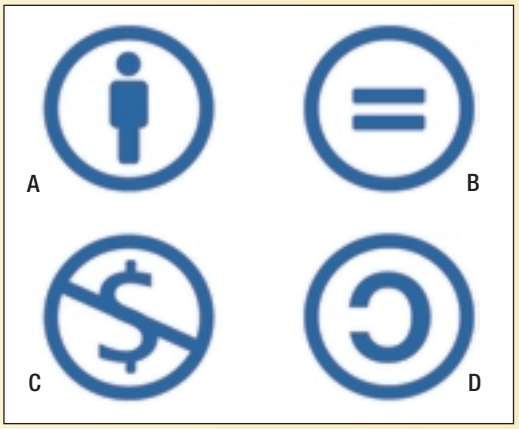


FIGURA 1
Le licenze Creative Commons

FIGURA 2
Logo di Creative Commons per le opere diffuse via Internet

Concludendo il progetto Creative Commons non intende effettuare una lotta indiscriminata con il tradizionale copyright ma di adattarlo alle nuove comunicazioni digitali e multimediali incentivando le opere letterarie, musicali e cinematografiche, concesse in condivisione anche attraverso lo sviluppo di un ampio catalogo di opere su diversi media, promuovendo l'etica basata sulla condivisione stessa.

Antonio Piva laureato in Scienze dell'Informazione, Vice Presidente dell' ALSI (Associazione Nazionale Laureati in Scienze dell'Informazione ed Informatica) e Presidente della commissione di informatica giuridica.
Docente a contratto di diritto dell'informatica all'Università di Udine.
Consulente sistemi informatici e Governo Elettronico nella PA locale, valutatore di sistemi di qualità ISO9000 ed ispettore AICA ECDL.
E-mail: antonio@piva.mobi

David D'Agostini avvocato, ha conseguito il master in informatica giuridica e diritto delle nuove tecnologie, fornisce consulenza e assistenza giudiziale e stragiudiziale in materia di *software, privacy* e sicurezza, contratti informatici, *e-commerce*, nomi a dominio, 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

[11] http://creativecommons.org o nel sito italiano www.creativecommons.it.

[12] Il link richiama la versione sintetica della licenza (facilmente riconoscibile e comprensibile dal grande pubblico degli utenti) che rimanderà alla versione "Legal Code" (che utilizza il linguaggio giuridico); infine vengono fornite le righe di codice per attivare la versione "Digital Code" che permette ai motori di ricerca di identificare l'opera in base alle condizioni di licenza e di conseguente utilizzo.

DENTRO LA SCATOLA

Rubrica a cura di

Fabio A. Schreiber

Dopo aver affrontato negli scorsi anni due argomenti fondanti dell'Informatica – il modo di codificare l'informazione digitale e la concreta possibilità di risolvere problemi mediante gli elaboratori elettronici – con questa terza serie andiamo ad esplorare "*Come parlano i calcolatori*". La teoria dei linguaggi e la creazione di linguaggi di programmazione hanno accompagnato di pari passo l'evolversi delle architetture di calcolo e di gestione dei dati, permettendo lo sviluppo di applicazioni sempre più complesse, svincolando il programmatore dall'architettura dei sistemi e consentendogli quindi di concentrarsi sull'essenza del problema da risolvere.
Lo sviluppo dell'Informatica distribuita ha comportato la nascita, accanto ai linguaggi per l'interazione tra programmatore e calcolatore, anche di linguaggi per far parlare i calcolatori tra di loro – i protocolli di comunicazione. Inoltre, la necessità di garantire la sicurezza e la privatezza delle comunicazioni, ha spinto allo sviluppo di tecniche per "non farsi capire" da terzi, di quì l'applicazione diffusa della crittografia.
Di questo e di altro parleranno le monografie quest'anno, come sempre affidate alla penna (dovrei dire tastiera!) di autori che uniscono una grande autorevolezza scientifica e professionale ad una notevole capacità divulgativa.

Linguaggi per la progettazione dell'hardware

William Fornaciari, Mariagiovanna Sami

1. Introduzione

La crescita della complessità dei sistemi digitali negli ultimi 30 anni è stata possibile grazie alla messa a punto di metodologie di progetto in grado di automatizzare molte fasi realizzative. Un ruolo fondamentale è ricoperto dai linguaggi di descrizione dell'hardware, di cui il presente articolo traccia lo stato presente e le possibili evoluzioni, in relazione anche ai mutati scenari tecnologici. Ormai non esiste più una netta distinzione fra progettazione Hw e Sw: siamo nell'era della progettazione concorrente a livello sistema.

2. Realizzazione di un sistema digitale

La crescita esponenziale della "densità" dei circuiti digitali, rappresentata dal numero dei transistori su un *chip* (ben prefigurata dalla cosiddetta "legge di Moore") ha portato alla necessità di definire fasi e modelli del processo di progettazione e realizzazione che consentissero di dominarne la complessità (oggi su un unico chip si possono integrare dalle decine di milioni ai miliardi di transistor, in architetture che possono comprendere diversi microprocessori oltre a unità di elaborazione dedicate e a memorie, con geometrie del singolo transistor dell'ordine dei 45 nm). Tale strutturazione, esemplificata ad alto livello nella figura 1, ha diversi vantaggi. Innanzitutto consente di *specializzare* le competenze dei progettisti su alcuni passi soltanto dell'intero processo; grazie ad una *formalizzazione* dei modelli consente poi di usare rappresentazioni utili alla *automatizzazione* del processo di ottimizzazione; infine, ma non meno importante, consente di *simulare* il sistema a livelli di astrazione e di accuratezza variabili, in modo da garantire che il comportamento della realizzazione fisica finale sia effettivamente coerente con il progetto originale. A partire da una specifica del sistema, che verrà espressa con un opportuno linguaggio di descrizione o modello rappresentativo, si seguiranno trasformazioni che faranno evolvere la descrizione rendendola sempre meno astratta e sempre più vicina ad una formulazione finale sufficientemente dettagliata da potere guidare il processo

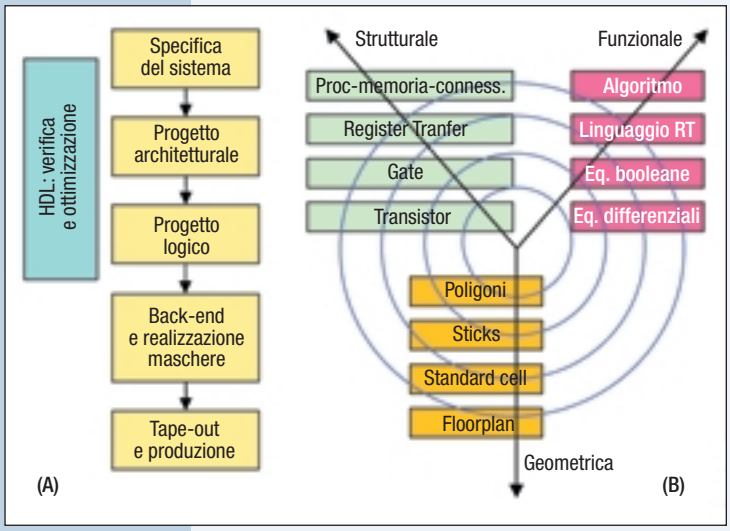


FIGURA 1

Passi del processo di progettazione di un sistema digitale (A), domini e livelli della modellazione e ottimizzazione (B). "Il cosiddetto "diagramma a Y" è stato introdtto da Gajski e Kuhn all'inizio degli anni '80"

di fabbricazione del silicio. Il percorso verso i livelli di astrazione più bassi non è una semplice traduzione, ma un processo di *design* complesso che prevede molte ottimizzazioni, (costo, velocità, potenza dissipata e tempo di progetto per citarne alcune) e, ad ogni passaggio, coinvolge attività di simulazione e verifica per garantire che il circuito continui a rispettare obiettivi e vincoli di progetto iniziali.

Per un sistema digitale, con buona approssimazione possiamo affermare che il processo di sintesi della macrofase detta *front-end* si arresta quando vengono identificati gli elementi digitali che compongono il sistema (esempio, porte logiche, memorie, registri) e le loro connessioni. A partire da questa descrizione inizia l'attività di *back-end*, che ottimizzerà ulteriormente il sistema cercando di disporre tali elementi in modo da usare la minore area di silicio possibile (floorplan) e arriverà a definire le caratteristiche geometriche e di drogaggio delle varie zone di silicio, informazioni che porteranno alla creazione di opportune "maschere" che saranno usate per la fase di produzione vera e propria dell'integrato. Oggi, una linea di produzione per un comune processo tecnologico digitale richiede in genere non meno di 4-5 settimane per la fase di *back-end* e di realizzazione di un insieme di prototipi su silicio, con un costo per la realizzazione delle maschere difficilmente inferiore al milione di euro. È quindi ovvio che l'identificazione e la messa a punto di eventuali errori dopo la fase di produzione in molti casi può significare il fallimento di un progetto; di conseguenza, molto sforzo è stato indirizzato verso lo sviluppo di strumenti di progettazione automatica (*Electronic Design Automation*, EDA), per consentire al progettista non solo di automatizzare molte fasi di sintesi ma anche (se non soprattutto) di effettuare una verifica approfondita e di lavorare il più possibile in modo *top-down*, pur mantenendo la possibilità di confrontare in tempi ragionevoli varie alternative di progetto.

Un progetto, come visto nella figura 1, può quindi essere visto a diversi livelli di astrazione, sia sotto il profilo *funzionale*, sia per quanto concerne la *struttura* e l'organizzazione dei blocchi che ne andranno a comporre l'architettura, sia per quanto attiene più da vicino le informazioni *geometriche* usate nel processo di produzione del silicio.

I linguaggi di descrizione dell'hardware (HDL), come il Verilog e il VHDL, hanno un ruolo fondamentale, perché hanno favorito la nascita di strumenti di *progetto automatico*, ridotto drasticamente i *tempi di progetto*, consentito la creazione di *figure professionali* con competenze sempre *meno verticali* e permesso la collaborazione di gruppi di progettisti a volte geograficamente distanti (passaggio pressoché inevitabile visti i tassi di crescita della complessità dei sistemi e delle loro tecnologie di realizzazione).

3. Linguaggi per la modellazione hardware: il VHDL

Per esemplificare l'organizzazione e l'utilizzo di un linguaggio per la descrizione dell'hardware, faremo riferimento al VHDL, che è il più diffuso in Europa. Il VHDL nasce negli USA nell'ambito del progetto VHSIC (*Very High Speed Integrated Circuits*) sponsorizzato dal Dipartimento delle Difesa (DoD) americano. La prima versione risale al 1984, la prima standardizzazione IEEE avviene nel 1987 e i successivi aggiornamenti sono stati compiuti negli anni 1992, 1997 e 2002. Gli obiettivi iniziali del progetto dovevano consentire al DoD di standardizzare le procedure di progettazione e documentazione degli apparati digitali e di svincolarsi da eventuali dipendenze da un particolare fornitore di sistemi elettronici. La struttura e l'utilizzo del linguaggio hanno forti similarità con un altro ben noto standard, il lin-

guaggio ADA, e si accompagnano ad una versatilità che gli consente di essere utilizzato sia per fare progettazione e verifica funzionale, sia come riferimento per l'intero processo di sintesi che porta alla realizzazione del sistema finale.
Nonostante l'apparente similarità con un normale linguaggio di programmazione, il VHDL ha particolarità che derivano dall'essere concepito per rappresentare sistemi hardware, dove non esiste un unico esecutore sequenziale delle operazioni e dove la sincronizzazione degli accessi alle varie unità non può essere implicitamente considerata come un dato di fatto. Le caratteristiche salienti di un HDL dotato di buona potenza rappresentativa, quale il VHDL, sono le seguenti:

- *Concorrenza*. Deve essere possibile rappresentare senza difficoltà l'intrinseco parallelismo dei sistemi hardware, così come, se necessario, forzare un ordinamento nell'esecuzione di gruppi di operazioni.
- *Astrazione*. Si vuole lavorare con diversi livelli di astrazione sia nella fase di specifica sia durante la simulazione.
- *Viste*. Si vuole operare con viste differenti dello stesso sistema, che possono per esempio avere le medesime interfacce ma avere poi rappresentazioni comportamentali piuttosto che strutturali.
- *Strutturazione*. Deve essere possibile rappresentare, per i diversi livelli di astrazione a cui si considera il sistema, i collegamenti fra i vari blocchi componenti. Si possono pertanto identificare gerarchie del sistema digitale; diviene possibile la progettazione modulare di sistemi complessi, grazie al supporto al riuso di unità di progetto o di librerie sviluppate anche da terze parti.
- *Tempo*. È necessario specificare precisi parametri temporali per rappresentare le caratteristiche dei sistemi reali che si otterranno dopo la sintesi, così come la sincronizzazione fra diversi sottosistemi, inclusa la gestione dei *clock*.

Nel caso del VHDL, c'è una separazione netta fra la specifica delle interfacce di un componente e il suo "corpo". Questa caratteristica, ereditata dal linguaggio ADA, consente di associare *ad una sola specifica* sia differenti viste, sia anche progetti alternativi da utilizzarsi in base alla fase del progetto o a specifiche esigenze applicative.
In VHDL ogni entità da modellare si chiama *design entity*, e si compone di una *entity declaration* e di una o più *architecture*. Per esemplificare l'uso del VHDL consideriamo un semplice sommatore completo (*full adder* nella Figura 2)

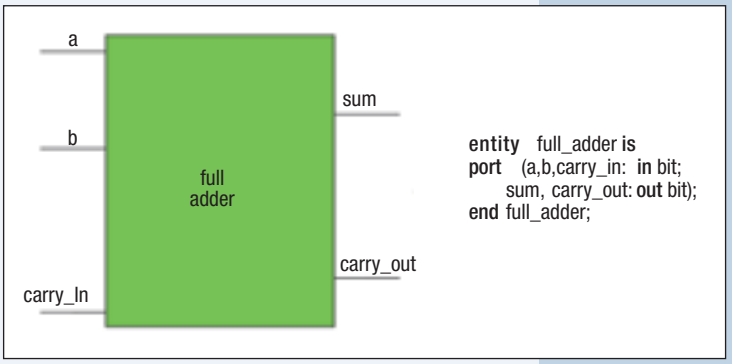


FIGURA 2
Descrizione in VHDL delle interfacce di un sommatore completo (entity declaration), dove si identificano le "porte" di ingresso e uscita

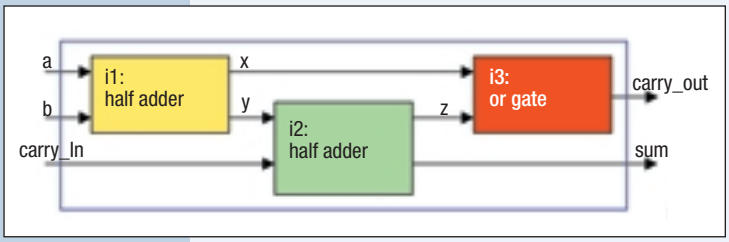
Riquadro 1

```
architecture behavior of full_adder is
begin
sum <= (a xor b) xor carry_in after 10 ns;
carry_out <= (a and b) or (a and carry_in) or
             (b and carry_in) after 10 ns;
end behavior;
```

che riceve in ingresso due bit addendi (a e b) e un eventuale riporto (carry_in) e produce il bit di somma (sum) e l'eventuale riporto in uscita (carry_out). La figura 2 mette in evidenza le interfacce del full adder che si traducono nella *entity declaration* riportata a destra.
L'architettura di una entity si compone di un *header* e di un *body* che può essere di tipo strutturale o comportamentale (behavioral). Una descrizione comportamentale fornisce informazioni sufficienti per il calcolo dei valori dei segnali di uscita a partire da quelli in ingresso oltre a eventuali informazioni di temporizzazione del circuito. Indicando con <= l'assegnamento di segnali, una possibile descrizione comportamentale del *full adder* è riportata nel riquadro 1.
Le informazioni di temporizzazione come quelle specificate con la parola chiave *after* sono utili in fase di simulazione per generare forme d'onda in risposta alla variazione degli stimoli di ingresso, tenendo conto della presenza nel circuito di ritardi reali e finiti per il calcolo delle uscite. Nel caso del riquadro 1, le uscite *sum* e *carry_out* saranno stabilizzate sul loro valore finale dopo 10 ns rispetto alla variazione degli ingressi.
Una rappresentazione del *full adder* di tipo *strutturale*, invece, focalizzerebbe l'attenzione sull'architettura di una realizzazione dello stesso circuito e non sulla sua funzionalità: si veda-

no la figura 3 e il riquadro 2, dove il *full adder* è realizzato connettendo due semisommatori (*half adder* i1 e i2) e una porta AND (i3).
I componenti debbono essere dichiarati all'interno della *component declaration*, che fornisce le informazioni salienti sul componente anche se la sua descrizione completa non è ancora presente nel database del progetto, come tipicamente avviene quando si adotta un approccio top-down alla progettazione. La corrispondenza fra le porte della entity e quelle dei componenti interni è descritta tramite il cosiddetto *port map*.
Così come col costrutto *after* si modella il ritardo nella generazione delle uscite, il VHDL consente di considerare i *livelli* (che sono un'astrazione dei valori di tensione) e la *forza* dei segnali (che invece modellano l'impedenza delle sorgenti elettriche) oltre all'eventuale indeterminatezza nei valori assunti da un segnale. In VHDL ogni uscita è associata ad un *driver*: qualora più uscite siano connesse fra loro, una funzione di *risoluzione* calcola il valore considerando i vari contributi con il loro valore e la loro forza[1]. Non meno importante, VHDL offre costrutti analoghi a quelli dei linguaggi di programmazione, come la selezione, *select/when*, la tipizzazione, i sottoprogrammi ecc. oltre a una raffinata gestione delle librerie di componenti.
La semantica del VHDL ha una formalizzazione che consente di realizzare *simulatori* pienamente deterministici, così da procedere a fasi di *verifica del progetto mediante simulazione*. Diverso è il discorso per gli strumenti di sintesi automatica, poiché ogni singolo produttore di strumento EDA, al fine di ottimizzare sia il progetto sia la complessità e l'efficienza dello strumento stesso, potrebbe far scelte che porterebbero – a partire dalla stessa descrizione VHDL – a risultati diversi (anche se funzionalmente equivalenti) a seconda dello strumento usato. Negli anni si è arrivati a definire un insieme di "buone pratiche" di specifica e di progetto per ottenere il massimo delle prestazioni dagli strumenti EDA.

FIGURA 3
Vista strutturale del full adder



Riquadro 2

```
architecture structure of full adder is
 component half_adder
  port  (in1, in2: in bit; carry: out bit; sum: out bit);
 end component;
 component or_gate
  port  (in1, in2 : in bit; o: out bit);
 end component;

 signal  x, y, z : bit;   -- segnali locali

 begin    -- connessione delle porte
  i1 : half_adder  port map (a, b, x, y);
  i2 : half_adder  port map (y,  carry_in, z, sum);
  i3 : or_gate  port map (x, y, carry_out);
 end structure;
```

4. Verso nuovi formalismi di descrizione

Molti sforzi sono attualmente legati alla messa a punto di strumenti che supportano il cosiddetto *IP-based design*, ovvero la possibilità di progettare rapidamente sistemi in modo non più verticale, ma utilizzando anche blocchi pre-progettati detti IP (*Intellectual Properties*) che possono essere eventualmente acquisiti da terze parti o sviluppati internamente allo scopo di essere facilmente riusabili. Allo stesso modo si sta cercando (con alterne fortune) di elevare il livello di astrazione dei formalismi per la descrizione dei sistemi, così da potere coprire sia le componenti hardware di un sistema sia quelle software. Probabilmente il linguaggio **SystemC** – una elaborazione del C++ nata nel 1999 e standardizzata da IEEE nel 2005 – è il rappresentante di tale filosofia con le maggiori probabilità di sopravvivenza[2]. SystemC, la cui architettura è mostrata nella figura 4, definisce una libreria di classi C++ in modo da fornire al progettista un insieme di elementi per effettuare modellazioni e simulazioni delle specifiche del sistema con l'accuratezza sino a livello di ciclo (cycle accurate). Le librerie SystemC contengono tutti quei costrutti per modellare sistemi, incluse temporizzazioni hardware, concor-

[1] Lo standard a cui si fa riferimento più comunemente, che assicura anche una ottima portabilità dei progetti, è quello IEEE 1164.

[2] Si veda www.systemc.org per maggiori informazioni circa tale iniziativa.

renza e comportamento reattivo che non sono nativi del C++. L'obiettivo è quello di consentire anche a chi ha competenze di sviluppatore esclusivamente software di muovere passi significativi nella progettazione di sistemi hw/sw complessi. La metodologia di progetto basata su SystemC fornisce la possibilità di utilizzare un unico formalismo per Hw e Sw, rimanendo al livello funzionale per la verifica del comportamento, e muovendo poi verso livelli incrementali di dettaglio sino a giungere al punto in cui esiste uno strumento di sintesi automatica. Purtroppo, almeno per ora, la mancanza di strumenti di sintesi con la maturità ed efficienza di quelli basati su VHDL porta i progettisti a ricorrere ai canonici flussi e strumenti commerciali per la sintesi di hardware reale. Ciononostante, la possibilità di creare modelli (dalla versione 2.0) con un livello di astrazione detto *transazionale* (TLM, *Transactional Level Model*), in virtù della efficienza in fase di simulazione e della potenza rappresentativa ne sta favorendo l'accettazione da parte dei progettisti. Uno stile di descrizione TLM porta a rappresentare i componenti di un sistema in modo affine all'invocazione di metodi remoti: vi sono moduli (SC_MODULE) che si interfacciano in modo più astratto rispetto al concetto di segnale, mediante invocazioni dirette (sc_port) di metodi realizzati da un modulo ed esportati (sc_export) tramite canali (sc_channel). Ovviamente, come accade per il VHDL, i costrutti utilizzabili per la sintesi delle parti hardware sono una restrizione rispetto a quelli messi a disposizione dal linguaggio. Per tale motivo la fase di sintesi prevede una serie di passaggi per raffinare la specifica così da renderla implementabile, ad esempio convertendo i tipi del linguaggio nella precisione effettivamente necessaria, eliminando le chiamate di sistema operativo, utilizzando solo costrutti per cui esiste semantica hardware ben definita (escludendo quindi allocazione dinamica della memoria, la ricorsione o l'uso del goto).

I vantaggi legati al livello di astrazione TLM sono particolarmente significativi quando si deve raffinare il modello del sistema, visto che si può iniziare l'analisi dagli aspetti puramente funzionali, rimandando la risoluzione delle questioni di dettaglio della comunicazione fra i moduli ad un momento successivo della progettazione, quando la sperimentazione di alternative è pressoché conclusa. Un semplice esempio di modellazione TLM in SystemC è riportato nel riquadro 3, dove si riporta un modello semplificato di bus che supporta operazioni *burst* di lettura e scrittura, senza entrare nei dettagli di aspetti più legati al mondo reale come l'arbitraggio, la risposta alle interruzioni o i *wait state* della memoria. Tali elementi legati all'attività del bus ad ogni ciclo di clock, non riportati per motivi di spazio nel testo, sono comunque rappresentabili rimanendo sempre al livello TLM della descrizione.

Nella sezione precedente si è accennato al linguaggio Verilog, alternativo al VHDL e più diffuso negli USA, molte caratteristiche del quale sono affini a quelle del VHDL ed il cui ruolo nella progettazione può essere considerato analogo. Le versioni a partire dalla 3.0 del 2003 sono note come **SystemVerilog** e contengono estensioni che ne ampliano potenzialità e livello di astrazio-

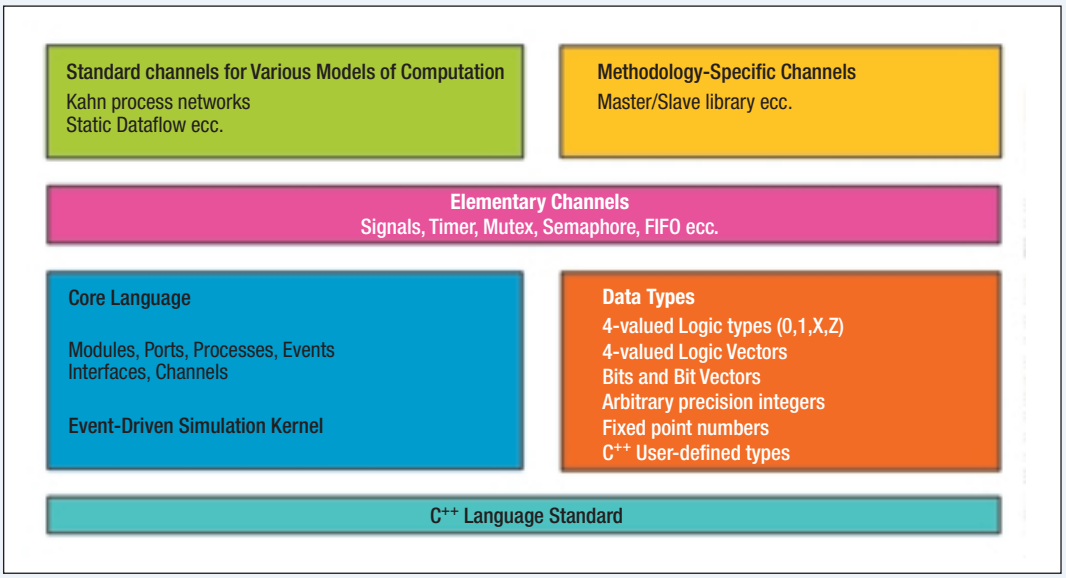


FIGURA 4
Architettura del linguaggio SystemC

Riquadro 3

```
class very_simple_bus_if : virtual public sc_interface
{
public:
      virtual void burst_read   (char  *data,
                        unsigned  addr,
                        unsigned  length)  = 0;
      virtual void burst_write   (char *data,
                        unsigned  addr,
                        unsigned  length)  = 0;
}

class very_simple_bus
      : public  very_simple_bus_if ,
      public  sc_channel
{
public:
      very_simple_bus(sc_module_name nm,  unsigned mem_size,
      sc_time  cycle_time)   :  sc_channel  (nm),
                                  _cycle_time  (cycle_time)
      {
// uso  di  un  array  per modellare l'accesso alla memoria
      _mem = new  char [mem_size] ;
// inizializzazione  a  zero  della  memoria
      memset  (_mem,  0,  mem_size) ;
      }

very_simple_bus () {delete [ ]  _mem ; }

virtual   void   burst_read (char *data ,  unsigned  addr,
                                 unsigned   length)
{
// uso di un mutex per modellare la contesa, ma senza arbitraggio

_bus_mutex.lock () ;

//  blocco  del chiamante  per  la durata  del  burst
wait (length   *_cycle_time);
//  copia  dei  dati  dalla  memoria  a  chi  ha  fatto  la  richiesta
memcpy (data, _mem + addr ,  length);
//  sblocco   del  mutex  per  consentire  l'accesso  al  bus  ad  altri
_bus_mutex.unlock () ;
}

virtual   void   burst_write (char   *data,   unsigned   addr,
                                 unsigned length)
{
_bus_mutex. lock () ;
wait  (length    *_cycle_time) ;
//  copia   dati   dal   richiedente alla memoria
memcpy (_mem + addr, data, length);
_bus_mutex. unlock () ;
}

protected:
      char*  _mem;
      sc_time   _cycle_time;
      sc_mutex _bus_mutex;
};
```

ne: consentono per esempio di chiamare funzioni C/C++, creare processi dinamicamente, standardizzare la comunicazione e sincronizzazione tra processi (inclusi i semafori) e predisporre una interfaccia standard per la verifica formale. La capacità di utilizzare anche il C/C++ - che dovrebbe rendere possibile l'interfacciamento verso modelli SystemC - un sistema di simulazione migliorato e l'apertura verso la verifica formale sono i pilastri principali dei fautori di tale linguaggio.

Bibliografia

[1] Ashenden P.: *VHDL Cookbook*. Testo liberamente disponibile in rete.

[2] Grotker T., Liao S., Martin G., Swan S.: *System Design with SystemC*. Kluwer Academic Publisher, 2002. ISBN: 1-4020-7072-1.

[3] Mentor Grahpics (EDA Vendor): www.mentor.com

[4] Synopsys (EDA Vendor): www.synopsys.com

[5] Lattice Semiconductor (produttore logiche programmabili): www.latticesemi.com

[6] Xilinx (produttore logiche programmabili): www.xilinx.com

[7] Altera (produttore logiche programmabili): www.altera.com

WILLIAM FORNACIARI si è laureato con lode in Ingegneria Elettronica (1989), ha svolto il Dottorato di Ricerca in Ing. Informatica e Automatica (1992), è stato ricercatore (1995) e dal 2001 è professore associato presso il Dipartimento di Elettronica e Informazione del Politecnico di Milano. Fra il 1993 e il 2005 è stato responsabile della embedded systems design unit (ESD) del centro di Ricerca CEFRIEL, di cui ora è mentor scientifico. È stato membro dello steering committee del VHDL user's group italiano e chairman dei primi workshop "VHDL Users Design Practice" svolti nel 1993 e 1994. Dal 1992 ricopre ruoli in committee di conferenze internazionali nell'ambito dei sistemi digitali e della progettazione a livello di sistema. È autore di oltre 100 pubblicazioni scientifiche, per cui ha ricevuto tre best paper awards (IEEE-ICONIP'95, IEEE-IJCNN'92 e IEEE-ICCD'98) e un Certification of Appreciation da parte della IEEE Circuits and Systems Society. I sui interessi scientifici sono legati alla progettazione di sistemi embedded, Hw-Sw codesign, Wireless Sensor Networks e sistemi a basso consumo di potenza.
E-mail: william.fornaciari@polimi.it

MARIAGIOVANNA SAMI è professore ordinario di prima fascia presso il Politecnico di Milano, nell'area dei Sistemi di Elaborazione. Dal 1987 al 1990 è stata Direttore del Dipartimento di Elettronica del Politecnico di Milano. È Direttore Scientifico del Master of Science in Embedded Systems Design presso l'Università della Svizzera Italiana a Lugano.
La sua attività di ricerca riguarda le architetture hardware dei sistemi digitali, con particolare riguardo alle metodologie di progetto di sistemi dedicati caratterizzati da elevate prestazioni, robustezza e basso consumo di potenza. È autrice o co-autrice di oltre duecento lavori scientifici in sede internazionale ed ha ricevuto alcuni premi per la sua attività di ricerca. È membro dell'Accademia Italiana delle Scienze (detta dei Quaranta).
E-mail: sami@elet.polimi.it

PROTOCOLLI E TELECOMUNICAZIONI
ALLA RICERCA DEL VALORE AGGIUNTO

Tracciare l'evoluzione dei protocolli significa tracciare l'evoluzione delle telecomunicazioni. Ovvero parlare di ICT anziché di TLC. Il valore aggiunto, insito nei protocolli di rete, si è evoluto passando dalla correzione d'errore, alla gestione del dialogo, alle funzioni collaborative per arrivare infine ai servizi applicativi, sempre più sofisticati e orientati all'erogazione dei servizi web. La pietra miliare è il modello OSI, sempre attuale, rivisto alla luce delle nuove tecnologie di rete wireless e dalla necessità di definire protocolli che supportino ambienti multimediali.

Andrea Baiocchi
Giacomo Zanotti

3·4

1. INTRODUZIONE

L'enciclopedia gratuita offerta su Internet, la ormai famosa Wikipedia (http://en.wikipedia.org/wiki/Main_Page), fornisce la definizione seguente del termine protocollo di comunicazione: *a set of rules governing communication between electronic devices*.
Dobbiamo notare che il generico termine protocollo è riferito a varie altre categorie, oltre alle comunicazioni: diplomazia e politica, informatica, medicina e scienza, letteratura e spettacoli. È interessante il fatto che per cercare la definizione corrente del termine protocollo con le sue varie accezioni, inclusa quella qui oggetto di trattazione nell'ambito delle telecomunicazioni, sono state invocate una complessa messe di protocolli di comunicazione: l'HTTP per dialogare con il programma applicativo che gestisce le pagine web di Wikipedia; i protocolli della famiglia TCP/IP per effettuare un trasferimento affidabile dei dati dal server remoto alla macchina client, per risalire dal nome del destinatario al suo indirizzo di rete, per determinare un percorso interno alle reti attraversate fino all'indirizzo di destinazione, per mantenere la continuità del collegamento; i protocolli specificati negli standard della famiglia IEEE 802 per accedere attraverso la rete locale Ethernet alla quale la macchina cliente è connessa nell'esempio in esame. Ma andiamo con ordine, per dipanare la matassa di sigle che si affastella non appena si esamina anche la più semplice attività di interconnessione in rete.
Il problema base della comunicazione tra entità remote può essere schematizzato come segue: una popolazione di entità, consistenti in esseri umani o processi eseguiti automaticamente da macchine, in particolare computer, e distribuiti geograficamente su una data area, hanno necessità di cooperare con altre entità facenti parte della medesima popolazione per conseguire i propri obbiettivi funzionali. La "cooperazione" può essere vista in generale come l'evoluzione di una macchina a stati distribuita, in cui ogni entità svolge azioni che dipendono dai dati e programmi/modi di procedere residenti localmente e dalle sollecitazioni esterne, in particolare dai risultati delle comunicazioni con altre entità della po-

polazione. Informalmente, i protocolli specificano le regole di questo tipo di interazione.
Un semplice esempio, per rendere meno astratta la definizione, ma che non rende necessariamente l'intera complessità che vi si nasconde, è una popolazione di esseri umani dispersi in un dato territorio che necessitano di effettuare conversazioni a voce. Il problema di comunicazione consiste, ridotto all'osso, nel fornire mezzi per cercare e allertare un destinatario da parte di un originatore, nel permettere il trasferimento intelleggibile della voce in modo bidirezionale tra le due parti durante la conversazione, eventualmente impiegando risorse di comunicazione condivise dai membri della popolazione di utenti, nel fornire i mezzi per terminare ordinatamente la conversazione e rilasciare eventuali risorse condivise utilizzate. Altro esempio è l'esecuzione di transazioni tra macchine (*mail server*) per la distribuzione della posta elettronica. Anche in questo caso il paradigma comporta la necessità di scoprire i potenziali partner, contattarli, negoziare le condizioni per l'esecuzione della transazione, trasferire dati e informazioni di controllo relativi alla transazione tra le macchine coinvolte, quindi terminare la transazione. Ogni passo di questo processo in ogni mail server è controllato da un programma (software). Questo software elabora i contenuti informativi scambiati (per esempio esegue traduzioni di codifica, memorizza in archivi) ed esegue i protocolli di comunicazione che permettono la cooperazione tra i processi residenti su macchine remote. In ambedue gli esempi, conversazioni e distribuzione della posta, sistemi intermedi con i loro protocolli, programmi e dati intervengono ad un livello più "basso" di quello dell'applicazione. Scopo dei sistemi intermedi è fornire e controllare la connettività e la qualità del servizio di rete: in sintesi, trasferire dati all'interno della popolazione di utenti in modo tecnicamente efficiente (date le tecnologie disponibili) ed economicamente conveniente (dati gli obiettivi e il valore dei servizi supportati).
Cos'è quindi un protocollo di comunicazione? La definizione data all'inizio lo identifica come un insieme di regole che permettono uno scambio di informazioni tra apparati elettronici, o più in generale, come illustrato nei due semplici esempi, tra processi applicativi (in senso lato, inclusi dunque gli esseri umani). È cruciale precisare il termine "scambio": lo scambio di informazioni fa pensare a un passaggio di dati da un'entità ad un'altra, eventualmente in modo bidirezionale. Focalizzare l'attenzione sulla fase di "scambio" dei dati elude però la parte forse più importante di un protocollo che è il "controllo" della scambio, cioè la definizione di procedure distribuite che permettono alle entità coinvolte di arrivare a scambiare i dati, controllando le condizioni tecnico-economiche nelle quali lo scambio avviene (ecco il valore aggiunto!).
A questo scopo un generico protocollo comprende tre parti costitutive:

- una *sintassi*, che precisa le strutture dei messaggi (formato, parti componenti, possibili valori attribuibili alle varie parti) usati nel protocollo con finalità di controllo e di trasferimento di dati;
- una *semantica*, che stabilisce il significato dei particolari valori inseriti nelle parti di ogni messaggio e degli eventi che si presentano nell'evoluzione del protocollo, anche in rapporto alle sequenza di messaggi scambiati a partire dall'inizio dell'interazione, e definisce quindi le azioni che una entità deve intraprendere; si tratta in sostanza della specificazione della logica della macchina a stati eseguita da ogni entità partecipe del protocollo;
- *temporizzatori*, che presiedono alla corretta evoluzione nel tempo del protocollo, ovvero delle macchine a stati associate al protocollo in ogni entità che prende parte alla sua esecuzione; scopo essenziale dei temporizzatori (*timer*) è regolare la "marcia" dell'evoluzione del protocollo, verificando che le transizioni di stato avvengano in lassi di tempo coerenti con una corretta funzionalità.

Nel riquadro di p. 23 è illustrato qualche esempio per ognuno dei tre concetti.
Notiamo che, in un protocollo, le funzioni "collaborative", anche quando comportano attività basilari, possono richiedere l'utilizzo di algoritmi molto complessi o anche condurre a situazioni (stati dell'automa) "indecidibili". In questo ultimo caso, il protocollo non è in grado di risolvere univocamente la situazione ma si limita ad "avvisare" tutti gli attori che partecipano al processo collaborativo che si è verificata una situazione anomala e che dunque è necessa-

Esempi di aspetti sintattici, semantici e di temporizzazione dei protocolli

Un'unità d'informazione (pacchetto) di un protocollo per il trasferimento dei dati in rete, quale per esempio IP, è descritta da una sequenza di ottetti (byte) raggruppati a formare cosiddetti campi. Ogni campo può avere solo determinati valori (dominio) e rappresenta un elemento d'informazione portato dal pacchetto: per esempio alcuni byte contengono l'indirizzo del mittente del pacchetto (interfaccia di rete dalla quale è originato il pacchetto), altri, l'indirizzo del destinatario (interfaccia alla quale è connessa l'entità destinataria finale del pacchetto). Altri byte possono contenere codici per il controllo di errori, altri il numero della versione del protocollo (anche i protocolli di comunicazione hanno una vita ed evolvono, come il software che installiamo sui nostri computer: hanno quindi un numero di versione, per motivi di verifica di compatibilità).

Seguendo il filo di questo esempio, la semantica specifica la sequenza di operazioni da fare con un pacchetto da parte di una generica entità di rete che lo riceve. Per esempio:

1. verifica la presenza di errori nel pacchetto;
2. se corretto, verifica se trattasi di versione compatibile con quelle disponibili localmente;
3. verifica se l'indirizzo destinatario coincide con uno degli indirizzi attribuiti all'entità;
4. in caso contrario, verifica se rientra tra gli indirizzi di una porzione di rete connessa all'entità, oppure se è un indirizzo di gruppo multicast; in tal caso si passa al punto 6;
5. scarta il pacchetto e notifica il mittente che l'indirizzo di destinazione non è noto;
6. consulta la tabella di instradamento per decidere su quale porta di uscita rilanciare il pacchetto.

Infine, un esempio di uso di temporizzatori si ha nel caso di controllo della corretta ricezione di un messaggio effettuato mediante riscontri: una entità A invia un messaggio a B ed attende da B una esplicita notifica di ricezione corretta del messaggio. Quando deve aspettare A? Come fa a sapere che "è troppo tardi"? Una risposta accurata può essere anche molto difficile da dare, per esempio nel caso di ritardo di trasferimento tra A e B variabile da pacchetto a pacchetto. È però chiara l'esigenza di mettere un termine all'attesa del riscontro da parte di A per evitare di mandare il protocollo in uno stato di stallo. La soluzione adottata universalmente è definire un "time-out", cioè un contatore che scade dopo un tempo prefissato a partire dall'invio dei dati. Se entro questa scadenza non si è ricevuto riscontro, si assume che i dati non siano giunti a destinazione. Porre una terminazione certa all'attesa del riscontro è un requisito esattamente analogo a quello di terminazione di un algoritmo. Con la difficoltà enorme del fatto che l"algoritmo" in questione (eseguito dal protocollo) è intrinsecamente di natura distribuita e prevede la collaborazione di sistemi di elaborazione remoti.

rio ritornare all'ultima situazione concordata, annullando tutte le situazioni pendenti (processo di uccisione degli orfani). Un esempio classico è costituito dal processo collaborativo per cui si aggiornano contemporaneamente più basi di dati distribuite. L'algoritmo utilizzato implica una regia piuttosto complessa in più fasi (*prepare, commit, rollback*); in caso di problemi si torna all'ultima condizione certa (*rollback*).

In altre situazioni è necessario ricorrere ad algoritmi "a latere", molto sofisticati e apparentemente scollegati dal processo in esame. Un esempio semplice, ma concreto, è costituito dalla banale richiesta di conferma della ricezione di un documento da parte del destinatario (si intende qui la ricezione effettiva, ossia la verifica che il documento sia stato effettivamente aperto sul PC del destinatario e non il più semplice problema della consegna "postale"). La certezza dell'effettivo display del documento si ottiene con algoritmi complessi di tipo "notarile" (ovvero è necessario l'intervento di un garante "esterno" che esegua ogni richiesta in duplice formato: in chiaro e "crittografata" con la chiave pubblica del mittente in modo che si possa sempre dimostrare che non ci sono discrepanze tra i documenti che il destinatario ha accettato di ricevere e quelli che gli vengono effettivamente consegnati).

Nel seguito dell'articolo, dopo una breve storia dei protocolli (paragrafo 2), sono presentate le idee di base ormai accettate e consolidate nelle architetture di protocolli di comunicazione. In questo contesto sono introdotte le architetture di comunicazione e si chiarisce il nesso con l'aggiunta di valore, caratteristica saliente di un vero protocollo (almeno di quelli utili). È anche brevemente discusso l'impatto sulle architetture di protocolli di requisiti trasversali: la qualità di servizio, la sicurezza nella comunicazione e il risparmio energetico. Spazio è dedicato nel paragrafo 4 a discutere il nesso tra il modello architetturale introdotto nel paragrafo 3 e la pila protocollare di Internet. Il paragrafo 5 contiene uno sguardo ad approcci "alternativi", in particolare al *cross-layering*, per il loro interesse negli sviluppi di specifiche importanti tecnologie come il *wireless*.

Esempi tratti dalle principali architetture di protocolli sono diffusi nel testo nel tentativo di rendere afferrabile ogni definizione astratta e di mostrare come si concretano le soluzioni a problemi di comunicazione in rete attraverso meccanismi protocollari.

Spesso questi contengono sottigliezze e dettagli che richiedono profonde conoscenze e riflessione attenta per essere comprese; ma altrettanto spesso, procedure protocollari infarcite di sigle e di tecnicismi sono il buon senso codificato!

2. UNA BREVE STORIA

I primi protocolli nascono per risolvere il problema della trasmissione remota di dati, intrinsecamente legato al problema della condivisione di una risorsa, ai tempi preziosissima, quale la banda trasmissiva e la capacità di elaborazione.
Normalmente sono noti con il nome di "protocolli orientati al carattere" [1, 3] in quanto il meccanismo per gestire la "regia" del dialogo consiste nell'eleggere alcuni caratteri a funzioni di controllo: per esempio un carattere rappresenta l'inizio del blocco di dati (STX, *Start of Text*), un altro la fine (EOT, *End of Transmssion*), un altro ancora la conferma di ricezione corretta – o meno – (ACK o NACK, *Acknowledgement o Not Acknowledgement*).
Questi protocolli sono quasi sempre utilizzati su una linea multipoint (ovvero su una linea condivisa da più dispositivi), per cui é necessario introdurre un meccanismo selettivo di trasmissione/ricezione (un meccanismo possibile é il *polling/selecting*, ovvero l'appello a turno ciclico per autorizzare la trasmissione).
I limiti di questi protocolli sono moltissimi. Ne elenchiamo i principali:

- ❑ non trasparenza al testo. Il testo non può contenere caratteri codificati come uno dei caratteri di controllo (per evitare ambiguità). Il problema si supera con un ulteriore livello di caratteri di controllo (DLE, *Data Link Escape*, per segnalare che il carattere successivo va inteso come testo e non come controllo);
- ❑ non trasparenza al controllo. L'inserimento di un nuovo carattere di controllo (per implementare una nuova funzione), può far insorgere ancora problemi di trasparenza;
- ❑ assenza di controllo di flusso: non é possibile inviare un nuovo blocco di informazioni se il ricevente non ha dato l'ACK a quello precedente (al massimo si può pensare a due blocchi, con una logica di ACK pari e dispari);
- ❑ ambiguità nei caratteri di controllo. Spieghiamo con un esempio: un ACK può essere interpretato a livello trasmissivo, oppure applicativo, oppure transazionale (ho ricevuto il blocco, l'ho ricevuto e l'ho passato all'applicazione, l'applicazione l'ha preso in carico ed ha logicamente chiuso la transazione). Questo problema é in assoluto il più grave in quanto rende il dialogo fortemente dipendente dal contesto in cui é inserito, a livello fisico e logico;
- ❑ povertà del controllo di errore, basato sui bit di parità;
- ❑ impossibilità di gestire (orchestrare) più flussi contemporanei.

Un passo importante nell'evoluzione dei protocolli é costituito dall'introduzione (da parte di IBM) dell'SDLC (*Synchronous Data Link Control*, [2]), da cui é derivato l'importantissimo (e tuttora vivo) standard HDLC (*Higher-Level Data Link Control*, [4]). HDLC e SDLC sono strutturalmente identici, ma sono utilizzati in contesti diversi. Non é importante capirne le differenze, ma é importante ragionare sulla novità introdotta.
Essi sono chiamati "protocolli orientati al bit" (in contrapposizione a quelli precedenti, orientati al carattere), perché si supera il concetto di carattere di controllo, ma si introduce quello di "trama" o "cornice". Ovvero la busta trasmissiva ha una struttura predefinita - inizio, campo indirizzo, caratteri di controllo, testo, controllo di errore, fine - per cui una sequenza di bit ha un significato per la posizione occupata all'interno della trama e non solo per la sua codifica. L'*n*-simo byte sarà sempre e solo un carattere di controllo (o di testo), in relazione alla sua posizione, e così sarà comunque interpretato.
Con i protocolli orientati al bit si superano enormi problemi.
Il testo é trasparente ed indipendente dalla codifica, a patto di rendere univoca la delimitazione delle trame (per esempio con meccanismi di *bit stuffing*).
Il campo indirizzo permette di indirizzare un numero elevato di stazioni, lavorando su linee punto a punto oppure su linee multipunto.
Il campo controllo contiene molteplici funzioni; una delle più importanti é la numerazione dei blocchi in trasmissione e ricezione, secondo un modulo aritmetico predefinito (per esempio 8 o 128). I numeri in trasmissione individuano la sequenza dei

blocchi in trasmissione, i numeri in ricezione individuano il limite superiore dei blocchi ricevuti correttamente nel verso opposto. Questo meccanismo (chiamato finestra) permette di avere un flusso trasmissivo regolabile, basato su meccanismi semplici e potenti di controllo di flusso e di errore. La regolazione del flusso riguarda ogni aspetto trasmissivo: sono infatti risolti problemi di sequenzialità e di integrità (per effetto della numerazione), di regolazione della quantità di informazioni in transito (si può arrivare a spegnere completamente il dialogo in una direzione, non aggiornando il limite inferiore della finestra e quindi esercitare un completo controllo di flusso oppure dargli il massimo dell'apertura aggiornando continuamente il limite inferiore della finestra in trasmissione), di regolazione del turno (il flusso può essere mono o bidirezionale, essendo i campi di controllo presenti anche in un "frame" di tipo informativo).
Il controllo di errore é basato su meccanismi CRC (*Ciclic Redundancy Check*) – ovvero sul calcolo del resto utilizzando il testo come dividendo e un polinomio standard come divisore; questo algoritmo rende la trasmissione praticamente immune da errore (la probabilità di avere due testi differenti con il medesimo resto é praticamente nulla).
I protocolli orientati al bit risolvono quindi brillantemente il problema della trasmissione tra due entità. Lasciano però, purtroppo, aperti due problemi essenziali:

- l'impossibilità di distinguere tra elementi di controllo trasmissivo ed elementi di controllo applicativo (l'ACK é ancora ambiguo – resistono ancora gli stessi problemi dei protocolli orientati al carattere);
- l'impossibilità di eseguire controlli selettivi all'interno del flusso trasmissivo (il non-aggiornamento del limite inferiore della finestra determina il blocco completo a livello trasmissivo, senza la possibilità di operare in modo selettivo su eventuali differenti flussi applicativi in essere tra i due interlocutori).

Contestualmente, dalla fine degli anni '60, sono stati messi a punto protocolli per il trasferimento dei dati a pacchetto in reti geografiche. Alcune delle idee più feconde sono maturate con un lento processo di *trial & error* nell'ambito del progetto ARPANET, il progenitore di Internet (vedi http://www.isoc.org/internet/history/; [5]).
I problemi della comunicazione di dati e la strutturazione dei relativi protocolli hanno portato nei primi anni '80 al maturare del cosiddetto modello OSI, oggetto del paragrafo seguente.

3. LE IDEE DI BASE DELLA MODERNA CONCEZIONE DEI PROTOCOLLI

Progettare e realizzare un protocollo è ad oggi ancora un'arte, come alcuni amano dire, o forse più appropriatamente, un'opera di artigianato. Questo non toglie che nel tempo sono stati sviluppati modelli e strumenti per rendere sistematico e verificabile il lavoro di definizione e sviluppo di un protocollo.
Il più famoso al riguardo è il modello OSI (*Open System Interconnection*), emesso come Standard ISO 7498 nel 1983 e recepito quindi come Raccomandazione ITU X.200. Il modello si propone di definire una architettura funzionale completa per descrivere la cooperazione di processi remoti, residenti cioè in sistemi distinti e comunicanti tra loro, in altre parole in grado di trasferire stringhe binarie con date caratteristiche di prestazione in termini di capacità e affidabilità. Punti chiave del modello sono la decomposizione del complesso delle funzioni di comunicazione e di elaborazione locale dell'informazione in sotto-insiemi strutturati gerarchicamente e la formalizzazione delle interazioni e dei flussi informativi nelle varie interfacce identificate nell'architettura. Questi aspetti sono l'eredità ancora viva del modello OSI, perfettamente identificabili e anzi alla base dello sviluppo di standard complessi relativi a qualsiasi tipo di rete. È quindi su questi aspetti che sarà posto l'accento nel seguito; per una descrizione esauriente del modello OSI si possono consultare, oltre agli standard, che ne sono la fonte primaria, molti lavori (ottima è la descrizione del modello OSI contenuta nella documentazione tecnica Cisco [6]).
L'idea di base del modello è che l'eterogeneità e la vastità delle funzioni coinvolte nel processo di cooperazione è affrontabile al meglio solo suddividendo queste funzioni in

gruppi omogenei per livello di astrazione del trattamento delle informazioni e per affinità tecnologica di realizzazione.
L'equalizzazione del canale trasmissivo, il recupero del cronosegnale associato al segnale numerico ricevuto, la rivelazione di errori nei blocchi di informazione trasferiti, l'instradamento dell'informazione tra i sistemi terminali dello scambio informativo attraverso eventuali sistemi intermedi, la transcodifica dell'informazione tra formati diversi di rappresentazione (per esempio, da mp3 a wav per un file audio; oppure trancodifiche tra i diversi formati adottati per la voce nelle reti cellulari o nell'interconnessione tra reti telefoniche e *VoIP*) sono tutti esempi di funzioni generalmente eseguite in ogni singola istanza di cooperazione tra processi remoti e chiaramente eterogenee tra loro. Le prime due funzioni possono essere realizzate tipicamente con hardware o firmware e considerano l'informazione trasferita tra due sistemi adiacenti (senza sistemi intermedi) come un semplice flusso di simboli, per esempio binari. La rivelazione di errori è realizzabile solo articolando l'informazione in blocchi delimitati e strutturati ed è spesso realizzata in software microprogrammato su schede dedicate. L'instradamento è una funzione di natura logica, che coinvolge più sistemi, richiede l'esecuzione di algoritmi e procedure tipicamente realizzati in software all'interno del *kernel* del sistema operativo e tratta informazioni assunte esenti da errori e strutturate in messaggi indirizzabili. Infine, la transcodifica si applica a dati di utente, è eseguita localmente dal processo che ha ricevuto l'informazione, assunta esente da errori e ricostruita nella sua interezza così come la sorgente l'ha generata.
Anche dagli esempi appena accennati è evidente che si possono individuare gruppi di funzioni separati, la cui esecuzione implica l'interazione di sistemi diversi, o meglio, di quelle parti dei sistemi che presiedono al gruppo funzionale considerato. È anche chiaro che le funzioni così raggruppate non sono svolte in un ordine qualsiasi, ma ordinate in modo gerarchico. Per il destinatario dell'informazione e con riferimento agli esempi fatti sopra, occorre prima rivelare il flusso di informazione ricevuto, eseguendo tra le altre l'equalizzazione e il recupero della sincronizzazione, quindi delimitare blocchi di informazione significativi e accertarsi della loro correttezza, recuperando eventuali errori, e così via.
Nasce quindi il fondamentale concetto di *strato*. Uno strato dell'architettura di comunicazione è l'insieme delle *entità* appartenenti a tutti i sistemi che svolgono un dato gruppo di funzioni. Le entità in concreto sono codice, hardware, schede; la loro cooperazione avviene mediante *protocolli*, che hanno appunto lo scopo di svolgere le funzioni relative allo strato cui appartengono le entità. Dal concetto di strato e di stratificazione delle funzioni nasce il valore aggiunto: ogni strato aggiunge "valore" al processo di cooperazione tra i processi remoti, mediante l'azione svolta dalle proprie entità nell'esecuzione dei protocolli cui sono preposte. È per esempio valore aggiunto il poter garantire l'affidabilità del trasferimento informativo ad un livello desiderato data un'infrastruttura fisica di per se non soddisfacente i requisiti. Oppure la capacità di recapitare al corretto destinatario l'informazione che gli compete, permettendo in ultima analisi l'interconnessione mediante una rete con condivisione delle risorse, piuttosto di una connessione diretta fisica, con risorse dedicate, tra ogni coppia di sistemi che debbano cooperare.
Il meccanismo di comunicazione tra processi in un'architettura stratificata è accuratamente descritto dal modello OSI, ma può forse essere esemplificato e compreso nella sua essenza con un colorito esempio, ispirato da Andrew S. Tanenbaum [1, paragrafo 1.3]. Un filosofo cinese e un bramino indiano desiderano dialogare; dato che parlano lingue diverse, ognuno dei due ingaggia un interprete, concordando una lingua comune (dal mandarino all'inglese uno, dal sanscrito all'inglese l'altro). Poiché i due interpreti a loro volta si trovano uno a Pechino, l'altro a Benares, essi ingaggiano un telegrafista ognuno e incaricano i telegrafisti di trasmettere i messaggi originati dai filosofi e tradotti in inglese. È chiaro che il bramino che esprime un pensiero sta (virtualmente) dialogando con il filosofo cinese e NON con il proprio interprete; per attuare questo dialo-

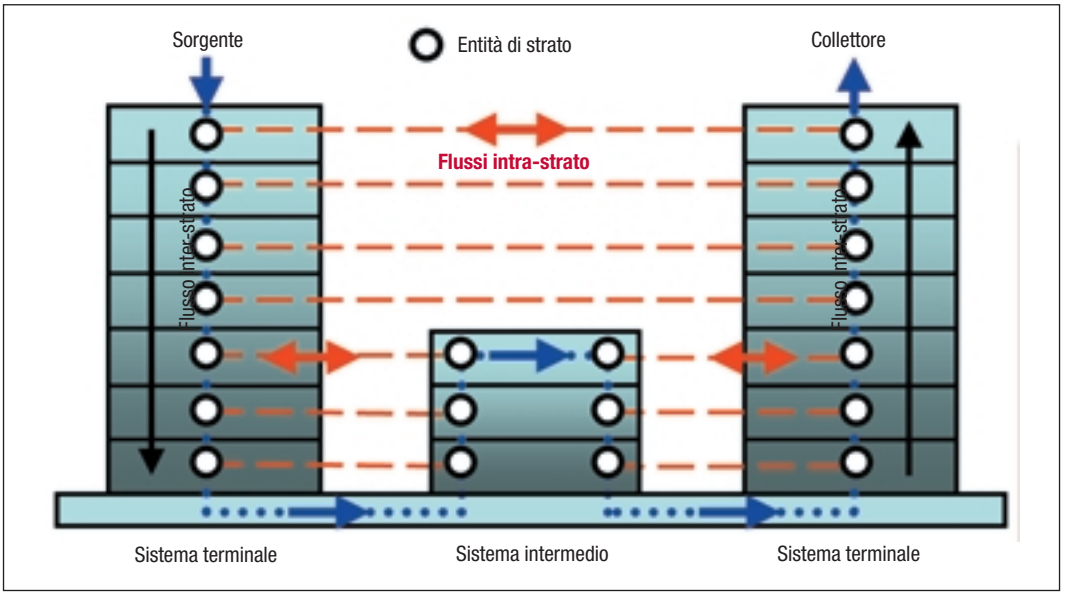


FIGURA 1
Trasferimento dell'informazione in un'architettura a strati

go tuttavia, egli affida fisicamente il proprio messaggio all'interprete, il quale a sua volta dialoga (virtualmente) con l'altro interprete; per far questo egli affida fisicamente il messaggio al telegrafista che lo trasmette tramite un mezzo fisico di comunicazione al telegrafista remoto. Approdato qui, il messaggio risale la "pila" in senso inverso fino al filosofo. Il paradigma di comunicazione è quindi "orizzontale" virtualmente, cioè i protocolli di comunicazione e cooperazione sono eseguiti tra *entità alla pari* (*peer entities*). È "verticale" nelle modalità di attuazione, cioè le entità di uno strato (dette appunto "utenti") si affidano a quelle dello strato gerarchicamente inferiore (dette "serventi") per supportare il proprio dialogo (Figura 1).
L'OSI individua sette strati (Figura 2). I quattro più in basso, in ordine Fisico (*Physical*), di Collegamento (*Data Link*), di Rete (*Network*) e di Trasporto (*Transport*) sono relativi alle funzioni di trasferimento dell'informazione. I tre più in alto, Sessione (*Session*), Presentazione (*Presentation*) e Applicativo (*Application*), sono relativi al trattamento locale dell'informazione. Questa particolare suddivisione non è che una tra le possibili (e certo non tra quelle di maggior successo). Il concetto di strato funzionale è invece diventato pervasivo e costituisce il mattone base di tutte le architetture di comunicazione concepite dagli anni '80 in poi (riquadro a p. 28).

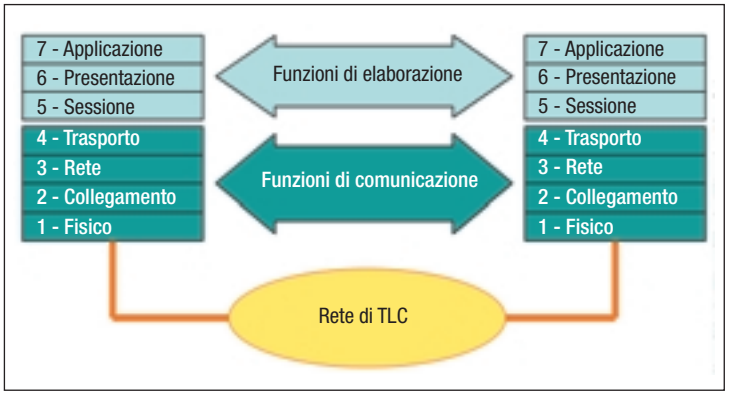


FIGURA 2
Schema dell'architettura OSI

Il modello introduce una descrizione generale degli strati e dei loro elementi componenti, riferendosi ad un generico (*N*)-strato, con *N* intero positivo.
Un elemento basilare dei protocolli è la definizione delle unità informative scambiate dalle entità di un dato strato, le così dette (*N*)-PDU (*Protocol Data Unit* dello strato *N*). Le trame di un protocollo MAC, i segmenti del TCP, i pacchetti IP sono tutti esempi di PDU. Il contributo del modello OSI sta nell'aver formalizzato gli elementi comuni di cui ogni PDU è formata, astraendo dai dettagli implementativi:
- una PCI (*Protocol Control Information*), cioè l'insieme delle informazioni di controllo che occorrono perché la PDU concorra all'assolvimento delle funzioni del protocollo;

I sette livelli proposti nel modello OSI

Livello "1": il livello fisico. É responsabile della codifica del segnale a livello elettrico, delle procedure di "handshaking" per la creazione e il mantenimento dei canali trasmissivi e anche delle specifiche strutturali dei connettori e dei mezzi di trasmissione. Le sue funzioni variano secondo la natura del canale trasmissivo.

Livello "2" il livello di collegamento. É responsabile della correttezza e della integrità dei dati trasmessi. I meccanismi utilizzati sono quelli descritti nel paragrafo 2.

Livello "3": il livello di rete. É responsabile dell'instradamento delle informazioni. É uno dei livelli più complessi e può essere "connesso" o "non connesso": un protocollo connesso invia dati al sistema terminale (DTE) destinatario solo dopo una fase di negoziazione in cui i due DTE concordano sulla modalità di trasmissione (dal banale "essere pronti a ricevere" ad una complessa negoziazione sulla qualità del servizio). Ci sono ovviamente vantaggi (e svantaggi) in ognuna delle due soluzioni. Esempi di protocolli connessi sono l'X.25 e l'ATM; un esempio universale di protocollo non connesso é l'IP di Internet (per inciso, questa affermazione ci permette di capire perché Internet non é nativamente adatta a gestire connessioni in cui é richiesta una qualità garantita, ad esempio la voce!).

Livello "4": il livello di trasporto. É il primo livello di dialogo tra i due DTE ed é responsabile della correttezza della trasmissione da estremo a estremo (end-to-end). Notiamo che il livello "2" e "3" hanno un significato locale: solo i due interlocutori remoti possono infatti avere il controllo completo (e inconfutabile) della correttezza della trasmissione. Ciò é evidente in caso di anomalie della rete (perdita di blocchi di informazione). Le funzioni del livello "4" variano in funzione della qualità della rete di trasporto. Altre importanti funzioni di questo livello sono il multiplexing (ovvero l'affasciamento contemporaneo di più connessioni eventualmente in essere tra i due DTE remoti), il controllo di congestione e il controllo di flusso.

Livello "5": il livello di sessione. Cominciamo ad essere vicini al mondo applicativo. Il livello "5" si occupa degli strumenti che servono alla gestione del dialogo applicativo (che può essere di natura estremamente varia: ad esempio un'applicazione che pilota una stampante remota, oppure due programmi che cooperano per eseguire una transazione congiunta). Il livello di sessione definisce ad esempio i turni di dialogo, mette delle "pietre miliari" per permettere ai programmi di sincronizzarsi su punti concordati, ad esempio in caso di anomalie nel flusso trasmissivo. É importante sottolineare che il significato di questi "paletti" é definito dall'applicativo che li utilizza; ancora dunque, come in tutta l'architettura OSI, essi sono un servizio fornito al livello superiore (l'applicazione, in questo caso) e dunque non hanno un significato assoluto.

Livello "6": é il livello di presentazione. Potrebbe essere inglobato nel livello di sessione. É responsabile della codifica e della presentazione dell'informazione.

Livello "7": il livello applicativo. Qui si possono ingenerare delle confusioni. Il livello applicativo infatti non definisce le "applicazioni", ma i servizi applicativi. Servizio applicativo é ciò che serve all'applicazione finale (il processo di business) per lavorare in un ambiente aperto e distribuito. Un esempio di servizio applicativo universalmente utilizzato é la posta elettronica (non é essa stessa un'applicazione, essendo l'applicazione il client di posta che permette di comporre il messaggio o l'applicazione verticale che muove informazioni usando i meccanismi di posta).

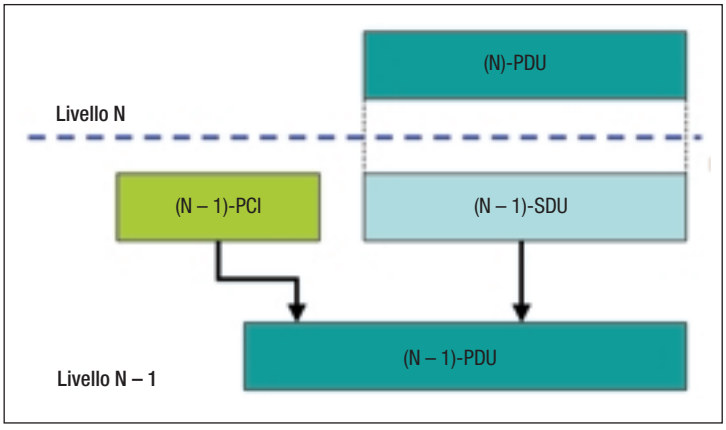


FIGURA 3
Struttura delle unità informative nel modello OSI

- una SDU (*Service Data Unit*), opzionale, parte dedicata al trasporto dei dati offerti dallo strato superiore (Figura 3).

È importante osservare che la definizione di (*N*)-PDU come unione di una (*N*)-PCI e di una (*N*)-SDU è ricorsiva, nel senso che a sua volta la (*N*)-SDU corrisponde (nel caso più semplice) ad una (*N* + 1)-PDU. Il punto di vista dello strato *N* sulla (*N*)-SDU è che essa contiene i dati del livello superiore (l'utente, in senso OSI naturalmente), affidati per il trasferimento alle entità dello strato *N* e trattati come una "scatola nera"; il protocollo dello strato *N* si impegna a effettuare il trasferimento consegnando i dati di utente alle entità di destinazione dello strato *N* + 1, dopo aver aggiunto il proprio "valore" (per esempio, garantito l'assenza di errori e di fuori sequenza entro un fissato margine di probabilità di successo). Gli stessi dati, visti dal livello *N* + 1, sono invece un gruppo strutturato di bit rappresentanti un messaggio dell'(*N* + 1)-protocollo (una (*N* + 1)-PDU), che può a sua volta essere separato in una parte che contiene informazioni di controllo, la (*N* + 1)-PCI, e in una eventuale che contiene i dati del livello superiore *N* + 2, la (*N* + 1)-SDU.

Può sembrare un modello astratto, artificiale e non corrispondente alla multiforme realtà dei protocolli reali. In effetti, è una delle chiavi interpretative e di razionalizzazione nella definizione e sviluppo di protocolli più utili

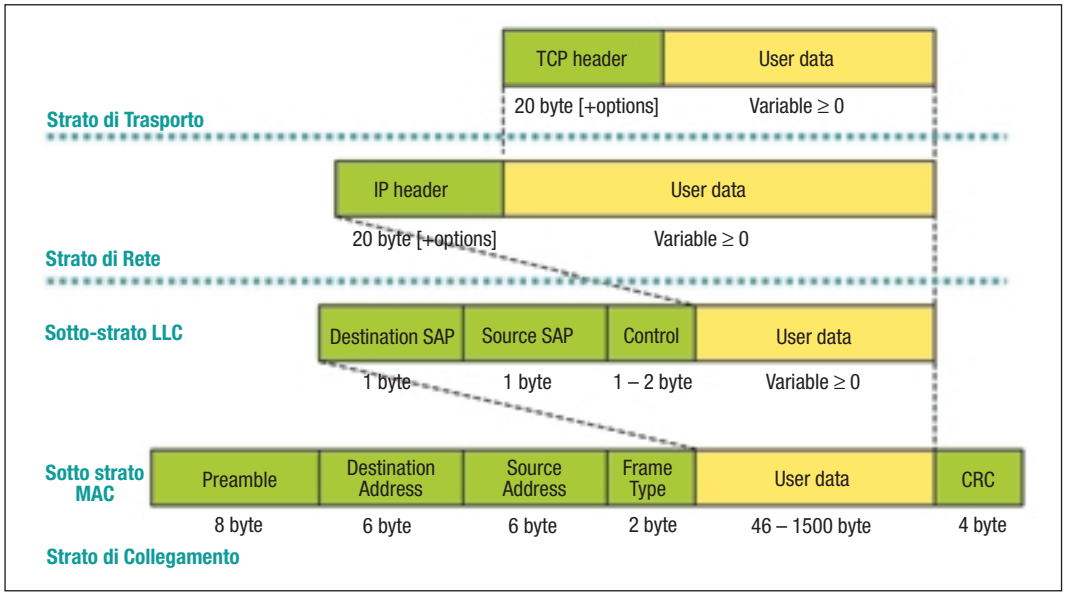


FIGURA 4

Esempio di relazione tra PDU: dall'alto verso il basso sono mostrati un segmento TCP, un pacchetto IP, una trama LLC e una trama Ethernet

tra quelle fornite dal modello OSI. L'approccio a scatole cinesi, indipendenti nella definizione delle procedure interne, è l'elemento abilitante per uno sviluppo modulare e interoperabile di sistemi complessi, che vanno spesso soggetti nel corso della loro vita ad aggiornamenti tecnologici.

Un esempio di annidamenti di PDU e aggiunta di informazioni di controllo da parte di entità di strati successivi è mostrato nella figura 4. Per ogni strato la relativa PDU è esplicitamente mostrata con la sua informazione di controllo (in verde) e la parte che ospita i dati dello strato superiore (in giallo). Ogni strato inserisce nella propria PCI quello che serve alle funzioni del relativo protocollo di strato (indirizzi, campi di versione, di lunghezza e qualificazione delle parti che seguono, numerazione per il controllo di sequenza, riscontri, flag per gestire la consegna dei dati, la loro eventuale frammentazione, campi di controllo di errore ecc.).

Una PDU può ridursi alla sola PCI: con riferimento all'esempio in figura 4, i segmenti TCP usati dal destinatario dei dati come riscontro sono tipicamente (ma non necessariamente) ridotti alla sola intestazione di 20 byte. In generale, un protocollo si svolge attraverso lo scambio di PDU tra le entità dei sistemi coinvolti. Le PDU assolvono compiti di controllo (per esempio quelli richieste nel caso di instaurazione di una connessione preliminarmente all'invio dei dati), di gestione (per esempio PDU per verificare la presenza dell'entità remota o per effettuare misure), di trasporto dei dati di utente. Nei protocolli moderni le PDU sono stringhe binarie, strutturate in "campi", cioè composte di sequenze di bit che sono interpretate in funzione della posizione che esse occupano nell'ambito della PDU, del proprio contenuto binario e di altri campi della PDU. È fondamentale rammentare che i bit sono bit e la rappresentazione di una PDU come composta da campi con i propri nomi in bell'ordine, come nella figura 4, è solo nella mente di chi concepisce il protocollo e non è "visibile" al codice o all'hardware che lo esegue. È quindi fondamentale garantire la possibilità di *delimitare* le PDU e quindi di scomporle in modo univoco nei campi costituenti, comprese le opzioni e le varianti possibili (questa funzione si chiama *parsing*).

Il formato delle PDU, sia esso specificato mediante rappresentazione dei campi componenti come nella figura 4 oppure con metodi più sofisticati quali ASN.1 (come accade in molti esempi di protocolli soprattutto di livello applicativo), permette di descrivere gli elementi con i quali realizzare il servizio offerto dal protocollo.

Un protocollo di strato *N* offre servizio alle (*N* + 1)-entità; esse possono stimolare le entità del livello servente (lo strato *N*) mediante "primitive" di servizio (Figura 5): per esempio, il trasferimento di una PDU di dati di utente avviene emettendo una primitiva di *Data-Request*, nella quale si passano come parametri quella che sarà la (*N*)-SDU e parametri aggiuntivi (parametri di QoS, indirizzi di destinazione, specifiche sulle modalità di trattamento dei dati da parte dell'(*N*)-protocollo, per esempio richiesta di cifratura). Il protocollo dello strato *N* effettua il trasferimento dei dati con tutti i meccanismi che gli sono propri (per esempio quelli di rivelazione e recupero degli errori), e presenta infine una primitiva di Data-Indication alla (*N* + 1)-entità destinataria.
In concreto le primitive possono essere segnali su circuiti interni a una macchina, interrupt (software o hardware), chiamate a funzioni o procedure nell'esecuzione di un programma, sia del sistema operativo sia di un processo di area utente, a seconda del livello architetturale degli strati cui le primitive si riferiscono. Questa enorme varietà di modalità realizzative viene ricondotta a concetti semplici e unitari, di grande valore nel concepire, definire e sviluppare in concreto i protocolli.
La razionalizzazione del funzionamento di un sistema inserito in una rete di comunicazione si è dimostrata feconda. Non c'è standard di sistemi di comunicazione che non descriva le funzioni svolte strato per strato e quindi le primitive per quanto riguarda le interfacce "verticali" e le procedure protocollari per quelle "orizzontali", cioè tra entità alla pari in sistemi remoti.
Altri due concetti fondamentali e ricorrenti nei sistemi di telecomunicazione sono la *multiplazione* (con la corrispondente funzione inversa di *demultiplazione*) e la *connessione*.
In termini OSI un protocollo di strato N effettua multiplazione quando si pone al servizio di una molteplicità di (*N* + 1)-entità (gli utenti del servizio erogato dal (*N*)-protocollo) e trasporta quindi con le proprie (*N*)-PDU flussi di dati appartenenti a diverse relazioni tra entità di strato *N* + 1. Un esempio è il protocollo IP che trasporta segmenti appartenenti a diverse connessioni di trasporto TCP, datagrammi di diversi flussi UDP e unità di dati di molti altri protocolli. Un aspetto cruciale, immediatamente ovvio dal modello astratto della multiplazione è che deve essere inserita nella (*N*)-PCI informazione sufficiente a identificare a quale (*N* + 1)-entità è destinata una data (*N*)-SDU. Nell'esempio del pacchetto IP, l'informazione in questione è l'indirizzo IP dell'interfaccia di destinazione del pacchetto e il campo Protocol Type che specifica a quale modulo software del destinatario va consegnato il contenuto di dati del pacchetto IP.
La connessione è un'associazione logica tra due o più entità remote che permette loro di istanziare e mantenere aggiornati coerentemente i rispettivi stati di evoluzione nell'esecuzione di un protocollo di strato. Il concetto di connessione accoppiato con quello di architettura stratificata si rivela molto potente: esso permette di evidenziare in modo chiaro e di razionalizzare le interdipendenze tra sistemi e tra diversi compiti svolti nel processo di comunicazione. Interagire con o senza connessione è lo spartiacque tra la possibilità di supportare qualità di servizio negoziata, affidabilità del trasferimento dei dati, e

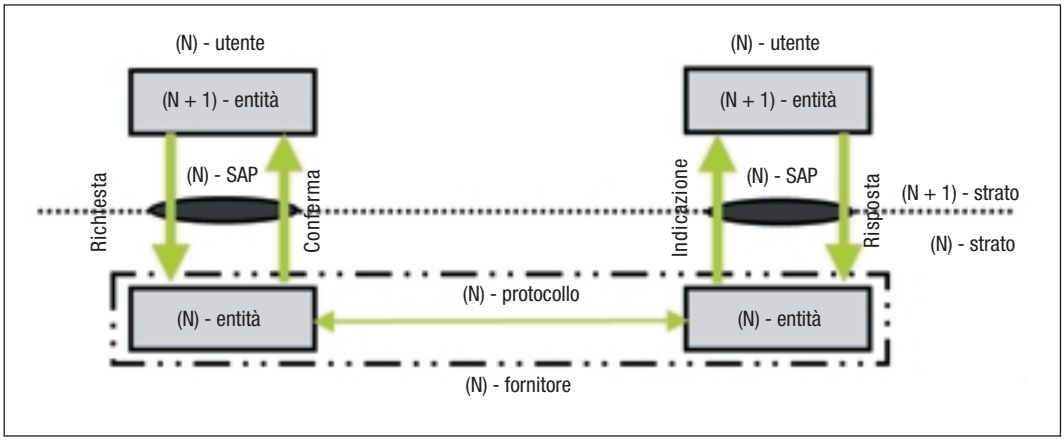


FIGURA 5
Interazioni tra strati adiacenti per la fornitura di un elemento di servizio secondo il modello OSI

molti altri arricchimenti del servizio offerto da un protocollo e la semplicità di realizzazione e quindi la parsimonia di risorse di memoria e calcolo richieste. Per mantenere in sequenza completa i diversi frammenti di dati trasferiti nell'ambito di un'interazione tra entità remote, controllare il flusso o garantire un fissato grado di servizio (ritardo per esempio) sono necessari contatori e meccanismi di recupero dei dati mancanti. Questo è possibile instaurando preventivamente una connessione (quindi inizializzando i contatori in modo coerente). D'altra parte questo richiede buffer e esecuzione di molte istruzioni in entrambi i "capi" della connessione. Inoltre rende complicato adattarsi a variazioni del servizio offerto dagli strati sottostanti.
È importante osservare che, coerentemente con il principio dell'indipendenza tra strati, la modalità di funzionamento con o senza connessione di un (*N*)-protocollo non influenza quelle dei protocolli degli strati adiacenti. Prendiamo il colloquio tra un PC di utente privato e il web server del suo ISP. Lo strato fisico tra PC e modem ADSL è tipicamente realizzato con interfacce USB (con connessione) ovvero Ethernet (senza connessione). Al di sopra si usa il PPP (con connessione) o il MAC Ethernet (senza connessione). Tra modem ADSL e NAS si usa una connessione ATM (strato due). Per lo strato di rete, il terzo, si usa IP tra PC e web server (senza connessione), al di sopra si usa TCP (con connessione) e al di sopra ancora HTTP (senza connessione).
Il vantaggio della modalità a connessione è la potenziale ricchezza di funzioni eseguibili solo grazie all'associazione logica stabilita tra le parti con la connessione; il vantaggio della modalità senza connessione è la semplicità delle entità che eseguono il protocollo[1]. Non esiste una soluzione migliore dell'altra. In ogni sistema e strato l'una o l'altra sono le più indicate, dato il contesto. Molti protocolli di strato due sono realizzati con la possibilità di optare per l'una o l'altra modalità, in funzione degli scopi e delle risorse che le entità coinvolte possono dedicare all'interazione.

[1] Una diffusa realizzazione (4.4BSD-Lite distribution) di TCP e IP (il primo con connessione, il secondo senza) consta di circa 15000 e 2000 righe di codice rispettivamente.

Importanti valori aggiunti dai protocolli di comunicazione riguardano aspetti trasversali dal punto di vista architetturale, nel senso che il loro trattamento non può essere confinato in generale ad un ben determinato strato. Se ne citano qui tre: la *Qualità di Servizio* (QoS, *Quality of Service*), la sicurezza (nel senso del termine anglosassone *security*, da non confondere con *safety*), il risparmio energetico.
La *QoS* si concreta in metriche di prestazione specifiche dello strato nella quale si considera: può riferisce al rapporto segnale-rumore o *BER* (*Bit Error Ratio*) di un collegamento trasmissivo nello strato fisico, alla portata media e al ritardo di una connessione nello strato di trasporto, alla flessibilità d'uso del protocollo da parte dell'utente nello strato applicativo. La "portata" della *QoS* è legata allo strato al quale essa è misurata: si tratta ovviamente di *QoS* da estremo a estremo negli strati alti, di *QoS* locale per gli strati bassi.
Il progetto di un protocollo è sempre la risposta a un'esigenza di *QoS*: gli algoritmi che ne sono alla base, eseguiti dalle entità coinvolte nel protocollo hanno sempre obiettivi prestazionali che mirano ad aggiungere funzionalità al trasferimento di segnali permesso da mezzi fisici di comunicazione allo scopo finale di supportare la cooperazione di processi applicativi più o meno sofisticati.
Gli aspetti di sicurezza sono anch'essi trasversali nel senso che possono essere introdotti in protocolli di tutti gli strati: si pensi alla cifratura a livello fisico, alla realizzazione di tunnel sicuri al livello rete, ai processi di autenticazione al livello di collegamento o applicativo tipici ormai della maggior parte delle nostre interazioni in rete (dall'accesso alle reti cellulari o WiFi, alle connessioni PPP via modem, all'accesso a porzioni riservate di siti web o serventi di posta elettronica, ad applicazioni di commercio elettronico). Basti qui osservare che l'introduzione di funzioni di sicurezza (confidenzialità, autenticazione, integrità, non-ripudio, disponibilità) può tradursi in aggiunta di procedure a protocolli esistenti, magari attraverso campi opzionali o nuove PDU definite nel protocollo, oppure può tradursi nella definizione di veri e propri nuovi protocolli, il cui precipuo obiettivo di arricchimento di valore è offrire servizi di sicurezza nell'am-

bito dell'interazione tra utenti e/o processi.
Infine, considerazioni di dissipazione di energia nella comunicazione diventano importanti quando almeno alcuni dei sistemi partecipanti alle interazioni protocollari sono alimentati a batteria, come nelle reti cellulari e wireless (WLAN, WPAN, reti di sensori). È intuitivo che i protocolli di livello fisico siano interessati nella gestione dell'energia (si veda per esempio la funzione di controllo di potenza), ma non meno di questi sono coinvolti nel risparmio energetico anche i protocolli degli strati di collegamento, rete e trasporto. Due esempi per chiarire. Il recupero di errori mediante ritrasmissione ha un costo energetico; l'urgenza e la stessa opportunità di ritrasmettere una PDU di dati va quindi bilanciata con il costo energetico che questo comporta. Per sistemi che prevedano un accesso multiplo coordinato da un apposito protocollo (si chiamano protocolli MAC, *Medium Access Control*), come la maggior parte dei sistemi di accesso wireless, le procedure di accesso prevedono spesso la funzione di ascolto del mezzo trasmissivo per evitare o almeno limitare le collisioni; una scheda wireless consuma una quantità di energia quasi uguale sia quando è in trasmissione sia quando è in ascolto[2]. È chiaro come anche l'aspetto energetico deve essere valutato nella definizione di protocolli che debbano funzionare in sistemi alimentati a batteria.

Come rinunciare alla connessione e mantenere uno stato: HTTP e i cookies

Nel caso di un server applicativo che deve contemporaneamente trattare numerose richieste di servizio da parte di una popolazione di client può essere molto oneroso inizializzare e far evolvere uno stato per ogni singola comunicazione, Queste considerazioni hanno per esempio portato a definire il protocollo alla base del World Wide Web , l'HTTP, prevedendo una modalità di comunicazione senza connessione (si dice che HTTP è *stateless*). Ogni interazione di un client con un server web si riduce ad uno schema query-response: il client chiede un contenuto, il server lo rintraccia, lo invia e ...si può dimenticare del client!
Per estendere le capacità dei server web, ferma restando la natura stateless di HTTP, sono stati introdotti i cosiddetti *cookies*. Un server impacchetta lo stato riguardante l'interazione con il client che lo ha contattato e lo invia al programma client, il quale lega il *cookie* all'URL del sito che glielo ha inviato. Alla successiva interazione indirizzata a quell'URL da parte di quel client, il cookie viene inviato al server, così "rammentandogli" lo stato della precedente interazione. Un bell'esempio di come mantenere la semplicità del paradigma di comunicazione connectionless e ottenere comunque una "memoria", magari con qualche rischio per la sicurezza.

4. UN CONFRONTO TRA OSI E TCP/IP

Le architetture a strati del modello OSI e di Internet sono rappresentate nella figura 6. Esistono infiniti commenti su tema del confronto tra le due; quasi tutti si arrampicano sui vetri cercando impossibili paragoni tra protocolli, strati e quant'altro.
Vale qui sottolineare l'unica vera differenza, concettuale e non strutturale. OSI nasce come un modello di riferimento, TCP/IP nasce invece come pila di protocolli. OSI è il risultato di uno sforzo di modellizzazione per creare regole di comportamento nell'interconnessione tra sistemi aperti. Proprio per questo OSI è più importante per quello che *non* dice rispetto a quello che dice (OSI infatti non dice nulla circa la dimensione dei sistemi, la loro dislocazione geografica, la qualità della rete di trasporto, sul numero di macchine in cui le varie funzioni vengono realizzate ecc.). Nell'OSI la definizione dei protocolli è stata aggiunta in un secondo tempo e rappresenta la componente meno significativa: i protocolli OSI sono morti prima di nascere, uccisi dalla loro generalità e universalità (che fa rima con ambiguità e complessità). TCP/IP è invece il risultato di uno sforzo per realizzare un sistema di scambio dati semplice, robusto, di facile e sicura interpretazione, implementabile in modo *quick and dirty*. Del resto il motto dell'IETF, l'ente che presiede allo sviluppo delle norme di riferimento per Internet, è *We believe in rough consensus and running code*.
Un altro punto importante è tenere presente che la razionalizzazione e astrazione definita con il modello OSI è storicamente posteriore alla concezione delle reti che oggi conosciamo come Internet; inoltre il punto di vista preso da coloro che hanno sviluppato quest'ultimo paradigma è assumere per dato di fatto l'eterogeneità tecnologica delle reti locali e geografiche usate per interconnettere i sistemi di elaborazione (le co-

[2] Nel caso di schede WiFi l'ascolto (*carrier sensing*) costa circa l'80% del consumo energetico richiesto in trasmissione.

siddette sotto-reti: reti SNA, DecNet, Appletalk, LAN Ethernet, reti wireless) e puntare a definire un livello di inter-rete per realizzare l'interconnessione globale (il collante universale è IP) e fornire il supporto per l'esecuzione della più ampia varietà di processi applicativi. Illustra efficacemente questa concezione dell'architettura Internet il cosiddetto modello a clessidra, dovuto a Henning Schulzrinne (Figura 7).

Nell'architettura Internet si distingue quindi un livello di sotto-rete, che comprende gli strati bassi dell'architettura: può ridursi agli strati fisico e di collegamento (come nel caso delle LAN e dell'ATM) può essere incredibilmente complesso (si veda il caso delle reti dorsali del GPRS).

Sopra questo macro-livello basso si pone il livello unificante di inter-rete, la cui funzione specifica è indirizzamento, multiplazione e commutazione. Queste funzioni sono svolte da un unico protocollo, l'*Internet Protocol* (IP), corredato di alcuni altri protocolli ausiliari (ICMP, ARP, RARP).

Il primo strato con significato da estremo a estremo è quello sopra al livello IP, corrispondente grosso modo al livello di trasporto dell'architettura OSI, così come il livello IP corrisponde essenzialmente allo strato di rete. I due principali protocolli di questo livello sono il *Transmission Control Protocol* (TCP) e lo *User Datagram Protocol* (UDP).

Infine, al di sopra del livello di trasporto si colloca il livello applicativo, corrispondente praticamente agli strati di utilizzazione del modello OSI, cioè quelli di sessione, presentazione e applicativo vero e proprio.

Da questa pur sommaria descrizione si nota il blando grado di dettaglio con il quale nel modello Internet si descrivono i livelli bassi dell'architettura, rispetto all'attenzione in essi posta dal modello OSI e la prolissa e industrialmente poco prolifica visione del modello OSI relativamente agli strati alti, compattati dal modello Internet in un generico livello applicativo, la cui effettiva realizzazione è tutta demandata al software applicativo residente sui sistemi terminali e in eventuali sistemi intermedi (*proxy, media gateway* ecc.). È chiaro che i due modelli rispondono a esigenze e visioni maturate in mondi diversi, almeno nel momento in cui i

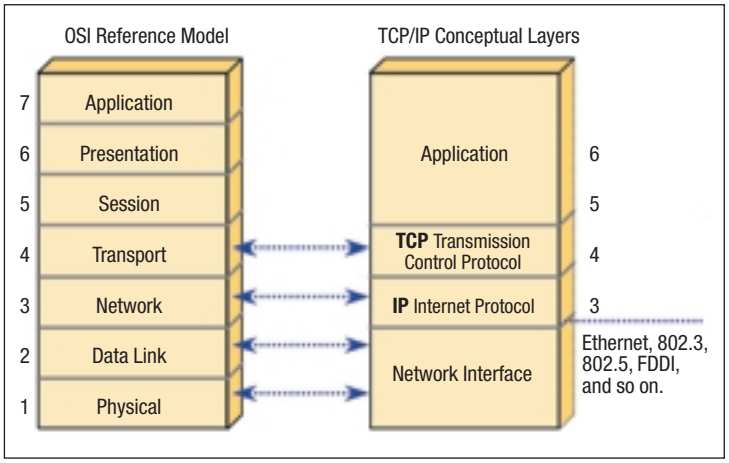


FIGURA 6
Pile protocollari OSI e Internet a confronto

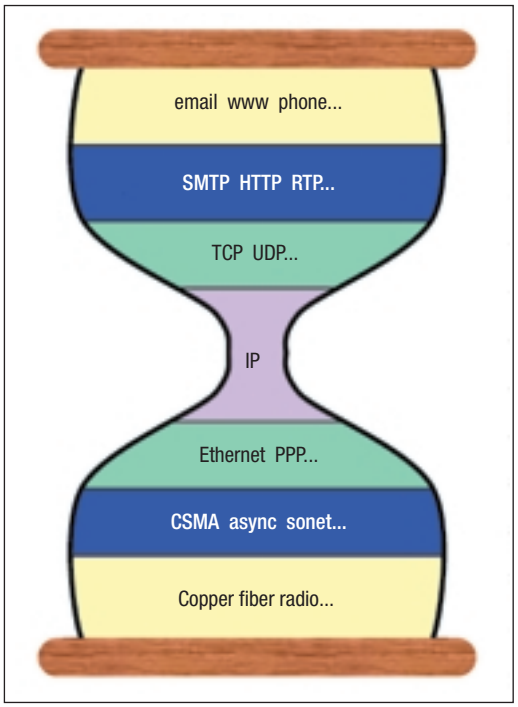


FIGURA 7
Modello a clessidra dell'architettura di Internet

modelli si affermarono (anni '70 e primissimi anni '80): l'industria dei computer e le comunicazioni di dati per il modello Internet; il mondo delle telecomunicazioni classiche, telefoniche prima di tutto, per il caso del modello OSI. Traccia di questo si ritrova, tra gli altri aspetti, nella matrice dei principali Enti di normativa promotori dello sviluppo di nuovi protocolli: ITU-T, ETSI, 3GPP per le reti cellulari, le reti di trasporto e in generale tutte quelle con un forte contenuto di tecnologia di telecomunicazione (vicine ai livelli più bassi dell'architettura: fisico,

di collegamento); IETF, IEEE vari Fora e consorzi industriali per quanto riguarda Internet, le reti wireless, le reti in area locale, in generale quelle reti miranti in primo luogo all'interconnessione dei computer (vicine per lo più ai livelli dalla rete in su, quasi esclusivamente realizzate in software, al più firmware per il caso delle reti wireless e delle schede per LAN).
La storia recente ha visto il crescente successo del modello Internet. Questo spostamento di paradigma avvenuto a partire dalla metà degli anni '90, manifestato anche nella terminologia (si parla oggi di ICT, non di telecomunicazioni), tra le molte conseguenze ha avuto anche un significativo impatto sul modo di concepire, ideare, sviluppare e gestire i protocolli. Da un approccio "telecom" connotato da tecnologie ad hoc, spesso dipendenti da specifici sviluppi hardware, comunque con piattaforme di sviluppo che solo grossi laboratori aziendali specializzati potevano affrontare (esempio classico: ISDN), si è passati con il prevalere del modello Internet a protocolli realizzati soprattutto in software, sia nel *kernel* dei sistemi operativi sia spesso con programmi che girano in *user space*; sempre più spesso i protocolli sono materialmente realizzati usando codice sorgente in C, java o altri linguaggi di livello relativamente alto; sempre più spesso c'è un intreccio inestricabile tra il software dei computer e quello dei protocolli rivolti alla soluzione di problemi di comunicazione e spesso il tutto è fornito come codice open source, soprattutto con il progressivo affermarsi di Unix nei suoi vari dialetti (vedi Linux sui PC commerciali); anche i protocolli dei livelli di collegamento del MAC e persino molte funzioni tradizionalmente appartenenti al livello fisico, sono oggi realizzati utilizzando DSP o FPGA, con uno sforzo di sviluppo focalizzato sulla programmazione a livello relativamente alto, grazie agli strumenti di CAD e compilazione che fungono da intermediari tra il codice sorgente prodotto dallo sviluppatore e quello che deve essere caricato nell'hardware. Queste tendenze hanno abbassato drammaticamente la soglia economica per lo sviluppo, la sperimentazione, la valutazione e misura di protocolli di comunicazione, nello stesso tempo fondendo le competenze necessarie con quelle proprie della computer industry. Se si vuole anche essere polemici, queste tendenze hanno anche ingenerato qualche confusione e concesso spazio a tecnici di scarsa professionalità.

5. CROSS-LAYERING OVVERO IL MODELLO OSI ULTIMA PAROLA?

Le architetture stratificate poggiano su un assioma: le funzioni attivate per la cooperazione di processi applicativi attraverso una rete di comunicazione sono raggruppate in insiemi omogenei (gli strati) e i protocolli che realizzano queste funzioni sono definiti all'interno di ciascuno strato *indipendentemente* dagli altri se non per i vincoli imposti dalle interfacce tra strati adiacenti (le specifiche delle primitive di servizio tra strato utente e strato fornitore); in altri termini, il servizio che le entità di uno strato si aspettano da quelle dello strato inferiore è specificato a livello dell'interfaccia tra i due strati (il Service Access Point tra lo strato $N+1$ e N, (N)-SAP, vedi Figura 5), ma non è specificato il *modo* in cui le entità dello strato N devono poi realizzare quel servizio (i protocolli dello strato N).
È un tipo di approccio che fa tesoro della massima "divide et impera", ed è un classico esempio di decomposizione di quello che può essere concepito come un grande problema di ottimizzazione vincolata in sottoproblemi collegati gerarchicamente, che possono essere tuttavia risolti separatamente, raggiungendo tipicamente un sub-ottimo come soluzione globale (ma alla soluzione si arriva!). Questo è l'approccio che ha reso possibile la definizione delle reti cellulari, delle reti di trasporto (SDH, ottiche), della stessa rete Internet.
Esistono alcuni contesti dove l'ottimizzazione perseguibile derogando alla rigida regola di separazione dei protocolli di strati diversi è abbastanza pagante da superare i vantaggi di flessibilità e potenziale semplicità offerti dalla decomposizione in strati internamente indipendenti. In questi casi, gli algoritmi che sono svolti dalle entità che partecipano al protocollo utilizzano informazioni (eventi, misure, dati) appartenenti anche ad altri strati per ottimizzare le prestazioni ottenibili: si parla di

protocolli *cross-layer*. È questo un approccio che ha riscontrato una fortuna crescente negli ultimi anni in almeno due contesti molto significativi: le reti wireless e il controllo della congestione nelle reti a pacchetto.
Nel primo caso, quando si tratta di servire una popolazione di utenti che condividono una risorsa radio, esiste una relazione stretta tra gli effetti di scelte effettuate nei protocolli di livello fisico, MAC, di collegamento e di rete. Per esempio, si pensi alle operazioni di scheduling dei pacchetti effettuati da una stazione base, utilizzando come metriche guida la QoS differenziata da offrire ai diversi flussi di dati oppure criteri di *fairness*; si tratta di una funzione collocata tipicamente nello strato di collegamento o di rete. È chiaro che questa funzione ha un nesso molto stretto con la gestione delle risorse radio in termini di quali codici, potenze, frequenze, time slot usare in trasmissione, la codifica e modulazione dei dati, tutte decisioni proprie dei protocolli di strato fisico. Molti lavori sia teorici sia sperimentali hanno mostrato gli ampi margini di guadagno possibili se si formula un problema congiunto (cross layer appunto) che tenga congiuntamente conto delle caratteristiche del *traffico* e del *canale radio* [7, 8].
In alcuni contesti, come le reti di sensori, il *cross-layering* porta a definire problemi che coinvolgono congiuntamente anche più di due strati. I parametri da usare per trasmettere i dati (potenza, modulazione, codifica), quando trasmettere tenendo conto di obiettivi di portata (*throughput*) equità di accesso e contenimento dell'interferenza o al limite delle collisioni, l'instradamento da seguire secondo un paradigma di tipo multihop (raggiungere la destinazione con un unico passo, consumando maggiore potenza e provocando più interferenza oppure effettuare più passi, aggiungendo quindi ritardo, consumo energetico e risorse di calcolo e memoria dei nodi intermedi?) sono tutti aspetti che devono essere considerati assieme se si vuole raggiungere un'efficienza ottima complessiva del sistema in termini di prestazioni e costo. Nel caso delle reti di sensori, il problema è abbastanza delimitato e le condizioni operative sufficientemente "estreme" da motivare un approccio olistico, di tipo *cross-layer*, anziché lo sviluppo di protocolli indipendenti nel senso delle architetture stratificate.
Altro esempio di *cross-layering* è dato dalle funzioni di controllo di congestione, svolta nei protocolli di trasporto da estremo a estremo (per esempio mediante controlli a finestra variabile e adattiva, come nel TCP) e le politiche di scheduling e gestione delle code interne ai nodi della rete, le cosiddette politiche di *Active Queue Management*, AQM. Queste ultime sono funzioni ovviamente appartenenti a protocolli dello strato di rete, ma il loro effetto sulle decisioni e le prestazioni ottenibili dai protocolli di trasporto sono tali che un approccio di ottimizzazione congiunta *cross layer* è posto all'attenzione della comunità di ricercatori e sviluppatori che lavorano su questi problemi [9].
Come tutti i modelli realmente utili, anche il modello OSI e gli analoghi modelli alternativi introducono astrazioni e concetti di base, fondamentali per uno sviluppo ordinato ed efficace di sistemi complessi, ma vanno capiti nelle loro motivazioni e usati con capacità di adattamento ai contesti tecnologici e applicativi specifici, senza dogmatismo. Del resto anche l'indipendenza funzionale e quindi realizzativa dei protocolli di strati distinti, infranta dall'ottimizzazione *cross-layer*, ha dimostrato di conseguire sì sub-ottimi, ma di potersi avvalere con estrema tempestività delle diverse velocità di evoluzione delle tecnologie base dei diversi strati: si veda l'esempio del Wi-Fi, nel quale il livello MAC è rimasto invariato nella sostanza dal 1997 a oggi[3], mentre sono state messe a punto quattro principali versioni del livello fisico (IEEE 802.11 base e successivamente IEEE 802.11b/a/g).

6. L'EVOLUZIONE DELLO STRATO APPLICATIVO

L'evoluzione del modello OSI é insita nella sua stessa struttura; OSI é assolutamente generale. La generalità del modello OSI e del suo rivoluzionario concetto di strato permette infatti di cambiare, sostituire, arricchire di funzioni uno "strato" senza alterare l'impalcatura.

[3] Solo con gli imminenti standard IEEE 802.11e e 802.11n si intende rimettere mano alla definizione del protocollo MAC.

Le nuovi reti wireless (WiFi, WiMax) possono rimpiazzare le tecnologie esistenti dando migliori o differenti servizi di connettività ai DTE finali e lasciandone inalterate le funzionalità applicative. Se si hanno reti di trasporto a larga banda o a qualità garantita, si possono costruire sessioni multimediali in cui sono presenti voce ed immagini. I principi non cambiano.

A livello applicativo abbiamo avuto le evoluzioni più significative – non in termini concettuali, ma in termini di completezza funzionale. Lo strato applicativo ha infatti tratto enormi benefici dal Web, ambiento aperto e distribuito. Ai tempi in cui l'OSI è stato concepito, gli unici servizi applicativi ipotizzabili erano quelli associati al trasporto dei dati (*file transfer*), alla consegna dei messaggi (posta elettronica) oppure alla cooperazione transazionale per eseguire funzioni distribuite ma necessariamente sincrone (CCR, *Commitment and Concurrency Recovery*, utilizzato per transazioni a più livelli o aggiornamenti multipli di basi di dati); in sostanza erano i servizi ipotizzabili in architetture costituite essenzialmente da mainframe ed elaboratori satellite.

Il Web, ambiente naturale in cui eseguire applicazioni con presentazione e metodi di accesso standard, permette di arricchire enormemente l'insieme dei servizi applicativi disponibili. Il connubio tra Web ed Internet, lo sviluppo di infrastrutture fisiche in cui le server farm sono sempre più un'appendice naturale della rete di telecomunicazione, permettono di creare un midlleware applicativo in cui le applicazioni sono una collezione di servizi, invocati ed eseguiti sulla base delle necessità, senza nulla imporre alla loro reale dislocazione fisica.

Nasce dunque l'idea dei *Web Services*, concettualmente analoghi agli *Application Service Elements* del modello OSI, ma molto più universali.

Il *Web Service* è un *middleware* che permette di trasformare un'applicazione (un programma che meccanizza un processo di business e produce dei risultati) in un "oggetto", dotato di attributi e metodi, utilizzabile come un mattoncino del lego in un ambiente distribuito ed aperto. Per far questo e' necessario disporre di una funzione di descrizione del servizio, di una sua catalogazione e di un meccanismo per poterlo invocare, passargli degli input e ricevere dei risultati. L'insieme di un'architettura aperta ed universale come il TCP/IP e di applicazioni create con la logica dei *Web Services* rappresenta la realizzazione più significativa del modello concepito dalla ISO nei lontani anni '80. Se poi pensiamo di aggiungere a questi servizi anche un'autostrada intelligente che coordini il flusso di informazioni scambiate tra i vari moduli applicativi (con funzioni di gestione delle code, delle priorità, delle *retry* ecc.) - ossia di aggiungere ciò che è comunemente chiamato *enterprise bus* - abbiamo creato una nuova impostazione dei sistemi informativi in cui l'enfasi si sposta dalla gerarchia dell'elaborazione alla granularità delle funzioni applicative e alla loro utilizzabilità nell'ambito delle applicazioni finali, sempre più governate dalla dinamica del business (SOA, *Service Oriented Architecture*).

Scendendo un po' nei dettagli, i *Web Service* definiti sono:

- WSDL (*Web Service Definition Language*): descrive cosa fa un *Web Service* nonché la sua operatività (come può essere invocato, come gli si possono passare i parametri);
- UDDI (*Universal Discovery Description and Integration*): facilita la pubblicazione e la ricerca dei servizi disponibili. Sono disponibili funzioni di "pagine bianche" e "pagine gialle" (classificazioni per nome specifico o categoria);
- SOAP (*Service Oriented Application Protocol o Simple Object Application Protocol*): offre – sulla base dello standard XML- meccanismi standard per l'interscambio di informazioni.

Non è obiettivo di questo articolo descrivere in dettaglio la logica dei *Web Services* (Mondo Digitale ha già trattato questo argomento in modo completo ed autorevole – si veda a tale proposito "Un'introduzione ragionata al mondo dei *Web Services*", marzo 2004), ma evidenziare il loro posizionamento nell'ambito del "valore aggiunto" ai protocolli di comunicazione.

Abbiamo sostituito mainframe, calcolatori satellite, terminali con PC e applicazioni in esecuzione presso *Web Farms*. Abbiamo

cambiato il terminale 3270 con una pagina Web, abbiamo cambiato le l*ogical unit* di IBM con l'HTTP, abbiamo introdotto l'IP come busta universale per il trasporto e l'instradamento. Abbiamo insomma ottenuto ambienti più flessibili, ma viviamo ancora dei concetti introdotti dalla *Service Definition* del modello OSI. L'ISO 7498 ha ancora molto da insegnare a tutti, vent'anni dopo.

Bibliografia

[1] Tanenbaum A.S.: *Computer Networks*. 4-th edition, Prentice Hall, 2003.

[2] IBM, Systems Network Architecture, 1978.

[3] Listanti M., et alii: *Telematica*. Capitolo del testo "Manuale di Informatica", Calderini, 1986, p. 781-902.

[4] Elementi di procedura, descrizione della trama HDLC e della modalita' "balanced" e "normal".

[5] Kleinrock L.: *Queueing Systems*. Computer applications, Vol. II, Wiley, 1976.

[6] Modello OSI, Service Definition e Protocol Specification.

[7] Song G., Li Y.: Cross-Layer Optimization for OFDM Wireless Networks - Part I: Theoretical Framework. *IEEE Transactions on Wireless Communications*, Vol. 4, n. 2, March 2005, p. 614- 624.

[8] Lau V.K.N., Kwok Y.-K.R.: *Channel adaptive technologies and cross layer designs for wireless systems with multiple antennas*. J. Wiley, 2006.

[9] Wang Jiantao, Li Lun, Low Steven H., Doyle John C.: Cross-Layer Optimization in TCP/IP networks. *IEEE/ACM Trans. on Networking*, Vol. 13, n. 3, Giugno 2005, p. 582-568.

GIACOMO ZANOTTI laureato in Ingegneria elettronica al Politecnico di Milano nel 1973, ha seguito dal nascere il tema dell'Information Communication Technology. Ha ricoperto incarichi di direttore marketing e di responsabile di Business Unit di system integration in primarie aziende di informatica e telecomunicazioni. Attualmente collabora in ITnet (Internet Service Provider del Gruppo Wind) alla definizione delle strategie di sviluppo business e allo sviluppo dei canali di vendita. Ha interesse all'ambiente accademico e tecnico-scientifico, che sviluppa come membro del consiglio AICA e con attività di docenza in Master organizzati da Università (Milano, Padova). Ha ricoperto per quattro anni il ruolo di membro italiano nella commissione Europea IST-prize.
E-mail : giacomo.zanotti@gmail.com

ANDREA BAIOCCHI si è laureato in Ingegneria Elettronica nel 1987 e ha conseguito il Dottorato di Ricerca in "ingegneria dell'Informazione e della Comunicazione" nel 1992 presso l'Università di Roma "La Sapienza". Dal gennaio 2005 è Professore Straordinario nel settore Telecomunicazioni presso la Facoltà di Ingegneria dell'Università "La Sapienza".
I principali contributi scientifici di Andrea Baiocchi sono sui modelli e algoritmi per il controllo del traffico in reti ATM e TCP/IP, sulla teoria delle code, sulla gestione delle risorse radio in reti cellulari. I suoi attuali interessi di ricerca sono concentrati sui modelli per l'analisi e il dimensionamento di reti a pacchetto con controllo della congestione da estremo a estremo, secondo il paradigma del TCP, e sul "mobile computing," in particolare adattamento del TCP al canale wireless e sulle gestione cross-layer delle risorse nell'interfaccia radio. Queste attività sono state e sono tutt'ora scolte nell'ambito di progetti sia nazionali (CNR, MIUR) sia internazionali (UE, ESA), ricoprendo anche posizioni di responsabile di unità di ricerca. Andrea Baiocchi ha al suo attivo oltre ottanta pubblicazioni su riviste scientifiche e su atti di conferenze internazionali, ha preso parte alle Commissioni Tecniche di Programma di sedici convegni internazionali, tutti nel settore delle reti di tlc; ha anche fatto parte della redazione del Notiziario Tecnico di Telecom Italia per dieci anni.
E-mail: andrea.baiocchi@uniroma1.it