Andrea Giachetti

Tecnologie informatiche e multimediali

per la comunicazione e la didattica

QuiEdit

Seconda edizione aggiornata

Indice

Introduzione

L'utilizzo del calcolatore e delle reti telematiche per codificare, trasmettere, rappresentare ed elaborare informazioni di tipo eterogeneo ha sicuramente rivoluzionato diversi aspetti della nostra vita sociale, culturale e lavorativa ed il suo effetto non può essere ignorato da coloro che operano nel mondo della comunicazione, dell'educazione, e di ogni disciplina legata al trattamento dell'informazione.

Internet è, infatti, diventato oggigiorno un mezzo fondamentale per l'accesso all'informazione ed alla cultura e consente innumerevoli modalità di comunicazione, apprendimento, divertimento e fruizione dei più svariati contenuti *multimediali*, cioè che combinano diversi mezzi, sia in senso fisico (suoni, immagini, video) che stilistico (narrazione, documento, ecc.).

Inoltre, computer e terminali in grado di riprodurre e visualizzare contenuti audio, video e testuali ci circondano ormai in ogni istante della nostra vita. I moderni telefoni, iPod, televisori digitali, elettrodomestici, cruscotti di automobili e via dicendo, contengono tutti al loro interno un elaboratore digitale non dissimile dal nostro PC da scrivania.

Nonostante tutto questo e nonostante la notevole eco che hanno sulla stampa gli argomenti legati al mondo digitale e telematico, si può facilmente verificare quanto la cultura di base relativa alle cosiddette nuove tecnologie sia spesso carente, e non soltanto tra coloro che non hanno accesso ad esse.

Basta, infatti, provare a chiedere, anche a giovani che utilizzano quotidianamente PC e Internet, cosa significhino alcuni termini dell'informatica che continuamente utilizziamo e leggiamo sui giornali (ad esempio “multimediale”, “client-server” o “byte”), per scoprire che il loro significato non è semplicemente ignoto, ma che la stragrande maggioranza delle persone ha conoscenze *errate* su tali concetti. Per fare un esempio, su un campione di iscritti al primo anno di un corso universitario, due studenti su tre hanno mostrato idee sbagliate sul significato di “multimediale” ed una percentuale ancora maggiore non è riuscita a classificare un browser web come un applicazione client, nonostante lo usi quotidianamente. Nessuno degli studenti del campione è stato in grado di dire a quanti bit corrisponda un Kilobyte (8192), scegliendo tra tre alternative.

La mancanza di diffusione delle conoscenze di base riguardanti le nuove tecnologie è un male sia nel caso si sia entusiasti fautori dell'utilizzo di esse, sia, come a volte accade nel mondo umanistico, le si veda con occhio molto critico, spaventati dai rischi reali o presunti cui queste ci sottoporrebbero. E' impossibile, infatti, sostenere ragionevolmente critiche a ciò che non si conosce, così come è impossibile utilizzare efficacemente a proprio vantaggio strumenti che non si padroneggiano a sufficienza.

Il fatto, poi, che le carenze si riscontrino anche nei giovani e in chi PC e reti li usa quotidianamente, significa che non c'è da preoccuparsi solo del cosiddetto “**digital divide**” cioè del fatto che una percentuale non trascurabile della popolazione del nostro paese non abbia accesso alle tecnologie, ma anche, e forse maggiormente, di una notevole carenza di cultura tecnico-scientifico diffusa tra coloro che le tecnologie le usano e che, quindi, *non ne fanno probabilmente buon uso*. Questo smentisce anche alcuni luoghi comuni sulla presunta

dimestichezza naturale dei giovani "**nativi digitali**" (nati cioè nell'era della grande diffusione dell'informatica) con calcolatori e programmi.

Una delle cause di questa insufficiente preparazione risiede probabilmente nel taglio eccessivamente pratico dei corsi di informatica che vengono impartiti ai non addetti ai lavori, che spesso si limitano alla manualistica ed ai trucchi per l'uso di programmi. A rigor di logica tali corsi non dovrebbero neppure essere chiamati corsi di informatica: dare un simile nome ad un corso che insegna ad usare Word o Internet Explorer equivale in fondo a chiamare corso di ingegneria meccanica quello che insegna a guidare l'automobile.

Un approccio più utile potrebbe essere quello di fornire la **cultura tecnica di base** che abiliti lo studente all'uso consapevole delle applicazioni informatiche e telematiche che possono essergli utili per le proprie attività. Tale base rende sicuramente molto più semplice imparare poi ad usare gli strumenti "informatici" specifici utili per il proprio mestiere.

Per fornire queste basi è importante da una parte vincere le resistenze ed i rifiuti che talvolta si oppongono all'acquisizione di competenze tecniche in ambito umanistico e, dall'altra, evitare di dare per scontate alcune conoscenze che in realtà, come abbiamo visto, non lo sono affatto.

Oggi, comunque la si pensi, non è più pensabile lo svolgimento di molte attività relative al mondo dell'educazione e della comunicazione senza utilizzare numerosi strumenti digitali. Mancare di informazioni di base per utilizzarli bene significa, quindi, semplicemente scegliere di *non fare il proprio mestiere nel migliore dei modi*. E le competenze informatiche più importanti non sono, come detto, quelle relative all'uso di programmi specializzati che magari tra un anno saranno obsoleti, ma quelle, più generali che permettono anche di poter capire e selezionare i vari strumenti, valutarne i costi ed i benefici, e così via. Perfino il semplice acquisto di materiale informatico o software non è semplice se non si hanno informazioni sufficienti: spesso aziende o pubbliche amministrazioni sprecano ingenti risorse perché non sono in grado di valutare adeguatamente i prodotti offerti.

Il rifiuto della cultura digitale in quanto difficile da capire per chi non è più giovane o non ha esperienza a riguardo dev'essere anch'esso respinto con forza.

Non si chiede, infatti, che un non addetto ai lavori conosca perfettamente il funzionamento dei microprocessori o sappia programmare le applicazioni che utilizza per il proprio mestiere. Le nozioni a lui utili di cui stiamo parlando (come, ad esempio, quali tipi di computer e programmi esistano e come funzionino a grandi linee, come funzioni e come possa creare un sito web, cosa sia un'immagine digitale e così via) sono relativamente facili da acquisire e non richiedono particolari competenze matematiche o scientifiche.

In questo testo si cercheranno di fornire al lettore alcune nozioni relative alle tecnologie di Internet ed al mondo del **web** e della **multimedialità digitale**. Si parlerà del funzionamento della rete, delle tipologie di siti Internet, della creazione della **pagine ipertestuali** per il web e della creazione di contenuti multimediali come **immagini, audio e video**. Si tratta di nozioni estremamente utili a chi lavora in ambito didattico o nella comunicazione, dato che non sarà possibile, in futuro, fare a meno di confrontarsi con questi mezzi per la trasmissione della

conoscenza. Il testo si configura come un corso introduttivo per studenti di corsi umanistici, che presuppone una conoscenza di base del funzionamento dei calcolatori (rimandiamo a testi di informatica di base, citati in bibliografia, chi sentisse di avesse lacune a riguardo), ma che dovrebbe risultare abbastanza comprensibile anche a chi non ha grande dimestichezza con la materia.

L'organizzazione dei capitoli è la seguente: il primo riassumerà in modo succinto alcune nozioni basilari sull'attuale tecnologia dei computer, il secondo presenterà alcune nozioni riguardanti le tecnologie di rete ed il funzionamento di Internet con i suoi diversi servizi e le sue problematiche d'uso. I capitoli seguenti descriveranno più in dettaglio il funzionamento dei siti web, il linguaggio di marcatura che permette di codificare gli ipertesti, cioè HTML (o XHTML) e le tipologie di oggetti multimediali utilizzati sui calcolatori ed integrabili nei siti web stessi, che dobbiamo saper creare per realizzare quei sistemi di didattica avanzata (e-learning) cui è fatto cenno nell'ultimo capitolo.

1 Computer ed informazione digitale

La possibilità di creare contenuti e comunicare che abbiamo a disposizione grazie ai moderni computer ed a tutta la serie di strumenti elettronici portatili o "nascosti" negli oggetti che ci circondano (cellulari, navigatori, agende, iPod e quant'altro) ha origine nella straordinaria evoluzione che gli strumenti e le applicazioni dell'informatica hanno avuto nell'arco degli ultimi decenni. Per capire il mondo di Internet, degli smartphone, dei social network e dell'e-learning, le informazioni di base che servono sono le stesse che si studiano nell'informatica di base, ed hanno a che fare sostanzialmente con il funzionamento di un calcolatore elettronico e la codifica digitale dell'informazione. Cerchiamo, quindi, di riassumerne brevemente i principi fondamentali.

1.1 Funzionamento ed applicazioni del calcolatore

I moderni "personal computer" (PC), ma anche tutti gli elaboratori "nascosti" ormai in moltissimi oggetti di uso comune (telefoni, lettori, multimediali, navigatori, televisori, ecc.) sono calcolatori elettronici che funzionano in maniera non dissimile dai primi elaboratori grossi come armadi che risalgono agli anni cinquanta del secolo scorso e che si vedono nei vecchi film di fantascienza. Essi ricevono in ingresso (input) i **dati** e i **programmi** per elaborare gli stessi ed eseguono automaticamente le operazioni programmate memorizzando e rappresentando su qualche dispositivo di uscita (output) il **risultato** dell'**elaborazione**.

La peculiarità dei calcolatori elettronici sta nel fatto che i dati che memorizzano ed elaborano sono **digitali**, ovvero numeri o più esattamente sequenze di **cifre binarie** (0 e 1) che nel calcolatore elettronico sono codificati mediante quantità fisiche come livelli bassi o alti di tensione, carica di condensatori, magnetizzazione di aree o altro segnale codificato da due stati differenti. L'elaborazione di questi dati consiste in sequenze delle **operazioni elementari** che sono consentite dai circuiti della **CPU** (Central Processing Unit, cioè quel dispositivo che nella sua realizzazione fisica chiamiamo anche **microprocessore**): operazioni aritmetiche e logiche sulle sequenze (stringhe) di bit, spostamenti di dati in memoria.

L'architettura base di tutti i calcolatori è rimasta quella introdotta da Von Neumann nel dopoguerra e schematizzata in Figura 1: dati e programmi sono inseriti in una **memoria centrale** che altro non è che un casellario di dati binari, che vengono continuamente aggiornati dai calcoli (operazioni aritmetiche e logiche) fatti dalla **CPU** seguendo gli ordini impartiti dall'utente mediante l'inserimento nella memoria centrale di liste di operazioni da eseguire (programmi). Ovviamente i calcolatori e i sistemi che li gestiscono sono diventati nel tempo sempre più complessi: oggi possiamo avere più CPU all'interno dello stesso calcolatore o dello stesso microprocessore (nei processori cosiddetti **multicore**), diversi tipi di **bus** di comunicazione (cioè di canali attraverso i quali i dati passano dalle memorie alla CPU e ai dispositivi periferici), e sono molto cambiati i sistemi di gestione dell'input/output. Tuttavia, nei suoi principi di funzionamento, il calcolatore è sempre la

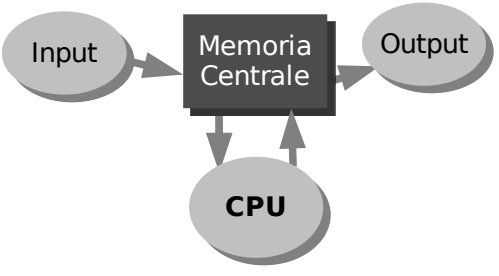


Figura 1: Architettura di Von Neumann

stessa cosa: un dispositivo capace di memorizzare ed elaborare dati digitali sulla base di sequenze di comandi preordinati. Nel corso degli anni, però, sono cresciute in modo esponenziale la potenza di calcolo delle CPU, la quantità di spazio a disposizione per memorizzare i dati e le velocità di trasferimento degli stessi all'interno ed all'esterno della macchina. Questo ha consentito di cambiare radicalmente le modalità di **interazione** con la macchina.

E' probabilmente nota a tutti la famosa "**legge di Moore**", un'ipotesi avanzata negli anni 60 da Gordon Moore, co-fondatore di Intel (nota casa produttrice di microprocessori) che prevedeva una crescita esponenziale (cioè un raddoppio ogni 18 mesi) del numero di transistor dei processori (e quindi un simile incremento delle capacità di calcolo) e che si è sostanzialmente rivelata esatta fino al nuovo millennio. Oggi acquistiamo normalmente personal computer con CPU che lavorano a **frequenze** intorno ai 2 GigaHertz, il ché significa che in un secondo vengono eseguiti in esse due miliardi di operazioni elementari sincronizzate dai cicli del **clock**, cioè dell'orologio che scandisce i tempi dell'attività della CPU stessa. Questo fa capire come un moderno calcolatore possa interagire in tempo reale con l'utente (nel tempo in cui inseriamo un dato o ci accorgiamo di un cambiamento sullo schermo, dell'ordine del decimo di secondo, la CPU ha tempo di accorgersi dell'azione dell'utente, eseguire milioni di operazioni di calcolo, cambiare la rappresentazione sullo schermo, e così via.

Un moderno PC ha, poi, una memoria RAM dell'ordine dei Gigabyte, cioè milioni di milioni di byte. Questo consente, ad esempio, di elaborare dati digitali che non rappresentano solo numeri o caratteri di testo, ma codificano anche grandezze fisiche come suoni ed immagini, consentendo le moderne applicazioni multimediali (anche grazie allo sviluppo di efficienti tecniche di **codifica** e **compressione** che permettono di rappresentare i dati multimediali utilizzando una quantità ridotta di memoria e di cui parleremo nel capitolo 5).

I calcolatori moderni sono inoltre dotati di dispositivi di memorizzazione di massa (cioè memorie che mantengono i dati permanentemente a macchina spenta) di grandissima capacità, ovvero dischi rigidi che oggi arrivano al Terabyte, cioè milioni di miliardi di byte, ma anche CD, DVD, Blu-Ray, memorie flash, che permettono di salvare permanentemente tali dati e averli accessibili all'occorrenza.

1.2 Algoritmi e programmi

L'incredibilmente grande capacità di calcolo e memorizzazione (ossia la potenza dell'hardware) non basta, però, a capire come mai i calcolatori siano diventati strumenti così versatili ed utilizzati per le più disparate applicazioni.

Per capirlo dobbiamo considerare quindi anche l'evoluzione del software, cioè dei programmi che vengono fatti eseguire ai calcolatori e dei dati che questi elaborano. I primi calcolatori venivano utilizzati per compiere operazioni ripetitive su dati perlopiù numerici oppure risolvere automaticamente problemi matematici di cui sia nota la metodologia di soluzione (ad esempio risolvere un'equazione di secondo grado dati i coefficienti o trovare il volume di un solido dati i parametri geometrici): sebbene la complicazione tecnica sia notevole, il fatto che tali compiti si possano realizzare con lo strumento elettronico è intuitivo. Basta infatti avere una rappresentazione binaria dei dati di interesse e tradurre le operazioni necessarie a completare l'elaborazione richiesta in una serie di azioni eseguibili dalla CPU. Svolgere questo compito è quello che in informatica si dice **implementare** (cioè realizzare in pratica) un **algoritmo**, creando un **programma** eseguibile. Nei primi modelli di calcolatore questo comportava un enorme lavoro di codifica, dato che era necessario scrivere direttamente il programma nel

linguaggio macchina del calcolatore, quindi utilizzando direttamente i codici operativi delle singole istruzioni (trasferimenti di caselle di memoria nella CPU, operazioni aritmetiche e logiche su tali sequenze binarie e così via). Come vedremo, l'evoluzione del software ha fortunatamente fatto sì che le operazioni di programmazione diventassero sempre più facili e questo ha ovviamente contribuito notevolmente alla diffusione ed al successo degli strumenti informatici

1.3 Software di base, i sistemi operativi

Uno dei passi fondamentali che hanno portato a trasformare il computer in uno strumento di larga diffusione è stato lo sviluppo dei **sistemi operativi**, cioè dei programmi di gestione delle risorse della macchina che permettono a programmatori ed utilizzatori di lavorare ad "alto livello" senza dover conoscere nel dettaglio la tecnologia delle macchine utilizzate.

I sistemi operativi creano le interfacce con cui l'utente riesce ad interagire con la macchina e da essi dipendono quindi la semplicità di uso e le potenzialità applicative della stessa.

I sistemi operativi si occupano quindi di fornire sistemi semplificati di gestione dell'esecuzione dei programmi, di allocare automaticamente ed in modo sicuro la memoria necessaria all'esecuzione degli stessi, di fornire sistemi di accesso e di uso di tutti i dispositivi a disposizione (dal disco rigido, con il **filesystem**, a tutti i sistemi di **input**/**output**). Con i primi sistemi operativi i programmatori ebbero vita molto semplificata: non dovevano più preoccuparsi della gestione dell'hardware e hanno poi avuto a disposizione **linguaggi** di **alto livello** per la **programmazione**, che hanno fatto sì che per realizzare applicazioni per il calcolatore non fosse più necessario conoscere i linguaggi macchina. Usando tali linguaggi (es. Fortran, C, C++) si possono scrivere programmi in codici testuali che somigliano al linguaggio naturale che vengono poi tradotti da software specifici (**compilatori** o **interpreti**) per diventare eseguibili. I sistemi operativi mettono poi a disposizione "librerie" che contengono tutte le funzioni per utilizzare in modo semplice le risorse della macchina, evitando di scrivere il codice specifico e consentendo quindi a chi scrive i programmi di operare su file, dispositivi esterni, ecc. come entità "astratte" utilizzando una sintassi predefinita. Per fare un esempio pratico, per scrivere un programma che apre una "finestra" sullo schermo, il programmatore non scrive il codice che colora i pixel dello schermo, ma "chiede" al sistema operativo di aprire un "oggetto" finestra con determinate dimensioni e caratteristiche. L'evoluzione dei sistemi operativi ha poi consentito di eseguire i programmi in maniera interattiva e sofisticata ponendo le basi per l'evoluzione dei sistemi utente moderni. Oggi quasi tutti i sistemi operativi dei vari tipi di computer permettono di fare eseguire contemporaneamente molti programmi (il cosiddetto **multitasking**) senza conflitti (cioè senza che l'esecuzione di uno influenzi quella di un altro) ed in modo **interattivo**, cioè consentendo all'utente di intervenire nell'esecuzione interrompendola e modificandone le condizioni. Questo avviene alternando nella CPU i differenti programmi in esecuzione (**processi**) per brevissimi intervalli di tempo in modo che tutti possano essere eseguiti apparentemente nello stesso momento e controllando le richieste di interruzione create dai dispositivi di ingresso. Queste caratteristiche e lo sviluppo di sempre più sofisticate **interfacce utente** (ad esempio le interfacce a riga di comando e i sistemi a finestra) hanno consentito di passare da una macchina in grado di risolvere una serie di problemi, in maniera poco o nulla interattiva, in un oggetto dalle mille funzioni, con il quale possiamo interagire in tempo reale attraverso differenti dispositivi di ingresso (tastiere, mouse, touchscreen, telecamere, microfoni) e di uscita (monitor e schede video, stampanti , casse acustiche e schede audio). Con il software dei sistemi sono state sviluppate anche determinate modalità o

paradigmi di interazione (ad esempio l'interfaccia “desktop” che simula la scrivania e consente di manipolare gli oggetti virtuali con il mouse) che hanno reso facile tale interazione, fornendo allo strumento computer una notevole usabilità (vedi capitolo 3.6) per la realizzazione di moltissimi compiti (scrittura, contabilità, gioco). Mentre una volta per inserire dati dovevamo far leggere una scheda perforata e leggevamo come risultato dell'elaborazione testo stampato, oggi muoviamo il mouse, “clicchiamo” e vediamo gli effetti su uno schermo LCD a colori compiendo azioni che ci sembrano “naturali” pur essendo totalmente “virtuali” (la creazione di file, lo spostamento degli stessi, ecc.).

Figura 2: I personal computer con interfaccia di tipo "desktop" (che simulano l'ambiente di una scrivania) e il mouse sono diventati la tipologia "classica" di calcolatore elettronico

La macchina che oggi chiamiamo genericamente “personal computer”, da tavolo o portatile, è, in definitiva, un tipo particolare di calcolatore che si è evoluto nel tempo combinando l'architettura di calcolo che abbiamo visto con particolari **sistemi operativi** e programmi **applicativi** per l'utente utili per diverse attività lavorative e lo svago. Lo sviluppo del software di base ha avuto due grosse spinte, la prima in ambito accademico, con lo sviluppo dei sistemi operativi per le cosiddette “workstation” scientifiche, la seconda in ambito domestico e di ufficio, con lo sviluppo di sistemi operativi come Windows e MacOS. L'evoluzione di questi sistemi ha portato alla creazione di calcolatori che hanno assunto alla fine caratteristiche abbastanza simili (multitasking, interfacce grafiche a finestre, paradigma della scrivania, uso del mouse come puntatore, ecc.). Essi, anche grazie all'aggiunta della connettività di rete, che ne ha consentito l'uso per comunicazione e trasmissione dati a distanza (vedi capitolo 2), hanno potuto trovare un ampio spettro di applicazioni che va molto al di là dell'uso scientifico o gestionale. Oggi quando parliamo di personal computer, a parte le differenze tra portatili e fissi, abbiamo più o meno in mente sistemi hardware/software con caratteristiche ben precise ed un interazione utente ben precisa. Su queste “piattaforme”, le aziende produttrici di software hanno sviluppato applicazioni che ci consentono farne svariati utilizzi: dallo strumento di scrittura e impaginazione al sistema di montaggio video dal telefono all'enciclopedia multimediale, dal videoregistratore al videogioco.

Ad aumentare ulteriormente la disponibilità di programmi utilizzabili da parte degli utilizzatori dei PC altra svolta importante è stata la diffusione del **software libero.**

Con la diffusione dei calcolatori e delle reti, infatti, è stato possibile il diffondersi di un modello alternativo di portare avanti i progetti software che, anziché essere realizzati solamente da aziende per essere immessi sul mercato, sono, in molti casi, sviluppati da comunità di persone (eventualmente supportate da sponsor), allo scopo di renderli liberamente disponibili e modificabili. Grazie al software libero (per chi fosse interessato a saperne qualcosa di più rimandiamo all'Appendice 1) oggi abbiamo a disposizione sui nostri calcolatori molte applicazioni informatiche efficienti e sofisticate che possono essere utilizzate senza necessariamente spendere soldi per costose licenze di utilizzo (oltre che di un completo sistema operativo “libero”, come GNU-Linux). Anche allo scopo di promuovere la conoscenza e la diffusione del software libero, tutti gli esempi di creazione di contenuti multimediali che presenteremo nel seguito del libro saranno realizzati con programmi di questo tipo.

1.4 L'informatica pervasiva e le sfide del futuro

Abbiamo detto che con i termini PC o computer, ci riferiamo di solito ad un modello particolare di macchina calcolatrice dotata di interfacce e software particolari e che deriva dai modelli di macchina da ufficio e da laboratorio sviluppati negli anni '70-'80. E' evidente, però, che oggi tale modello non è più l'unica forma di calcolatore elettronico ad avere diffusione di massa e, anzi, tale modello sta perdendo via via "centralità" nel mondo dell'informatica e nel relativo mercato.

Lo sviluppo di nuovi tipi di dispositivi di ingresso (touchscreen, telecamere, sensori di posizione, accelerometri) e di display (schermi LCD minuscoli o enormi, proiettori, sistemi 3D, ecc.) e la miniaturizzazione dei componenti ha consentito, infatti, di realizzare computer di varia forma e dimensione che sono stati integrati in moltissimi oggetti del mondo che ci circonda e che permettono esperienze di utilizzo molto differenti.

Al giorno d'oggi processori di calcolo e sistemi operativi sono "nascosti" in cellulari, lettori multimediali, libri elettronici, proiettori, stazioni meteo, navigatori GPS, televisori LCD, elettrodomestici di ogni tipo. Tutti questi dispositivi possono essere, in un certo senso, considerati veri e propri computer che si differenziano dal classico modello da scrivania per i loro particolari tipi di interfacce e sistemi di interazione e per i diversi scenari di utilizzo. E' la cosiddetta "**informatica pervasiva**" o "ubiquitous computing": oggi possiamo svolgere molte delle operazioni che prima potevano essere ottenute solo col PC (leggere e scrivere documenti, collegarsi a Internet, scambiare messaggi di testo, consultate mappe, ecc.) con strumenti portatili di vario genere (telefoni cellulari, smartphone, tablet PC, navigatori, postazioni pubbliche, ecc.), mentre svariati oggetti precedentemente "analogici" sono diventati elaboratori di dati digitali (telefoni, televisori, elettrodomestici vari).

Questi nuovi strumenti presentano, come accennato, specifiche peculiarità riguardanti i cosiddetti dispositivi di input/output con i quali comunichiamo con la macchina, ed ovviamente hanno anche specificità riguardanti l'hardware ed i sistemi operativi con le relative interfacce utente. Dall'uso dei classici tastiera, mouse e schermo si sta passando a strumenti sempre più sofisticati che però consentono un'interazione quasi naturale: dal touchscreen con opzioni multitouch che permette di impartire ordini al cellulare o al portatile muovendo le dita sullo schermo, alle telecamere con riconoscimento di gesti, usate per interagire con schermi lavagne e videogiochi, al riconoscimento vocale. Molti di questi "computer" possono anche acquisire informazioni su ciò che li circonda in maniera autonoma, attraverso sensori in grado di "percepire" il mondo esterno indipendentemente dall'azione dell'utente: basti pensare a due componenti ormai integrati in molti dispositivi di uso comune come il GPS (Global Positioning System) in grado di fornire allo strumento stesso la propria collocazione geografica, e l'accelerometro, che fornisce indicazione all'apparecchio riguardo alla sua orientazione spaziale e al movimento. Anche i display di queste macchine moderne sono molto vari: abbiamo schermi piccoli sui dispositivi portatili e schermi piatti enormi su cui si possono realizzare presentazioni pubbliche o costruire lavagne interattive, esistono schermi in grado di rappresentare scene stereo, cioè di creare effetti di tridimensionalità usando occhialini o filtri o addirittura schermi olografici 3D che consentono di simulare oggetti reali. Ognuno dei nuovi tipi di calcolatore che stanno emergendo ha i suoi particolari elementi hardware, sistemi operativi, linguaggi di programmazione, eccetera. Ci sono i processori "embedded" (nascosti), sistemi operativi "embedded" per vari tipi di dispositivo, poi ci sono componenti hardware e sistemi operativi specifici per palmari e cellulari, come iPhone OS, Windows Phone, Symbian, Android, su cui la battaglia commerciale è oggi più accesa e gli investimenti maggiori che per

quanto riguarda i PC da tavolo. Dal punto di vista concettuale, però, il funzionamento di tutte queste piattaforme rimane sostanzialmente lo stesso ed è analogo a quello dei PC classici. Se vogliamo, l'evoluzione dei calcolatori ha seguito il percorso schematizzato in Figura 4: da macchina di calcolo, il calcolatore si è trasformato in macchina domestica con il PC, questa ha poi assunto, grazie all'accresciuta potenza, una grande varietà di differenti utilizzi, che sta naturalmente portando ad una differenziazione e specializzazione delle interfacce fisiche con cui utilizzarli.

Figura 3: Un moderno smartphone funziona sostanzialmente come un Personal Computer (ed è un calcolatore molto più potente dei PC anni '80-'90)

La rivoluzione dell'informatica pervasiva è oggi nel pieno del suo boom ed è difficile prevederne le evoluzioni future. Essa sta comunque già cambiando anche il modo di utilizzare la rete, comunicare e fruire di contenuti multimediali, ossia ciò di cui parleremo nei prossimi capitoli. Si prevede, per esempio, che entro pochi anni il traffico sulle reti telematiche dovuto a dispositivi mobili come smartphone (cellulari-computer) o tablet (come il recente iPad della Apple) supererà quello dovuto ai calcolatori fissi. E già oggi (2011), come ha sottolineato un guru delle tecnologie come Chris Anderson (direttore di Wired USA) in un articolo intitolato “Il web è morto, lunga vita a Internet” il modo di fruire dei contenuti in rete è già spesso passato dal classico PC con browser per “navigare” (vedi capitolo 2.6) ad altri tipi di modalità (tramite le cosiddette applicazioni o “app” dei cellulari, le console dei videogiochi, i televisori di nuova generazione, ecc.). Se, da una parte, questa rivoluzione non cambia molto nelle competenze di base che servono per creare contenuti e comunicare con calcolatori e reti, essa richiede sempre più di abbinare le conoscenze tecnologiche a quelle sull'analisi dell'interazione tra uomo e macchine cui faremo cenni durante i prossimi capitoli, e che tenta di rendere gli strumenti più familiari e semplici da usare per gli utenti, in funzione dei compiti che essi devono eseguire.

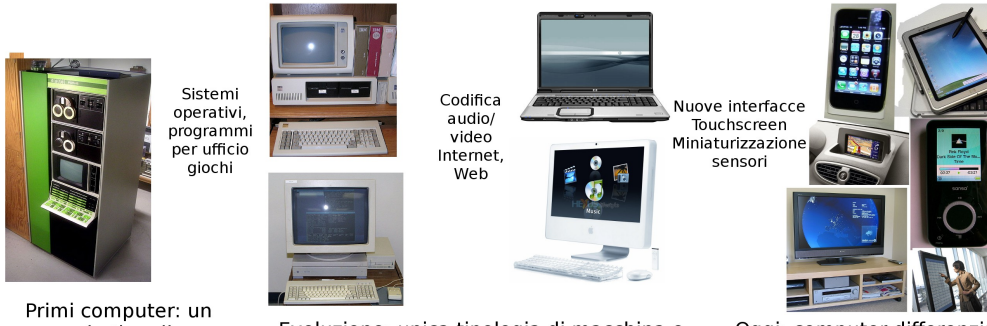


Figura 4: L'evoluzione dei calcolatori ha portato dapprima ad un singolo tipo di calcolatore ed interfaccia utente con molte differenti funzioni, ma oggi si stanno sviluppando sistemi con modalità di interazione ed interfacce differenti.

2 Reti telematiche ed Internet

Se lo sviluppo di computer e sistemi operativi ha consentito di creare, elaborare e rappresentare contenuti complessi, una componente altrettanto cruciale per lo sviluppo di gran parte delle moderne applicazioni di successo dei calcolatori elettronici è stata la possibilità di trasmettere a distanza i dati digitali.

Anche in questo caso, tecnologie inizialmente nate per scopi relativamente semplici (ad esempio duplicare informazione a distanza per motivi di sicurezza o condividere risorse di ufficio come stampanti o server di calcolo) si sono poi rivelate utili per una grande varietà di applicazioni, inimmaginabili all'epoca all'epoca della loro introduzione.

E se l'evoluzione tecnologica è stata fondamentale per incrementare le prestazioni ed arrivare, ad esempio, a trasmettere musica e video di alta qualità o ad usare la rete per telefonate e videoconferenze, le modalità con cui avviene la comunicazione digitale oggi sono rimaste simili a quelle utilizzate per le prime reti.

Quello che è necessario per trasmettere e ricevere dati digitali su un canale di trasmissione è sostanzialmente definire dei **protocolli**, cioè degli insiemi di regole accettate da chi trasmette e chi riceve i dati stessi. Le complicazioni tecniche nel definire tali protocolli e realizzare praticamente le reti sono notevoli, ma le idee su cosa essi debbano definire sono intuitive e comprensibili anche a chi non è un tecnico.

E' chiaro che, se si vogliono far parlare tra loro i calcolatori di una rete locale (LAN) attraverso segnali di tensione, occorre, prima di tutto, definire forma e tipologia dei cavi di collegamento e fissare codifica e temporizzazione dei segnali elettrici corrispondenti alle cifre binarie 0 e 1 (ricordiamo che vogliamo trasmettere dati digitali).

Se si vuole creare una rete che colleghi più macchine e non semplicemente un collegamento punto a punto, e se la rete, come usualmente accade è di tipo **broadcast**, cioè ha un unico canale che unisce diverse macchine, occorre che i protocolli definiscano anche **indirizzi** univocamente riconosciuti per fare in modo di poter inviare il messaggio al destinatario desiderato. Occorre poi evitare che sul canale vengano inviati contemporaneamente più messaggi (che diventerebbero illeggibili) o, almeno, adottare delle strategie di reinvio dei dati se ciò accade. Occorre, infine, che la struttura dei messaggi sia nota a trasmittente e ricevente affinché sia possibile la decodifica. Questo è quanto, in effetti, stabiliscono i protocolli per gestire le reti locali.

Per gestire trasmissioni tra reti locali differenti e costruire le cosiddette reti geografiche occorrerà poi regolare il funzionamento dei nodi (bridge, router, ecc.) che collegano le sottoreti componenti.

Infine occorre che vengano sviluppati programmi applicativi che costruiscano, attraverso questo scambio organizzato di informazioni tra PC lontani, attività ad alto livello come la distribuzione di documenti, la comunicazione scritta, o quella vocale.

I protocolli che si usano per le reti di calcolatori sono stratificati a **livelli** in una maniera che ricorda quella dei livelli di interazione con i computer definiti nei sistemi operativi (in cui i vari moduli si stratificano dal nucleo che interagisce con la CPU alle interfacce che interagiscono con l'utente).

Nelle reti, se a basso livello si stabilisce quali sono i meccanismi con cui i bit possono passare da una macchina all'altra (segnale elettrico, cavo o onde radio, ecc), ai livelli superiori, cui

accedono programmatori ed utenti, si sovrappongono protocolli via via sempre più astratti (relativi all'indirizzamento, alla struttura dei messaggi, ai servizi). Chi utilizza un programma di chat o un browser o magari telefona via Internet con il programma Skype, non vede che i messaggi che invia e riceve vengono tradotti a vari livelli, fino a diventare i pacchetti standard del protocollo di rete (cioè sequenze di bit opportunamente strutturate) che passano su differenti tipi di mezzo fisico (cioè i cavi, le onde radio, ecc.). L'International Standard Organisation (ISO), organismo per la standardizzazione internazionale, ha definito sette livelli teorici per le reti di telecomunicazione: livello fisico (hardware, definizione cavi e collegamenti, livelli di tensione, temporizzazioni), livello data link (controllo del flusso di dati), livello rete (gestione dell'indirizzamento), livello trasporto (gestione del messaggio trasmesso da sorgente a destinazione), livello sessione (gestione delle connessioni tra macchine), livello presentazione (formati dati, crittografia) e livello applicazione (interfacce utente). Il modello risultante viene detto **ISO/OSI,** cioè Open System Interconnection. Nelle implementazioni pratiche dei sistemi di rete, si creano le componenti hardware e software per elaborare i messaggi ad ogni livello, facendo in modo che a ciascun livello non si abbia percezione di cosa ci stia sotto o sopra. In questo modo si possono sviluppare indipendentemente applicazioni ad alto livello e gestione della rete cambiando eventualmente solo la parte di “traduzione”.

Sono stati sviluppati diversi protocolli e metodi di gestione delle reti, anche se oggi quasi tutte le applicazioni che utilizziamo sono basate su un'insieme di protocolli standard legati allo sviluppo di Internet. Non è certo lo scopo di questo testo descriverli in dettaglio, ricorderemo qui di seguito soltanto alcune idee di base relative alla trasmissione dei dati attraverso le reti di calcolatori che possono essere utili per farsi un'idea del funzionamento (e dei problemi) delle reti telematiche:

- Le macchine possono essere collegate punto a punto (allora non ci sono problemi di indirizzamento) oppure essere collegate direttamente o indirettamente ad uno stesso canale (bus). In questo caso occorre chiaramente che si definiscano indirizzi relativi alle macchine collegate in rete ed accessibili.
- La trasmissione dei dati avviene a pacchetti: i protocolli di rete di basso livello stabiliscono che la trasmissione dei messaggi sia spezzettata ed organizzata a blocchi (pacchetti) di bit che vengono, in genere, trasmessi indipendentemente (questo significa che i pacchetti conterranno tutta l'informazione relativa all'indirizzamento e al loro riordinamento necessario per la ricostruzione del messaggio completo a destinazione)
- Per quanto riguarda l'accesso alla rete su un bus con diversi computer, il protocollo dominante per le reti locali si chiama CSMA/CD (Carrier Sense Multiple Access with Collision Detection): si tratta di un protocollo che permette a più stazioni di trasmettere contemporaneamente sullo stesso canale gestendo le collisioni mediante rilevazione del problema e conseguente ritrasmissione dei messaggi con sfasamento temporale. Questo vuol dire, in pratica, che, sebbene il metodo si sia rivelato efficiente, non c'è alcuna garanzia che in condizioni di congestione della rete non ci possa essere un ritardo infinito nella trasmissione di un pacchetto (non c'è insomma un tempo massimo garantito di consegna del pacchetto stesso). Quindi quando si blocca la rete, non sta accadendo un fatto assurdo, ma semplicemente una condizione rara ma verificabile sulla base del protocollo usato.
- Dato che il numero e la distanza dei computer collegati in rete è limitato da questioni tecniche legate al tipo di cavi, alle interferenze ed al traffico di dati, per costruire reti più grandi delle **reti locali (LAN)**, cioè quelle relative a uffici, scuole, ecc, occorre collegare piccole sottoreti tra loro attraverso macchine particolari, che si chiamano **router, bridge,**

gateway o in generale **interface message processor**, che smistano il traffico tra sottoreti differenti e creano grandi reti di tipo **commutato** (cioè in cui il traffico viene smistato da stazioni intermedie, come nell'esempio di Figura 5).

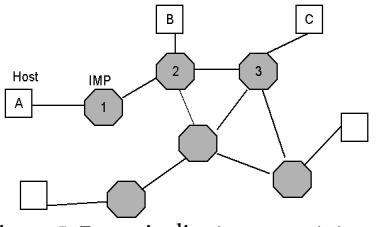


Figura 5: Esempio di rete commutata: per raggiungere l'host C dall'host A si attraversano diversi nodi di smistamento (ad es. 1,2,3), ed il cammino non è univoco.

- La trasmissione a pacchetti lungo una rete commutata (cioè con stazioni intermedie che collegano sottoreti, come avviene in Internet) utilizza quindi la cosiddetta **commutazione di pacchetto**: in essa, per ogni trasmissione, non viene bloccato un circuito che collega sorgente e destinazione per tutto il tempo della comunicazione, ma la trasmissione dei pacchetti è indipendente. Questa è una differenza sostanziale rispetto alla classica commutazione di circuito delle linee telefoniche: se più stazioni si collegano alla stessa, non troveranno quindi mai "occupato", al più ci potrà essere un rallentamento nelle prestazioni dovuto alla congestione della rete.

Le reti di calcolatori sono nate inizialmente con scopi principalmente locali (Local Area Network o LAN): condividere le risorse di calcolo di un computer centrale, oppure servizi quali, ad esempio, la stampa. Le reti sono poi state interconnesse tra loro in reti più ampie, consentendo di lavorare su computer distanti o di trasferire dati anche tra diverse nazioni o continenti (reti urbane, geografiche, wide-area, ecc.).

Internet in pratica è l'unico esempio di rete ad accesso pubblico di dimensione globale: essa è una rete di reti con un unico sistema di indirizzi che permette di mandare e ricevere messaggi ai diversi nodi.

Vediamo in un breve excursus storico come si sia arrivati alla creazione di questa rete che collega in pratica tutto il mondo informatizzato e quali problemi ed opportunità questo abbia creato.

2.1 Internet

Le origini di Internet risalgono agli anni sessanta del secolo scorso, quando una prima infrastruttura di rete di reti fu creata dal Dipartimento della Difesa degli Stati Uniti, con finalità prettamente militari. Nel pieno della guerra fredda si intendeva, infatti, creare una rete di calcolatori in grado di garantire la continuità della comunicazione nell'eventualità di attacchi nucleari. A tale scopo era necessario realizzare una rete in grado di funzionare anche in caso di una sua parziale distruzione: un messaggio proveniente da un computer doveva poter raggiungere una destinazione seguendo strade differenti in modo che se una fosse stata interrotta, se ne potesse utilizzare un'altra. Il progetto dell'infrastruttura di rete prese il nome di ARPAnet.

Per la realizzazione di tale progetto vennero impegnate enormi risorse, sia tecniche che umane, queste ultime provenienti principalmente dal mondo accademico.

Venne così creato il protocollo IP (Internet Protocol), attraverso il quale venne garantita la comunicazione tra i primi nodi della rete di calcolatori, di per sé rivoluzionario, in quanto non richiede che i computer collegati siano dello stesso tipo, e nacquero le prime reti su aree locali

(LAN).

Successivamente si dovevano interconnettere le LAN, fare in modo cioè che ogni computer connesso ad una rete locale potesse connettersi non solo agli altri computer della stessa LAN ma anche agli altri computer di altre LAN e quindi ai servizi ARPAnet.

Mentre il progetto ARPAnet cresceva e si sviluppava, altre organizzazioni di differente estrazione iniziarono a sfruttare il protocollo IP per mettere in comunicazione propri calcolatori: negli anni '80, quando in ambito scientifico le reti iniziavano a proliferare la Fondazione Scientifica Nazionale (NSF) statunitense iniziò a sviluppare una rete di reti propria, differente da quella militare, sempre basata sulla tecnologia IP.

Per connettere i calcolatori si usarono anche le linee telefoniche: si formarono reti regionali per la connessione delle strutture educative circostanti al centro più vicino; ogni centro veniva poi collegato con gli altri centri di supercalcolo, con il risultato che qualsiasi computer della rete poteva comunicare con qualsiasi altro, purché quest'ultimo fosse connesso ad una rete a sua volta collegata ad un centro.

Il successo fu immediato, anche se la veloce crescita del numero di utenti e l'intensificarsi del traffico resero presto la struttura inadeguata (prestazioni insufficienti). Verso la fine degli anni '80 si assistette così ad una completa ristrutturazione della rete telefonica statunitense: le vecchie linee vennero sostituite con linee di tecnologia più recente, più veloci ed affidabili. Grazie ai miglioramenti infrastrutturali, mantenendo lo stesso protocollo, l'accesso alla rete non rimase prerogativa di ricercatori e militari, ma divenne possibile anche agli studenti dei college. Questo cambiò completamente il modo di fruire dei servizi di rete e nacque così l'Internet che oggi conosciamo, con servizi pensati per il grande pubblico.

Nel frattempo erano state anche migliorate le tecnologie di rete e i servizi di rete basati su TCP/IP furono estesi anche alle università europee e extra-europee: Internet iniziò così a crescere esponenzialmente. Alla fine degli anni '80 il mondo accademico era già connesso in un'unica rete mondiale.

Mentre nei primi anni '90, la vecchia rete militare ARPAnet veniva eliminata in quanto ritenuta troppo costosa per essere gestita dal ministero della difesa, accadde un ulteriore evento che impresse una svolta decisiva allo sviluppo della Rete: la nascita del World Wide Web (WWW), lo strumento di fatto più semplice ed immediato per l'esplorazione di Internet, risultato di una ricerca portata avanti dal CERN (centro di ricerca di fisica nucleare di Ginevra).

Si tratta di un progetto per distribuire informazioni attraverso l'uso dello strumento dell'**ipertesto** (vedi cap. 2.7), che ottenne subito un grandissimo successo.

Nel 1992 il centro nazionale statunitense per il supercalcolo (NCSA) realizzò il primo **browser** per il Web (ovvero il programma cliente per usufruire dei servizi messi a disposizione dai siti), chiamato **Mosaic** ed il primo **sito web** pubblico a cui tutti potevano connettersi. In breve tempo moltissimi siti vennero realizzati e il numero dei server web e delle informazioni rese disponibili divenne ben presto notevole. Altri produttori iniziarono a sviluppare i browser (ad esempio Netscape) e questo contribuì anche a migliorare i meccanismi di codifica (le aggiunte al codice degli ipertesti operate da essi vennero sono spesso poi diventate standard) ed arricchire i siti. Dell'evoluzione e delle caratteristiche del web ci occuperemo in dettaglio nel capitolo 2.6.

Internet ha meccanismi di standardizzazione diversi da quelli generici della rete, gestiti dall'ISO. Quando venne creata ARPAnet il Dipartimento della Difesa creò un comitato per supervisionarla, in seguito detto IAB (Internet Architecture Board www.iab.org), i cui gruppi di

lavoro producevano rapporti tecnici chiamati RFC (Request For Comments). I RFC sono in linea (*www.ietf.org/rfc.html*) e possono essere letti da chiunque sia interessato. Sono numerati in ordine cronologico di creazione; ne esistono quasi 2.000. Nel 1989 IAB venne riorganizzato. Venne creata l'Internet Society (*www.isoc.org*), associazione di singoli interessati allo sviluppo di Internet. IAB divenne un comitato afferente a Internet Society, e vennero create due nuove entità: IRTF (Internet Research Task Force www.irtf.org), e IETF (Internet Engineering Task Force www.ietf.org). IRTF si concentra sulla ricerca a lungo termine, mentre l'IETF gestisce i problemi dell'ingegnerizzazione a breve. Nel 2000 il governo americano ha formato un organismo internazionale indipendente e non-profit, Internet Corporation of Internet Names and Numbers, ICANN (*http://www.icann.org*), col compito di provvedere alla gestione dello spazio dei nomi di Internet.

Il **W3C** (*http://**www.w3c.org***) è un insieme di istituzioni pubbliche e private che si occupa della definizione e supervisione degli standard tecnologici del WWW (es.: HTML, XML sono standard del W3C). Il W3C propone nuovi standard tecnologici per il WWW, che continuano a evolvere (si pensi alla continua evoluzione dei servizi di trasmissione video, ai contenuti dinamici e così via).

Come funziona

Internet è un sistema di informazione globale che:

- è logicamente connesso mediante un unico spazio di indirizzi basato sul protocollo IP (Internet Protocol);
- è in grado di gestire le comunicazioni tramite il protocollo TCP (Transmission Control Protocol);
- rende accessibili agli utenti, sia pubblicamente che privatamente, servizi di alto livello dedicati allo scambio di informazioni (mail, testo, file, etc…).

L'insieme dei due protocolli appena definiti viene spesso riferito come il protocollo TCP/IP. In realtà, come si è visto, sono due protocolli distinti che agiscono su livelli differenti. Il modello di Internet differisce da quello definito dall'OSI; sono definiti 5 livelli contro i 7 di ISO/OSI:

1. I nodi-computer vengono fisicamente connessi da cavi: questo è il livello dei protocolli fisici, che include anche i dispositivi modem
2. I nodi-computer condividono un codice di trasmissione di bit: questo è il livello dei protocolli data-link
3. Un pacchetto di bit riesce a viaggiare sulla rete via più nodi: questo è il livello del protocollo rete (es.: IP)
4. Una sequenza di pacchetti riesce a raggiungere una destinazione per vie diverse e ad essere ricostruita: questo è il livello del protocollo trasporto (es.: TCP)
5. Le applicazioni riescono a stabilire delle regole per comunicarsi dati e documenti in forma binaria: questo è il livello dei protocolli di tipo applicazione (per esempio HTTP, SMTP, ecc.).

Il livello di rete

Il livello rete (network) si occupa di fare arrivare i pacchetti dalla sorgente alla destinazione. Dato che stiamo parlando di una rete **commutata** complessa, nel percorso può essere necessario attraversare diverse stazioni intermedie (Interface Message Processor) che smistano il traffico

tra le diverse sottoreti. IP è il protocollo di livello rete di Internet: non prevede controlli sull'integrità dei dati dalla partenza alla destinazione, si occupa solo della confezione e dello smistamento dei pacchetti.

Esistono molti tipi di reti: reti locali, reti geografiche, reti via cavo, reti via etere, reti statiche, reti dinamiche, ecc.: non ha senso progettare un'unica rete standard, meglio adattarsi a convivere con la nozione di rete di reti; le diverse reti vengono connesse da macchine specializzate.

Ricordiamo qui brevemente che, così come i mezzi trasmissivi per costruire reti sono molto variabili (cavi, onde radio, fibre ottiche), anche i dispositivi per connettere tratti di rete possono essere di vario tipo e riflettono in generale il livello di protocollo a cui si collocano: se sono a livello fisico si dicono **repeater** e non fanno alcuna traduzione o filtro dei pacchetti, se sono a livello data-link si dicono **bridge** o **switch** e collegano reti anche eterogenee "punto a punto", a livello rete (in questo caso IP) si parla di **router** che instradano i pacchetti che gli arrivano in base all'indirizzo di destinazione, a livello TCP si parla invece di **gateway**, macchine in grado di gestire completamente il traffico dei dati.

IP definisce una rete a commutazione di pacchetto e con trasporto inaffidabile (non si garantisce la consegna dei pacchetti): necessita di un altro protocollo a livello superiore che rilevi e gestisca i pacchetti persi o fuori sequenza. Tale protocollo è in genere TCP (esistono alternative, ad esempio UDP, usato in applicazioni ove è meno importante la garanzia di consegna). Ogni pacchetto è un messaggio indipendente dagli altri; eventuali associazioni tra pacchetti devono quindi essere definite da protocolli di livello superiore.

Ogni pacchetto deve contenere, oltre la sequenza di bit del messaggio, anche delle informazioni aggiuntive necessarie all'indirizzamento e alla ricostruzione dei messaggi priva di errori. I campi più importanti sono gli indirizzi di mittente e destinatario, che permettono di individuare in modo univoco le macchine coinvolte nella comunicazione. Vi sono inoltre campi di controllo per scartare pacchetti corrotti.

Le reti che usano il protocollo IP assegnano a ciascun nodo della rete un **indirizzo su 32 bit,** di solito raggruppati in 4 byte e rappresentati graficamente con quattro numeri interi tra 0 e 255 (es. 156.148.21.222). Chi naviga su Internet ha sicuramente visto nei menu di configurazione o collegandosi ai siti indirizzi scritti in questa forma.

Ogni indirizzo IP contiene 3 campi: la classe, l'identificatore di rete e l'identificatore di host. I numeri sono inizialmente assegnati da NIC (Network Information Center – www.internic.net); esistono entità nazionali che gestiscono i numeri delle varie nazioni.

I bit che compongono un indirizzo di IP individuano 4 classi di reti:

- classe A: reti giganti (primo bit 0; 7 bit per identificare la sottorete e 24 per gli host, cioè i singoli computer all'interno della sottorete)
- classe B: reti medie (primi due bit=10, 14 bit per la rete, 16 per gli host)
- classe C: reti piccole (primi tre bit=110, 21 net bit, 8 per gli host)
- classe D: indirizzi multicast (primi quattro bit =1110 + 28 bit)

Per esempio, la macchina che ha per indirizzo IP 157.27.10.55 è situata presso l'Università di Verona. La prima parte dell'indirizzo (157.27) individua una rete di classe B ed è stato assegnato dal NIC all'Università di Verona. La seconda parte è stata assegnata dal gruppo dei sistemisti dell'azienda (10.55 vuol dire computer n. 55 collegato alla sottorete n. 10). Le classi di formato A, B, C e D contengono (A) 126 reti con 16 milioni di host ciascuna, (B) 16382 reti

con 64K host ciascuna, (C) 2 milioni di reti (ad es., LAN) con 254 host ciascuna e (D) un insieme di indirizzi di multicast, nei quali un pacchetto viene spedito a più di un host.

Il modo di indirizzare poteva essere considerato uno degli "errori di progetto" più gravi di Internet, dal momento che la crescita dell'uso della rete ha portato all'avvicinarsi della saturazione degli indirizzi (al massimo con 32 bit sarebbero 2^{32} cioè circa 4 miliardi). Questo pericolo è stato per il momento scongiurato dall'uso di tecniche che limitano l'uso di macchine quasi sempre collegate alla rete e con indirizzo IP fisso (in termine tecnico si dice "**IP statico**"). Gran parte delle macchine connesse alle sottoreti aziendali, infatti, come i PC domestici che usiamo connettendoci alla rete, non hanno un indirizzo IP di questo tipo assegnato, ma viene loro assegnato all'interno della sottorete aziendale (o di una sottorete gestita dal provider locale quando ci colleghiamo privatamente) un indirizzo temporaneo (**IP dinamico**) per il solo tempo di collegamento, o addirittura locale, non visibile all'esterno in quanto filtrato poi da altre apparecchiature che "traducono" gli indirizzi tra la rete interna e la esterna (Network Address Translator).

Utilizzando tali tecniche, anche se non disponiamo di un nostro indirizzo IP statico possiamo accedere alla rete, navigare, accedere ai servizi di tutte le macchine visibili in rete, ma dobbiamo ricordare che il nostro PC non sarà visibile con un indirizzo IP fisso dall'esterno e non sarà quindi adatto ad ospitare un programma "server".

Per ovviare al comunque preoccupante problema del futuro esaurimento degli indirizzi di rete, è stato sviluppato un nuovo spazio di indirizzi (IPv6), una sorta di Internet 2 con indirizzamento in grado di gestire 10^{38} differenti indirizzi che dovrebbe via via sostituire l'attuale.

Il Domain Name Service (DNS)

Ricordarsi il nome di una macchina collegata in rete tramite un indirizzo di IP non risulta molto comodo. Ed infatti di solito non usiamo direttamente l'indirizzo IP per accedere ai servizi forniti dai vari host che contattiamo. Questo accade grazie al fatto che nel 1984 è stato introdotto il **Domain Name Service** (DNS) che consente di usare dei nomi (stringhe di caratteri alfabetici) per individuare univocamente gli host stessi.

Il vantaggio è evidente, dato che risulta certamente più comodo ricordarsi un indirizzo testuale come www.repubblica.it (il nome del server web del quotidiano online più consultato) piuttosto che 194.185.98.154 (l'indirizzo IP di quel server).

Occorre, però, dotare la rete di un sistema per la conversione degli indirizzi che possa gestire l'enorme mole degli stessi. La soluzione creata è un classico sistema client-server: le conversioni di indirizzi sono eseguite da opportuni calcolatori che prendono il nome di **server DNS**. Possiamo pensare ad ogni server DNS come una macchina in grado di fornire un meccanismo di conversione delle stringhe (nomi domini) in indirizzi di rete IP (che sono poi quelli che Internet usa per indirizzare i dati). In teoria un solo server DNS potrebbe contenere l'intero database DNS mondiale; in pratica esso sarebbe così sovraccarico da essere in pratica inutilizzabile. Inoltre, se mai si guastasse, l'intera Internet sarebbe bloccata.

Per tale motivo ci sono più server DNS che formano un sistema **distribuito** (ogni DNS ha le proprie tabelle di conversione stringa-indirizzo IP). I server DNS sono organizzati gerarchicamente. Quando un programma deve trasformare un nome in un indirizzo IP chiama una procedura (software) detta risolutrice (**resolver**), passandole il nome come parametro di ingresso. Il resolver interroga un server DNS locale, che cerca il nome nelle sue tabelle e restituisce l'indirizzo al resolver, che a sua volta lo trasmette al programma chiamante (usando

poi tale indirizzo IP il programma può quindi aprire una normale connessione TCP con la destinazione).

Se il DNS locale non trova l'informazione nelle sue tabelle, allora invia un'interrogazione per il dominio richiesto al name server al livello più alto della gerarchia. La gerarchia viene scalata finché il DNS locale non riceve la risposta cercata (in tal caso la inserirà nelle sue tabelle in modo da non dover più ripetere l'interrogazione qualora l'utente accedesse nuovamente a quel dominio). Le tabelle sono dinamiche: se il server DNS ha bisogno di inserire una nuova associazione dominio-IP e non c'è spazio, provvederà a mettere tale informazione al posto della riga contenente l'associazione meno usata (questa tecnica è detta caching).

Qualora nessun DNS sia in grado di effettuare la conversione, alla macchina richiedente verrà inviato un messaggio di errore "Host not found!" (il dominio richiesto non corrisponde a nessuna macchina).

Esempio: Supponiamo che un name server locale non abbia mai ricevuto prima un'interrogazione per il dominio a.b.edu e non ne sappia niente. Può chiedere ai name server più vicini, ma se nessuno di questi conosce tale dominio, bisogna interrogare il server radice .edu. Se anche il server radice non conosce l'indirizzo di a.b.edu, deve però per definizione conoscere tutti i propri figli, domini di secondo livello, quindi invia la richiesta al name server per b.edu. A sua volta questo invia la richiesta al dominio a.b.edu, che deve avere la tabella di conversione. Una volta che l'indirizzo di IP record è restituito al name server di partenza, qui verrà conservato temporaneamente nel caso serva più tardi.

2.2 L'accesso ad Internet

Abbiamo finora definito come sono organizzati i protocolli e gli indirizzi che permettono di far dialogare gli host collegati direttamente alla rete (con cavi di vario tipo, dispositivi di interfaccia tra sottoreti, ecc.). Ma non tutti i computer che utilizziamo anche se "collegati" ad Internet, sono in realtà host direttamente collegati alla rete e dotati di indirizzo IP "**statico**", cioè assegnato permanentemente, e magari con nome associato dal DNS. Del resto, come abbiamo visto, gli indirizzi a disposizione, ormai in via di saturazione, non sarebbero certamente sufficienti a soddisfare tutti gli utenti di PC. Inoltre occorrerebbe realizzare complesse infrastrutture di collegamento.

La maggior parte degli utenti dei servizi di Internet accede alla rete indirettamente, collegando il suo PC o la sottorete della sua azienda alla rete Internet attraverso un cosiddetto **ISP** (**Internet Service Provider,** ovvero una ditta che fornisce l'accesso ad Internet). All'interno della sottorete locale o sul proprio computer connesso via ADSL o wireless, si utilizza, per la comunicazione, sempre il protocollo IP, ma gli indirizzi assegnati alle proprie macchine sono temporanei o dinamici, di solito ottenuti mediante un meccanismo client-server detto DHCP, cioè Dynamic Host Configuration Protocol. La connessione con la rete esterna è mediata dal router del provider che instrada i pacchetti da e verso gli altri nodi, traducendo gli indirizzi. Questo fa sì che il PC dell'utente non abbia un indirizzo IP fisso, ed inoltre tale indirizzo potrebbe anche non essere visibile dall'esterno, dato che potrebbero essere utilizzati sistemi di traduzione degli indirizzi. Questo significa che non è in generale possibile utilizzare il proprio computer per fornire determinati tipi di servizi Internet.

Da un punto di vista pratico, il collegamento da casa in Italia viene realizzato in genere tramite il canale telefonico o attraverso trasmissioni radio. Esistono oggi diverse tecniche per il collegamento cablato: se un tempo il metodo dominante per le utenze domestiche era il modem

analogico, oggi si riescono ad ottenere alte velocità con le connessioni ADSL. Vediamo qualche dettaglio tecnico.

Collegamento tramite modem analogico

Un modo per far passare il segnale digitale attraverso la rete telefonica è di utilizzare un particolare dispositivo di I/O chiamato **modem** (**mo**dulatore / **dem**odulatore).

In uscita il modem converte il segnale digitale elaborato dal computer in un segnale analogico adatto ad essere trasportato sulla rete telefonica. Questa operazione si chiama tecnicamente modulazione (il modem effettua una modulazione in frequenza, assegnando una frequenza al bit di valore 0 e un'altra frequenza al bit di valore 1).

Alla ricezione (cioè in entrata) il modem preleva il segnale modulato e lo ritrasforma in sequenze di bit (questa operazione si chiama demodulazione).

Il computer attiva il modem il quale effettua una normale chiamata telefonica ad un numero fornito dall'ISP, corrispondente al cosiddetto POP (Point Of Presence), cioè l'apparecchio gestito dal provider che accede alla rete.

Il segnale che viaggia sul cavo è un normale segnale telefonico, quindi la linea viene occupata dalla trasmissione (e nessun altro può usare il telefono). Al POP di accesso corrisponde un altro modem che "risponde" al chiamante e la comunicazione può avere inizio. In questo caso si usa (tra modem e modem) un collegamento a commutazione di circuito. Il modem dell'ISP è poi collegato alla rete di Internet. Il modem analogico, ormai quasi in disuso, ha come pregio l'economicità, non richiedendo installazione di linee o dispositivi particolari. Il maggiore difetto consiste però nella lentezza: il segnale telefonico ha una banda molto ristretta e quindi non molto adatta alle trasmissioni dati. La banda teorica offerta per i modem analogici è tipicamente 57.6 Kbps in ricezione e 33.6 Kbps in trasmissione, ma, in pratica, raramente è possibile raggiungere i 15-20 Kbps. Il modem si deve adattare alla velocità della rete, quindi avere un modem a 64Kbps oppure a 32Kpbs è praticamente equivalente (poiché alla fine entrambi saranno utilizzati a 15-20 Kbps o meno). Di fatto è impossibile usufruire dei moderni contenuti multimediali con una banda così limitata. Chi fornisce contenuti in rete deve comunque ricordare che, purtroppo, esistono ancora zone del nostro paese non raggiunte da ADSL ed altri tipi di connessione veloce e che, dunque devono accedere alla rete con tali tecnologie lente.

Collegamento tramite ASDL

ADSL (Asymmetric Digital Subscriber Line) è una tecnologia che permette di trasformare la linea telefonica analogica (il tradizionale doppino telefonico in rame) in una linea digitale ad alta velocità per un accesso ad Internet ultra-veloce. La tecnologia ADSL utilizza la banda delle frequenze superiori a quella normalmente utilizzata per la voce (fonia) nella linea telefonica tradizionale per trasportare in forma numerica dati, audio e immagini. Quando sulla linea telefonica è abilitata ADSL, la banda disponibile viene suddivisa in tre sotto-bande di frequenza distinte: una dedicata alla ricezione dei dati da Internet (downstream), una dedicata all'invio dei dati verso Internet (upstream) ed un'altra molto ristretta riservata al traffico voce.

Per utilizzare il servizio è richiesta l'installazione di un'apposita coppia di filtri ADSL alle estremità della linea telefonica (presso la centrale telefonica e l'abitazione). Ci sono due possibili tipi di connessione, che garantiscono entrambi l'utilizzo della linea sia in fonia che dati. Nel primo caso (il più comune nel caso di utenze domestiche) si installa su ogni presa telefonica un dispositivo che prende il nome di **microfiltro** che permette di utilizzare in ogni diramazione sia il servizio ADSL che il comune telefono o fax. Il microfiltro separa il segnale

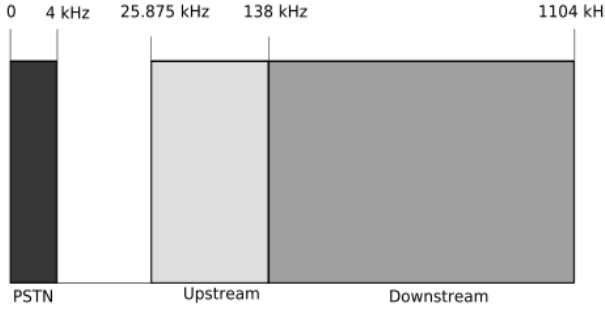


Figura 6: La suddivisione delle frequenze nei canali voce e dati utilizzata in ADSL.

telefonico standard dai diversi segnali che viaggiano sullo stesso cavo (le linee dati di ADSL): il telefono o il fax sono quelli usuali poiché a monte del filtro il protocollo di trasmissione è quello usuale (cioè il segnale è solo uno: quello telefonico).

Nel secondo caso viene installato un dispositivo simile al microfiltro, detto **splitter,** che si preoccupa di effettuare questo "taglio" di frequenze all'ingresso della linea ADSL nell'edificio. In questo secondo caso, la versatilità risulta essere ridotta, dato che la linea ADSL non arriverà necessariamente fino a tutti gli ambienti dove invece è presente la linea telefonica. Il lato positivo è dato dal fatto che non è necessario disseminare tutte le prese telefoniche con i microfiltri.

Nella **centrale di commutazione,** come è avvenuto nella sede del cliente un filtro separa le frequenze della linea telefonica da quelle della linea dati. In questo modo, le frequenze telefoniche prendono la strada della **PSTN** (Public Switched Telephone Network/Rete Telefonica Pubblica a Commutazione, la rete telefonica "normale"), invece quelle dati (ADSL) vengono convogliate verso un dispositivo che prende il nome di **DSLAM** (DSL Access Multiplier) che fa da interfaccia di collegamento con la rete Internet.

Le velocità solitamente disponibili per le connessioni ADSL sono dell'ordine dei 4 Mbps in ricezione e 640 in trasmissione, ma esistono anche le offerte *ADSL 2+*, a 12, 20 e 24 Megabit. Si tenga però conto che le velocità reali sono in genere più basse perché la banda è frazionata fra molti utenti.

I principali vantaggi offerti da ADSL sono:

- Linea telefonica indipendente dalla linea dati: è possibile usare il telefono e contemporaneamente usare Internet.
- Velocità di navigazione e trasferimento file teoricamente molto più alte di quelle di una linea digitale ISDN.
- Costi contenuti (non richiede nell'abitazione l'installazione di linee telefoniche supplementari o speciali).
- Il collegamento sempre attivo consente la ricezione immediata degli e-mail e di essere subito avvisati appena arriva un messaggio.
- L'utente è permanentemente connesso a Internet, senza la necessità di attivare ogni volta la connessione via modem.

Proprio il fatto di essere sempre collegati a Internet rappresenta un fattore di rischio per gli utenti. Infatti una connessione continua li espone all'attacco di **pirati informatici (cracker[1])**, persone che cercano di accedere, senza autorizzazione, alle risorse dei computer (aziendali e non). Occorre quindi prevedere una protezione adeguata (ad esempio installare dei **firewall**, cioè insiemi di programmi che prevengono l'accesso non controllato da parte di altri utenti).

Connessioni Wireless

Le cosiddette connessioni wireless (senza fili) sfruttano come mezzo di trasmissione le onde radio. Esistono vari tipi di protocolli di trasmissione di dati digitali attraverso onde elettromagnetiche, che sfruttano diversi intervalli di banda e che consentono differenti prestazioni per quanto riguarda la velocità di trasmissione dati. Citiamo qui di seguito i principali protocolli che si utilizzano per connettere i calcolatori a Internet.

Wi-Fi

Abbreviazione di Wireless Fidelity, è il nome comune attribuito agli apparecchi che si collegano mediante il protocollo IEEE 802.11. Il meccanismo di collegamento è analogo a quello che si ha con i normali provider via cavo: il provider fornisce un punto di accesso (**access point**) agli utenti, che trasmette e riceve via radio i pacchetti dai clienti.

Creare un punto di accesso WiFi è relativamente semplice ed a basso costo, tanto che è possibile crearlo anche a livello domestico utilizzando i cosiddetti "router wireless" che si collegano via cavo alla connessione di rete (ad esempio ADSL) e possono far fruire così del collegamento Internet tutti gli apparecchi di casa, senza l'uso di ulteriori cavi.

Le frequenze di trasmissione sono quelle delle microonde (2,4-2,5 GHz) e le distanze entro cui il segnale inviato dalle antenne del punto di accesso è utilizzabile dai clienti sono dell'ordine del centinaio di metri, la velocità di trasmissione dell'ordine delle decine di Mb/s, quindi si parla di "banda larga".

Punti di accesso WiFi gestiti da enti pubblici o società private sono spesso presenti in aeroporti, stazioni, scuole, molti comuni hanno intrapreso recentemente iniziative per creare reti WiFi ad accesso pubblico installando vari punti di accesso nei centri abitati.

L'accesso può essere libero e il segnale non criptato (con problematiche di riservatezza) ma esistono comunque protocolli per la cifratura e il controllo dell'accesso, che si possono utilizzare anche per le reti create a livello domestico. Un limite all'uso di questo protocollo è legato alla scarsa adattabilità all'uso in movimento.

UMTS, GSM

I protocolli di trasmissione radio utilizzati dalla telefonia cellulare possono anch'essi essere utilizzati per la trasmissione dati e quindi per collegare computer e smartphone ad Internet. I gestori delle reti quindi, propongono oggi sistemi per l'accesso ad Internet basati sull'uso diretto del telefono o su dispositivi (le cosiddette "Internet Key") da collegare al PC. La tecnologia più recente ed efficiente è legata al protocollo **High Speed Downlink Packet Access (HDSPA)** che permette il download a velocità teoriche (7.2Mb/s) simili a quelle del collegamento ADSL usando la rete telefonica UMTS.

---

1 *Si noti che l'utilizzo del termine "hacker" per indicare i pirati informatici è errato: quest'ultimo termine si intendono persone che cercano di superare qualche barriera legata a difficoltà tecniche senza alcun fine malevolo.*

In assenza di segnale o telefono/scheda UMTS, si possono anche, con prestazioni inferiori, usare le vecchie tecnologie GSM/GPRS/EDGE.

Utilizzando questo collegamento non si è dipendenti dalla presenza di Access Point e non si è limitati nella mobilità ovviamente se si rimane nelle zone coperte dal segnale da parte delle compagnie telefoniche. I costi sono recentemente calati in modo tale da rendere concorrenziale questo tipo di connessione con gli abbonamenti ADSL o l'uso delle connessioni Wi-Fi a pagamento.

WiMax

WiMax (Worldwide Interoperability for Microwave Access), è una recente tecnologia per connessioni wireless che dovrebbe portare molti vantaggi come l'ampio raggio di copertura di un punto di accesso (teoricamente decine di chilometri), la capacità di funzionare anche in condizioni di ostruzione (es. presenza di monti), la capacità di funzionare anche in movimento (fino a 120 Km/h). Le prestazioni reali, però, potrebbero essere molto inferiori a quelle previste. In ogni caso, sono state in Italia assegnate per tale servizio le frequenze libere (banda 3.4-3.6 Ghz) con un'asta tenutasi nel Febbraio 2008, e sono oggi disponibili sul mercato i primi WISP (Wireless Internet Service Providers) che utilizzano tale tecnologia.

2.3 Servizi di Internet

Internet viene principalmente utilizzata per fornire e fruire di servizi di vario tipo. Una volta connessi alla rete, infatti, possiamo accedere a computer che forniscono particolari tipi di servizi per il pubblico.

Un servizio di Internet è un'architettura software (cioè un insieme di programmi che collaborano tra loro per svolgere un determinato compito) che utilizza per la gestione e l'ordinamento dei pacchetti il protocollo IP. In generale, esistono due architetture di base per i servizi di rete: client/server e peer to peer.

Architettura client/server

Un processo servente (**server**, o nella terminologia più moderna **daemon**) offre un servizio ai processi clienti (**client**, o nella terminologia più moderna **agent**), che sono i soli abilitati ad iniziare la comunicazione.

Un processo servente:

- aspetta richieste da un processo cliente;
- può servire più richieste allo stesso tempo;
- applica una politica di priorità tra le richieste;
- può attivare altri processi di servizio;
- è più robusto e affidabile dei clienti;
- tiene traccia storica dei collegamenti (mediante i logfile).

L'esempio delle pagine web, che rivedremo in seguito, può aiutare a comprendere meglio. Supponiamo che un utente, tramite il suo Browser (il client del web, cioè un programma come Microsoft Internet Explorer, Apple Safari, Mozilla Firefox, ecc), voglia leggere il contenuto della pagina principale del sito *http://elvira.univr.it*, cioè index.html.

Sula macchina che corrisponde al dominio *elvira.univr.it* (che ha indirizzo IP 157.27.10.55) è

sempre attivo un programma servente che si chiama httpd (HTTP Daemon). Il cliente (ad esempio Internet Explorer) invia la richiesta "GET http://elvira.univr.it/index.html" ed il server HTTP cercherà il documento (file) corrispondente e lo invierà al cliente. Notare che il server non fa nulla finché non arriva una richiesta del client.

Il server HTTP (httpd) è sempre in ascolto e può servire più client contemporaneamente (cioè più utenti possono fare richieste contemporanee). Se richiesto da opportune istruzioni contenute nel documento (file) al quale si vuole accedere, httpd può attivare altri processi (es. il programma che conta il numero di accessi). Infine httpd registra ogni richiesta (il nome della macchina chiamante, non l'identificativo dell'utente) in un opportuno file (logfile) che serve a fini statistici o per controllare il verificarsi di errori che impediscano agli utenti una corretta lettura dei documenti.

Architettura peer-to-peer

L'architettura client-server è tipica di quasi tutti i servizi che possiamo fornire o usare su Internet, ma esistono alcune eccezioni, come i cosiddetti servizi **peer to peer**.

Si dice peer to peer una rete in cui i nodi sono tutti equivalenti. In tale rete due applicazioni su due computer diversi possono indifferentemente iniziare una comunicazione, a differenza del caso client-server, in cui il cliente fa sempre la richiesta. Questo avviene, per esempio, nei ben noti applicativi di file sharing, anche se i servizi sono, in realtà, spesso **ibridi** e usano spesso un server per la ricerca dei nodi contenenti i file. Un esempio classico di scenario di utilizzo di questi servizi è quello dell'utente che vuole scaricare dalla rete delle canzoni. Dopo essersi registrato on-line in un sito opportuno (pagando l'eventuale abbonamento scelto), egli ha l'accesso a dei cataloghi mediante un programma fornitogli con l'abbonamento. Seleziona il nome del cantante e le canzoni che gli interessano dal menu interattivo. A questo punto, il processo iniziato dall'utente termina e viene avviata una procedure per la quale in qualunque momento un nodo che contiene i dati di interesse può iniziare autonomamente la trasmissione alla macchina dell'utente. Questa è una comunicazione peer-to-peer, poiché le macchine che partecipano a questa distribuzione di dati sono paritarie: ognuna è in grado di iniziare una trasmissione di dati (anche quella dell'utente che abbiamo descritto che potrà a sua volta se necessario inviare i file ricevuti ad altri utenti).

Un processo peer-to-peer:

- richiede una procedura di registrazione dell'utente che vuole accedere al servizio;
- può chiedere informazioni a server specifici (es. cataloghi di prodotti);
- mette a disposizione sulla rete dei file dichiarati pubblici (cioè leggibili e scaricabili);
- può servire più richieste allo stesso tempo;
- non è particolarmente robusto e affidabile;
- di solito NON tiene traccia storica dei collegamenti.

Si noti che anche altri servizi “classici” di Internet, come ad esempio il servizio di distribuzione delle news (protocollo NNTP, vedi dopo) sono organizzati in modo peer to peer. I servizi e relativi protocolli per lo scambio di file (es. bitTorrent) hanno avuto grande successo in quanto la distribuzione del carico relativo alla trasmissione dei file in una serie di connessioni indipendenti con molti nodi invece che utilizzando un unico server permette di sfruttare molto più efficacemente la banda complessiva disponibile nella rete (in pratica di “scaricare” molto

più velocemente i file). Si noti che l'uso degli applicativi di file sharing e dei protocolli di condivisione di file p2p non è illegale, ma potrebbe esserlo rendere disponibile materiale non libero su tali sistemi.

Servizi sincroni ed asincroni

Un servizio di Internet può essere:

- Sincrono, cioè richiedere una tempistica determinata di trasmissione/ricezione
- Asincrono, ovvero non sincrono, in cui trasmissione e fruizione non hanno vincoli di tempo

Esempi di servizi di Internet sono:

- Servizi di tracciamento: verifica dell'esistenza e connessione di un account o host su Internet dato l'indirizzo e del percorso di rete per raggiungerlo
- Servizi di comunicazione: per scambio messaggi, flussi di dati o programmi fra due o più corrispondenti
- Servizi di cooperazione: condivisione e modifica di risorse (dati, programmi, documenti) fra più corrispondenti
- Servizi di coordinazione: attività concordata di persone, servizi o programmi

Nella tabella Tabella 1 sono riportati i più noti servizi di tracciamento comunicazione, cooperazione e coordinazione, sincroni ed asincroni. Per ciascun tipo di servizi possono esistere differenti protocolli con cui li si possono realizzare.

Servizi	Asincroni	Sincroni
Tracciamento	Finger	Ping
Comunicazione	e-mail, news	Chat, streaming audio/video, VoIP (telefonia)
Cooperazione Coordinazione	Ftp, Telnet, Gopher, World Wide Web	File sharing, Multi user games

Tabella 1: Servizi di Internet sincroni ed asincroni

Uniform Resource Locator

Un aspetto particolare del funzionamento di molti dei servizi di Internet è la tecnica di indirizzamento dei documenti, ovvero il modo in cui è possibile far riferimento ad un determinato documento tra tutti quelli che sono pubblicati sulla rete.

La soluzione che è stata adottata per far fronte a questa importante esigenza si chiama **Uniform Resource Locator (URL),** a volte detto anche Uniform Resource Identifier (URI). L' URL di un documento corrisponde in sostanza al suo indirizzo in rete; ogni risorsa informativa (computer o file) presente su Internet viene rintracciata e raggiunta dai nostri programmi client attraverso la sua URL. Prima della introduzione di questa tecnica non esisteva alcun modo per indicare formalmente dove fosse una certa risorsa informativa su Internet.

Una URL ha una sintassi molto semplice, che nella sua forma normale si compone di tre parti:

```
protocollo://nomehost/nomefile
```

La prima parte indica con una parola chiave il tipo di server (o meglio il protocollo di comunicazione cui il server risponde) a cui si punta (può trattarsi di un server di un server

HTTP, di un server FTP, SCP, gopher, e così via); la seconda indica il nome simbolico dell'host su cui si trova il file indirizzato; al posto del nome può essere fornito l'indirizzo numerico; la terza indica nome e posizione ('path') del singolo documento o file a cui ci si riferisce. Tra la prima e la seconda parte vanno inseriti i caratteri '://'.

Un esempio di URL è il seguente:

```
http://www.liberliber.it/index.html
```

La parola chiave 'http' segnala che ci si riferisce ad un server web (protocollo HTTP), che si trova sul computer denominato 'www.liberliber.it', dal quale vogliamo che ci venga inviato il file in formato HTML il cui nome è 'index.html'.

Mutando le sigle è possibile fare riferimento anche ad altri tipi di servizi di rete Internet, ad esempio:

- 'ftp' per i server FTP;
- 'gopher' per i server gopher;
- 'telnet' per i server telnet.

Occorre notare che questa sintassi può essere utilizzata sia nelle istruzioni ipertestuali dei file HTML, sia con i comandi che i singoli client, ciascuno a suo modo, mettono a disposizione per raggiungere un particolare server o documento. È bene pertanto che anche il normale utente della rete Internet impari a servirsene correttamente.

2.4 Esempi di servizi: la posta elettronica

Il servizio di **posta elettronica** (e-mail) permette la comunicazione asincrona uno-a-uno (cioè un utente con un utente) o uno-a-molti (cioè un utente con molti utenti). Quando un utente desidera inviare una lettera (e-mail) deve conoscere l'indirizzo (e-mail address) del destinatario. Un indirizzo di posta elettronica di Internet ha la forma: **nome@indirizzo-dominio-di-intenet**, per esempio andrea.giachetti@univr.it.

Di solito gli Internet Service Provider forniscono anche ai loro utenti un indirizzo di e-mail e, conseguentemente, uno spazio su cui vengono salvati i messaggi, il servizio che permette di scaricarli e visualizzarli e quello che permette di inviarli. E' anche possibile richiedere gratuitamente un indirizzo ad altri fornitori utilizzando i loro siti web (es. gmail, hotmail, eccetera). Questi servizi permettono, in genere, di accede alla propria casella di posta direttamente tramite il browser web (in tal caso si parla di servizio di **webmail**).

Per accedere e gestire la propria casella di posta, però, normalmente si utilizza un programma client specifico (detto **client di posta elettronica** o Mail User Agent, come ad esempio Mozilla Thunderbird o Microsoft Outlook), opportunamente configurato per accedere al server della posta del provider scelto. Il meccanismo di gestione del servizio non è semplicissimo, dato che implica l'utilizzo di differenti programmi e protocolli; uno schema semplificato è rappresentato in Figura 7. Riassumiamo di seguito i punti principali di preparazione, trasferimento e fruizione dei messaggi:

- il programma client di posta viene usato dall'utente nella composizione del messaggio; il programma riempie alcuni campi in modo automatico (indirizzo del mittente, data, reply-to, riservatezza, priorità, durata, ecc.), aggiungendo eventuali allegati e lo predispone per

l'invio.

- Un altro programma, detto Message Transfer Agent (MTA) provvede al trasporto del messaggio, dialogando via rete con un MTA remoto; più MTA possono essere coinvolti nel trasporto di un messaggio. L'MTA è un daemon (cioè un programma server, normalmente denominato sendmail) in esecuzione su una macchina che viene chiamata server della posta (quello il cui indirizzo è fornito dall'Internet Provider al momento della sottoscrizione dell'abbonamento). Il daemon MTA legge il campo indirizzo del destinatario e consegna subito il messaggio se il destinatario è locale. Altrimenti dopo aver usato il DNS per convertire l'indirizzo di email del destinatario nell'indirizzo IP del server della posta del destinatario ed invia il messaggio al destinatario utilizzando un protocollo di trasmissione detto **SMTP (Simple Mail Transfer Protocol).** Per completare questa operazione il server della posta del mittente stabilisce una connessione TCP con il server della posta del destinatario. Su quest'ultima macchina c'è in ascolto il daemon MTA che "parla" il protocollo SMTP ed è in grado di accettare le connessioni in arrivo e copiare i messaggi nelle caselle postali destinatarie. Se non è possibile spedire un messaggio, al mittente viene restituita una notifica di errore contenente la prima parte del messaggio non spedito. Dopo aver stabilito la connessione il mittente opera come un cliente e attende la risposta del server destinatario. Il server inizia spedendo una linea di testo che lo identifica e dice se è pronto o no a ricevere la posta. Se non lo è, il cliente rilascia la connessione e riproverà più tardi. Se il server può accettare la posta il cliente annuncia mittente e destinatario del messaggio. Se il destinatario è noto, il server dice al cliente di proseguire nell'invio. Il cliente quindi invia il messaggio e il server ne dà conferma.
- Per la lettura della posta esistono due metodi: **offline**, connettendosi al proprio server e scaricando sul proprio PC i messaggi che possono poi essere visualizzati con calma senza essere connessi e **online,** lasciando i messaggi sul server in modo tale da poterli visualizzare su qualunque PC purché connesso in rete. Un protocollo che si usa nel primo caso è il **POP3** (Post Office Protocol – RFC 1225). Ha comandi per permettere all'utente di connettersi, disconnettersi, recuperare messaggi e cancellarli. Lo scopo di POP3 è recuperare la posta dalla casella remota e memorizzarla nella macchina locale dell'utente per leggerla e gestirla fuori linea (offline), cioè senza il collegamento ad Internet attivo. Per la gestione della posta online si usa in genere un differente protocollo detto **IMAP** (Interactive Mail Access Protocol-RFC 1064). In questo caso il mail server conserva un deposito centrale accessibile da qualsiasi macchina cliente. A differenza di POP3, IMAP non copia la posta sulla macchina personale dell'utente, cosa che può essere appunto utile se quest'ultimo utilizza computer differenti, la gestione è quindi online (più costosa), ma più sicura (non vengono lasciate copie in giro delle proprie e-mail). Ovviamente in questo modo non si possono leggere i messaggi in assenza di connessione, anche se alcuni client creano una copia locale degli ultimi messaggi per rendere

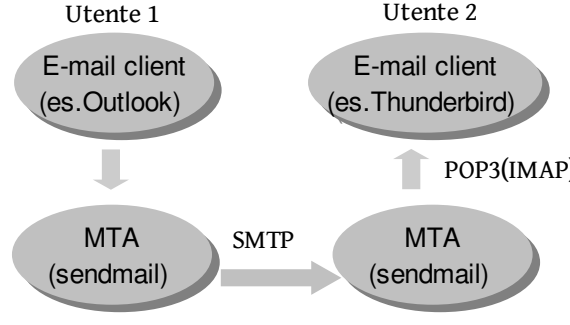


Figura 7: Creazione e trasferimento di un messaggio di posta elettronica

meno problematico questo aspetto.

Per come è stato progettato il servizio di posta elettronica non garantisce alcun sistema di controllo sull'avvenuto ricevimento delle e-mail e non esiste ovviamente neppure la possibilità di garantire dei tempi di consegna. Ciò significa che **non** è possibile essere certi che una nostra e-mail sia stata ricevuta dal destinatario. Basta che un server della posta si blocchi ed essa non viene consegnata. Il mittente non viene avvisato di questo evento e quindi non può sapere se deve ritrasmettere il messaggio oppure no. Per ovviare in parte a questo problema, alcuni client di posta consentono di attivare un'opzione sull'e-mail che sta per essere inviata che farà aprire una finestra di dialogo al ricevente per chiedergli se voglia o meno confermare l'avvenuta ricezione (viene chiamata spesso ricevuta di consegna). L'utente finale può, tuttavia, anche decidere di non confermare (è una sua libera scelta) vanificando così la volontà del mittente, oppure per gli stessi problemi di affidabilità già spiegati, può succedere che la conferma non raggiunga mai il mittente. Il sistema di posta elettronica è, dunque, inaffidabile (per utilizzare la posta elettronica per attività che richiedano totale sicurezza e affidabilità si cercano infatti di utilizzare strumenti resi affidabili, come la **posta elettronica certificata** o PEC).

I programmi che comunemente utilizziamo permettono, come abbiamo detto, di comporre in maniera assistita i messaggi e di collegarsi ai server via POP3 o IMAP per la consultazione dei messaggi, che possono essere scaricati, ordinati e catalogati. I programmi consentono anche di gestire gli indirizzi e anche di raggruppare sotto un unico indirizzo, un insieme (lista) di indirizzi corrispondenti ad altrettanti utenti. Quando una persona invierà un messaggio a tale indirizzo esso verrà inviato a tutti gli utenti della lista. In questo caso si dice che la comunicazione avviene tramite **mailing list** (lista di utenti della posta). Nelle versioni recenti di questi client esiste la possibilità di filtrare i messaggi ricevuti per scartare quelli indesiderati (**spam**). I programmi spesso includono la possibilità di essere utilizzati come client di altri servizi, come i newsgroup o i web feed (vedi cap. 3.8).

✔ *Nota: lo spam*

Probabilmente il concetto di “spam” come posta elettronica indesiderata (detta anche “junk mail”) è ormai noto a quasi tutti, mentre meno noto è il fatto che il nome dato a questo fenomeno deriva da uno sketch comico del gruppo inglese “Monty Python” in cui un cameriere cercava di fornire insistentemente a degli infastiditi clienti piatti contenenti un tipo di carne in scatola con quel nome (e veramente esistente). Il fenomeno è molto vasto (si stima che i due terzi della posta elettronica gestita dai server sia di questo tipo) ed è in genere creato con lo scopo di adescare potenziali vittime di truffe o diffondere programmi malevoli. Oggi sia i client di posta elettronica che i provider, filtrano i messaggi in arrivo in modo da eliminare buona parte di questi messaggi, con il rischio tuttavia di far perdere posta realmente utile ai clienti. In ogni caso bastano pochi accorgimenti per limitare i potenziali danni derivanti da tale pratica:

- non rendere pubblico se non necessario il proprio indirizzo email (scrivendolo quindi su pagine web, messaggi in chiaro, bacheche)
- non visualizzare sul client messaggi HTML (cioè che possano far eseguire codice al proprio PC) provenienti da indirizzi sconosciuti. Controllare bene intestazioni e indirizzi degli eventuali link del messaggio per controllare che siano esattamente quelli dei propri contatti (o della propria banca, ecc.)

- non aprire allegati provenienti da indirizzi non sicuri
- non rispondere ad offerte di prodotti a condizioni eccezionali, appelli a far circolare storie per raccogliere fondi e cose simili se non provengono da persone note e di fiducia

Usando solo questi semplici accorgimenti, gli unici danni che può provenire dalla posta elettronica indesiderata sono quelli relativi al tempo necessario alla cancellazione di messaggi e alla (rara) perdita di qualche messaggio utile a causa dell'uso dei filtri.

2.5 Esempi di servizi: i newsgroup

Un **newsgroup** è una bacheca elettronica su cui è possibile inserire un proprio messaggio, analogo all'e-mail, che verrà poi letto da tutti i fruitori del servizio. I newsgroup sono divisi in canali tematici e sono un'utile fonte di informazioni.

Ad esempio, se un utente vuole avere una precisazione su una legge può scrivere a it.diritto. Chi legge quel messaggio affisso in bacheca può decidere di rispondere e fornire delle delucidazioni. Oppure può aprirsi un confronto/dibattito tra più utenti su un argomento (che dovrebbe essere sempre inerente al newsgroup, i cosiddetti messaggi "off topic", ossia fuori argomento, non sono graditi ai frequentatori).

Per accedere al servizio non occorre iscriversi. Alcuni newsgroup possono essere moderati, cioè i messaggi, prima di essere inseriti nella bacheca, passano il vaglio di un censore che controlla che siano attinenti all'argomento del newsgroup e che non siano ingiuriosi o illegali. Ci sono newsgroup per tutti i generi e gusti, per fare qualche esempio: alt.alien.visitors, it.hobby.umorismo, it.tlc.cellulari, comp.os.ms-windows, it.comp.hardware.cd, it.discussioni.sentimenti, e così via. La gerarchia .it è gestita da www.news.nic.it.

Per accedere ad un newsgroup basta usare il programma client della posta (occorre configurare il programma indicando il news server a cui collegarsi; tale informazione è normalmente fornita dal provider presso il quale si è sottoscritto un abbonamento ad Internet). Esistono anche siti Web che tengono attivi dei link ai newsgroup e che consentono la lettura e scrittura di messaggi sui newsgroup. Il servizio di newsgroup ha avuto una buona popolarità tra gli utilizzatori di Internet negli anni passati. Oggi, tuttavia, i newsgroup tendono ad essere rimpiazzati da servizi web come forum o social network specializzati (vedi in seguito), anche se mantengono un pubblico affezionato tra gli esperti dei vari settori.

2.6 Il servizio per eccellenza: il World Wide Web

World Wide Web (cui ci si riferisce spesso con gli acronimi **WWW** o **W3**) è sicuramente il servizio di maggiore successo su Internet. Il successo del Web è stato tale che attualmente, per la maggior parte degli utenti, esso coincide con la rete stessa.

Sebbene questa convinzione sia tecnicamente scorretta, è indubbio che gran parte dell'esplosione del "fenomeno Internet" a cui si è assistito in questi ultimi anni sia legata proprio alla diffusione di questo servizio e che, nel linguaggio comune, l'errata equivalenza tra Internet e Web sia comunemente utilizzata. Il Web, tuttavia, non è altro che un sistema di distribuzione di contenuti ipertestuali basato su uno specifico protocollo (HTTP, HyperText Transfer Protocol) e realizzato attraverso particolari programmi server per la distribuzione di contenuti e client per la loro visualizzazione (i browser web). La storia del World Wide Web ha avuto inizio nel maggio del 1990, quando Tim Berners-Lee, un ricercatore del CERN di Ginevra (ove

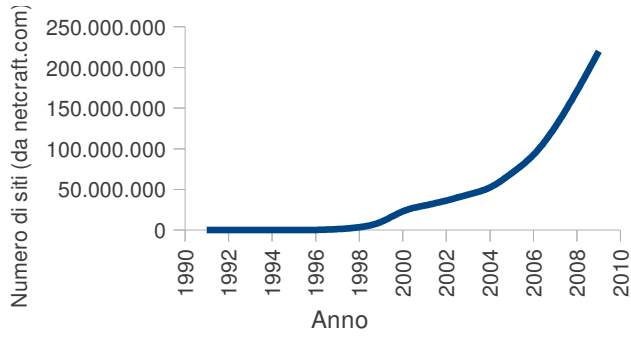


Figura 8: Crescita del numero di siti web dalla nascita ad oggi

si studia principalmente la fisica delle particelle) presentò ai dirigenti dei laboratori una relazione intitolata "Information Management: a Proposal". La proposta di Berners-Lee aveva l'obiettivo di sviluppare un sistema di pubblicazione e reperimento dell'informazione distribuito su rete geografica che tenesse in contatto la comunità internazionale dei fisici e racchiudeva le idee base del web che conosciamo oggi. Nell'ottobre di quello stesso anno iniziarono le prime sperimentazioni. Il World Wide Web che iniziò a svilupparsi all'epoca era, però, uno strumento per pochi. L'impulso decisivo al suo sviluppo arrivò solo agli inizi del 1993, presso il National Center for Supercomputing Applications (NCSA) dell'Università dell'Illinois.

Basandosi sul lavoro del CERN, Marc Andreessen (poi co-fondatore con Jim Clark della Netscape Communication) ed Eric Bina svilupparono un'interfaccia grafica multipiattaforma per l'accesso ai documenti presenti su World Wide Web: **Mosaic,** cioè il primo **browser web**, distribuendola gratuitamente a tutta la comunità di utenti della rete. Come per tutta una serie di fortunate applicazioni di pubblico dominio o di software libero successive (vedi Appendice 1), l'esistenza di un interfaccia di facile uso e reperibile gratuitamente rappresentò il punto di partenza per rendere il sistema World Wide Web interessante per un'enorme quantità di nuovi autori ed editori telematici.

Questo interesse determinò i ritmi di crescita più che esponenziali del servizio negli anni 90 del secolo scorso (vedi Figura 8). Nel 1993 esistevano solo duecento server web: oggi ce ne sono centinaia di milioni.

Se inizialmente World Wide Web era uno strumento di pubblicazione prevalentemente scientifico ove si trovavano le pagine di centri di ricerca universitari (che informano sulle proprie attività e mettono a disposizione pubblicazioni, dati ecc.) e quelle dei grandi enti che gestivano la rete (con le ultime notizie su protocolli e specifiche di comunicazione, le ultime versioni dei software per l'accesso alla rete o per la gestione di servizi, ecc.), in breve tempo tutti cominciarono ad utilizzare il mezzo per diffondere il proprio materiale. Via via cominciarono così ad apparire sui server web contenuti di ogni genere: riviste letterarie, gallerie d'arte telematiche, musei virtuali con immagini digitalizzate dei quadri, biblioteche con le riproduzioni di rari manoscritti, servizi meteorologici con immagini in tempo reale provenienti dai satelliti, fototeche, notizie di borsa aggiornate in tempo reale e integrate da grafici, negozi di ogni tipo, e così via. Oggi sarebbe probabilmente più facile elencare le tipologie di dati o servizi che non sono accessibili in rete piuttosto che quelle che vi si possono trovare.

Descriveremo meglio nel capitolo 3 le tipologie principali dei “siti” web oggigiorno accessibili

al pubblico di Internet, classificati sulla base del tipo gestione, pubblico e servizi offerti.

Le caratteristiche che hanno reso il World Wide Web una vera e propria rivoluzione nel mondo della telematica possono essere riassunte nei seguenti punti:

- la sua diffusione planetaria;
- la facilità di utilizzazione delle interfacce;
- la sua organizzazione ipertestuale;
- la possibilità di trasmettere/ricevere informazioni multimediali integrate con il documento;
- la semplicità di gestione per i fornitori di informazione.

Dal punto di vista dell'utente, il web si presenta come un illimitato universo di documenti multimediali integrati ed interconnessi tramite una rete di collegamenti dinamici. Uno spazio informativo in cui è possibile muoversi facilmente alla ricerca di informazioni, testi, immagini, dati, curiosità, prodotti. Non solo: fin da subito i browser web si sono caratterizzati per essere programmi molto flessibili in grado di fare più del semplice rendering (rappresentazione grafica) delle pagine ipertestuali.

Essi infatti consentono di accedere in maniera semplice a molte altre risorse e servizi presenti su Internet, anche se gestite con altri protocolli come FTP e oggi sono in grado di integrare client di nuovi servizi come streaming audio e video, feed RSS, sistemi di messaggistica e molto altro.

Dal punto di vista dei fornitori di informazione il web è uno strumento per la diffusione telematica di documenti elettronici multimediali decisamente semplice da utilizzare, poco costoso e dotato del canale di distribuzione più vasto e ramificato del mondo.

Tra i diversi aspetti innovativi del World Wide Web i più notevoli sono decisamente l'organizzazione ipertestuale e la possibilità di trasmettere informazioni multimediali. Cerchiamo di definire queste proprietà in modo preciso.

2.7 Ipertesti e multimedialità

Da diversi anni queste due parole, uscite dal ristretto ambiente specialistico degli informatici, ricorrono sempre più spesso negli ambiti più disparati, dalla pubblicistica specializzata fino alle pagine culturali dei quotidiani. Purtroppo questo non basta ad evitare la generale diffusa confusione che sussiste sul loro significato.

In primo luogo è bene distinguere il concetto di **multimedialità** da quello di **ipertesto**. Mentre il primo si riferisce ai mezzi diversi con cui può avvenire una comunicazione, il secondo riguarda la sfera più complessa dell'organizzazione dell'informazione.

Un documento si dice multimediale se può contenere più mezzi (media) di comunicazione differenti (video, audio, immagini, ecc.).

Una pagina di un libro con delle figure è un documento multimediale (perché oltre allo scritto è presente un altro media, cioè le immagini). Da notare che spesso vengono considerati **media** diversi anche differenti stili nell'uso di un medesimo mezzo fisico, per esempio l'immagine fotografica e il disegno, la poesia e il racconto.

Anche se multimediale in quanto contenente testo di tipo diverso ed immagini, un libro può essere letto solo sequenzialmente (pagina dopo pagina, dalla prima all'ultima). L'utente deve tenere conto mentalmente dei legami logici che esistono tra le varie parti del libro con il rischio di dimenticare e perdere il significato del tutto (per esempio, leggendo questo libro, potreste trovare ad un certo punto in riferimento al sistema DNS che non riuscite a capire perché la

definizione era scritta molte pagine indietro). Grazie all'avvento dei calcolatori è stato possibile creare dei documenti organizzati in modo completamente differente, ovvero gli ipertesti. Un **ipertesto** è un documento che può essere letto in modo **non lineare**, cioè il lettore può leggerlo saltando da un'informazione all'altra, in modo relativamente libero. Se progettato bene, un ipertesto consente una chiave di lettura personale del documento (aiutando la memorizzazione dei legami logici), e apre al lettore la possibilità di scoprire nuovi legami tra le varie informazioni contenute. Nell'esempio fatto prima, un collegamento dal testo che parlava di DNS alla pagina dove questo viene descritto permetterebbe all'utente di rispondere ai propri dubbi senza perdere tempo in ricerche complesse.

Un ipertesto si basa, quindi, su un'organizzazione **reticolare** dell'informazione, ed è costituito da un insieme di unità informative (i nodi) e da un insieme di collegamenti (detti, nel gergo tecnico, **link**) che da un nodo permettono di passare ad uno o più nodi. Se le informazioni che sono collegate tra loro nella rete non sono solo documenti testuali, ma in generale informazioni veicolate da media differenti (testi, immagini, suoni, video), l'ipertesto diventa multimediale, e viene definito a volte **ipermedia**. Una idea intuitiva di cosa sia un ipertesto multimediale può essere ricavata dalla Figura 9: i documenti, l'immagine il suono e il filmato sono i nodi dell'ipertesto, mentre le linee rappresentano i collegamenti (link) tra i vari nodi. Il documento in alto, ad esempio, contiene tre link, da dove è possibile saltare ad altri documenti o alla sequenza video. Il lettore, dunque, non è vincolato dalla sequenza lineare dei contenuti di un certo documento, ma può muoversi da una unità testuale ad un'altra (o ad un blocco di informazioni veicolato da un altro medium), costruendosi ogni volta un proprio percorso di lettura. Naturalmente i vari collegamenti devono essere collocati in punti in cui il riferimento ad altre informazioni sia **semanticamente rilevante**: per fornire un approfondimento o proseguire il percorso informativo personalizzato. In caso contrario si rischia di rendere inconsistente l'intera base informativa, o di far smarrire il lettore in peregrinazioni prive di senso.

Dal punto di vista della implementazione concreta, un ipertesto digitale si presenta come un documento elettronico in cui alcune porzioni di testo o immagini presenti sullo schermo, evidenziate attraverso artifici grafici (icone, colore, tipo e stile del carattere), rappresentano i diversi collegamenti disponibili nella pagina. Questi funzionano come dei pulsanti che attivano il collegamento e consentono di passare, sullo schermo, al documento di destinazione. Il pulsante viene premuto attraverso un dispositivo di input, generalmente il mouse o una combinazioni di tasti, o un tocco su uno schermo a sfioramento.

L'incontro tra ipertesto, multimedialità e interattività rappresenta dunque la nuova frontiera delle tecnologie comunicative. Il problema della comprensione teorica e del pieno sfruttamento delle enormi potenzialità di tali strumenti, specialmente in campo didattico, pedagogico e divulgativo è naturalmente ancora in gran parte aperto: si tratta di un settore nel quale vi sono state negli ultimi anni innovazioni di notevole portata ed in cui è lecito aspettarsene altrettante nel prossimo futuro.

Il World Wide Web è stato una di queste innovazioni, probabilmente quella di maggiore successo. WWW è infatti un sistema ipermediale con la particolarità che i diversi nodi della rete ipertestuale sono distribuiti sui vari computer (host) collegati tra loro tramite la rete Internet. Attivando un singolo link si può dunque passare da un documento ad un altro documento che si trova archiviato fisicamente su un altro computer della rete.

Il funzionamento di World Wide Web non differisce molto da quello delle altre applicazioni Internet. Anche in questo caso il sistema si basa su una interazione tra un client ed un server. Il protocollo di comunicazione che i due moduli utilizzano per interagire si chiama **HyperText**

Transfer Protocol (HTTP). L'unica, ma importante, differenza specifica è la presenza di un formato speciale in cui debbono essere memorizzati i documenti inseriti su Web, denominato **HyperText Markup Language (HTML).**

I client web sono gli strumenti di interfaccia tra l'utente ed il sistema; le funzioni principali che svolgono sono:

- ricevere i comandi dell'utente;
- richiedere ai server i documenti;
- interpretare il formato e presentarlo all'utente.

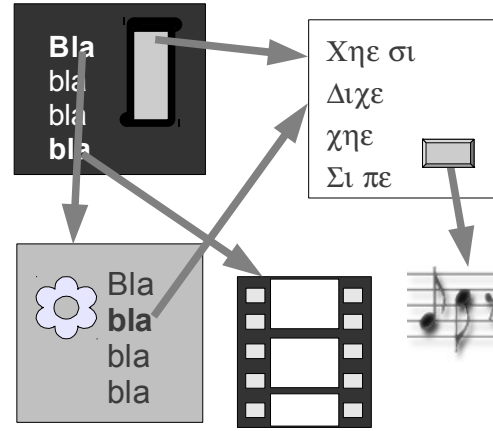


Figura 9: Schema logico di ipertesto multimediale

Nel gergo telematico questi programmi vengono chiamati anche **browser**, dall'inglese to browse, sfogliare, poiché essi permettono appunto di scorrere i documenti. Ne descriveremo in dettaglio le caratteristiche nel capitolo 3.

Nel momento in cui l'utente attiva un collegamento, agendo su un link o specificando esplicitamente l'indirizzo di un documento, il client invia una richiesta ad un determinato server con l'indicazione del file che deve ricevere.

Il server web, o server HTTP, per contro, si occupa della gestione, del reperimento e del recapito dei singoli documenti richiesti dai client. Naturalmente esso è in grado di servire più richieste contemporaneamente. Ma un server può svolgere anche altre funzioni. Una tipica mansione dei server HTTP è l'interazione con altri programmi, interazione che permette di modificare documenti in modo dinamico (ad esempio, la pagina web di un giornale è aggiornata da uno speciale programma che inserisce gli articoli mano a mano, durante la giornata, arrivano le notizie in redazione). Anche i dettagli di questi meccanismi saranno descritti nel capitolo 3.

2.8 Altri servizi che usano il protocollo IP

Che Internet non coincida comunque con il solo web sta diventando sempre più chiaro dato che sopra i protocolli di Internet si stanno oggi lanciando molti nuovi servizi che sfruttano la banda larga disponibile grazie alle connessioni ADSL e UMTS per riuscire ad ottenere, utilizzando quindi le stesse modalità di distribuzione dei pacchetti di informazione agli indirizzi degli host visti finora, servizi interattivi audio/video come la televisione on demand (IPTV) la telefonia (VOIP, Voice over IP) o varie forme di informazione/intrattenimento (giochi multiutente, sistemi informativi geografici, mappe online, ecc.).

Skype è un client per la telefonia via Internet ormai largamente diffuso e tutti i principali fornitori di servizi ADSL offrono tra i loro servizi anche il broadcasting televisivo. Si presuppone ormai una prossima convergenza su Internet di tutti i canali di comunicazione, informazione ed intrattenimento, cosa che, secondo alcuni critici, potrebbe causare seri problemi alla funzionalità della rete.

Inoltre, come abbiamo visto, molti di questi servizi ed anche di quelli comunemente fruiti attraverso il browser web come motori di ricerca, archivi, social network ecc. sono e saranno sempre più accessibili attraverso i vari tipi di computer "alternativi" , smartphone, tv lcd,

tablet, ecc, mediante applicazioni che recuperano i contenuti senza necessariamente passare per l'uso di un browser web.

2.9 Evoluzione del web. Il web 2.0

Per quanto riguarda il web, occorre dire che esso ha subito una rapida evoluzione negli ultimi anni, legata sia all'arricchimento tecnologico, sia al modo di proporre i contenuti.,

Le pagine web che originariamente contenevano solo un testo fisso, immagini e collegamenti ipertestuali, sono diventate "dinamiche" e interattive e consentono all'utente di interagire con parti di essa, avviando video, giocando, comunicando con altri utenti.

Queste possibilità sono realizzate attraverso varie tecnologie: il cosiddetto HTML dinamico (DHTML), con l'uso di programmi interpretati ed eseguiti dal browser stesso, fa sì che l'utente possa modificare interattivamente la pagina che sta consultando (si parla in questo caso di elaborazione "**client side**", cioè dal lato del cliente ossia sul nostro PC). Oltre ai metodi "standard" esistono poi tecnologie particolari (in genere proprietarie), come Flash o Java, che consentono di integrare nelle pagine web contenuti complessi ed interattivi (ad esempio animazioni 2D e 3D, giochi).

D'altro canto, il potenziamento delle reti e dei sistemi di gestione delle basi di dati ha consentito di rendere più complessi i servizi realizzati sui server dei siti. Oggi la stragrande maggioranza dei siti più visitati consente di realizzare attività complesse (es. ricerca, archivio dati, comunicazione, distribuzione di documenti, ecc.) attraverso programmi e basi di dati installate sulle macchine server, cui si fanno richieste attraverso l'interfaccia della pagina web utilizzando elementi che possono essere inseriti in essa come caselle di testo, selettori e pulsanti (si parla in questo caso di elaborazione "**server side**", cioè che avviene sul computer che ospita il sito). Nei capitoli seguenti vedremo dettagli sui servizi e sulle tecnologie che li realizzano.

Al di là dell'evoluzione delle tecnologie, però, è importante notare come anche l'uso del web sia molto cambiato rispetto agli albori. Un pubblico molto più grande ed eterogeneo ha cominciato ad usarlo e sono cambiate anche le modalità di interazione degli utenti con i siti. Ad esempio, sono emerse tipologie particolari di siti che permettono a utenti non specialisti (che non sarebbero cioè in grado di scrivere il codice delle pagine) di inserire contenuti multimediali personalizzati (come i cosiddetti "blog" o i siti per pubblicare in modo facilitato foto o video (es. Flickr, YouTube). Alcune aziende hanno creato piattaforme complete per gestire la comunicazione tramite messaggi di vario tipo entro comunità più o meno ristrette di persone (i cosiddetti "social network").

Altri siti si avvalgono sistemi di creazione di contenuto collaborativi per permettere a gruppi di utenti di creare progetti comuni (emblematico il caso di Wikipedia, un'enciclopedia online creata da centinaia di migliaia di autori indipendenti).

Ci occuperemo di queste tipologie di siti e di tendenze nel capitolo seguente. Ricordiamo ancora soltanto che sui media esse sono spesso collegate al termine "web 2.0", locuzione che in qualche modo vorrebbe sottolineare l'evoluzione del web stesso ad uno stadio successivo a quello iniziale. Questo termine, tuttavia, non rispecchia realmente un cambio improvviso di tecnologie o protocolli usati in rete ed è per questo motivo molto criticato. Forse quello che è maggiormente cambiato nell'uso recente del web rispetto a quello originario è l'approccio dell'utente ai contenuti in rete, un approccio sicuramente più attivo e accessibile ad un pubblico molto più ampio.

3 Funzionamento e gestione dei siti web

Cerchiamo ora di capire in dettaglio che cosa sia effettivamente un sito web. Per **sito** si intende un insieme di pagine e servizi web, in genere ospitate su un unico server, attraverso le quali l'utente può svolgere dei compiti ben definiti (leggere un giornale, comprare un libro, pubblicare le proprie opinioni, ecc.).

Nella sua forma più semplice, un sito è una semplice collezione di pagine ipertestuali collegate tra loro, cioè una collezione di documenti di testo impaginati con possibilità di inserire figure (poi estesa all'integrazione di varie componenti multimediali) e la possibilità di inserire collegamenti per passare non linearmente da una pagina all'altra. Un sito di questo tipo viene oggi di solito chiamato "statico".

Le **pagine** di tale sito sono quindi documenti testuali codificati con il linguaggio **di markup** HTML che permette di inserire insieme al testo la metainformazione utile al browser per impaginare, assegnare stili grafici, creare i collegamenti ed inserire i vari oggetti multimediali (il linguaggio "marca" sezioni di testo come paragrafi, titoli, tabelle, àncore, ecc, dando loro un "significato" che viene interpretato dal browser.).

Navigare sul sito significa visualizzare le pagine una dopo l'altra seguendo una serie di collegamenti o utilizzando le opzioni del browser (per esempio il tasto "indietro").

Cosa succede in rete quando richiediamo una pagina di un sito?

La richiesta di una pagina da parte del programma cliente che utilizziamo per accedere al sito (il browser) viene fatta inviando un messaggio codificato con il protocollo HTTP al programma server in esecuzione su un host collegato a Internet. Questo programma "server" (ad esempio il programma open source Apache) è in grado di interpretare la richiesta del documento e di inviare, sempre tramite HTTP, la risposta contenente il codice del documento stesso.

Ecco cosa succede quando visualizziamo una semplice pagina web (Figura 10). Sull'host che contattiamo c'è il server in ascolto alla porta 80 del protocollo TCP (al di là del dettaglio tecnico, vuol dire che c'è un programma in esecuzione che controlla costantemente se arrivano ad un particolare indirizzo messaggi dalla rete).

Dalla parte del client, quando scriviamo l'indirizzo dell'host da contattare e la pagina richiesta, il browser determina l'URL, cerca l'indirizzo IP corrispondente all'host del server utilizzando il sistema DNS, quando ha la risposta dal DNS, il browser può connettersi al server alla porta 80 ed inviare la richiesta utilizzando il protocollo HTTP (protocollo basato su messaggi testuali simile a SMTP):

```
GET /fondinfo/esami.htm HTTP/1.0
Referer:  http://utenti.tripod.it/fondinfo/index.htm
User-Agent: Mozilla/3.0 (Win98; I)
Host ...
```

Il server identifica il file HTML richiesto, lo recupera nel disco rigido dell'host ed invia al client un messaggio che lo contiene o un messaggio di errore se il documento non era presente. La risposta è quindi sempre un messaggio testuale codificato in HTTP, es:

```
HTTP/1.0/200 OK
content-type: text/html

<html><head>
<title>  Calendario esami </title>
</head>
<body>
<h1>Titolo </h1>…
```

Il messaggio contiene uno "status code", se lo stato è "ok" il codice è 200, se la pagina non è trovata è 404, se c'è un errore dovuto al server, 500. Importante è anche l'indicazione del **content type,** in genere, come nel caso di esempio, è una pagina web e quindi è identificata come text/html.

Questa indicazione del tipo prende il nome di MIME-Type ed identifica la tipologia di documento allegata al messaggio. Il MIME-Type è un meccanismo di definizione delle tipologie di documento multimediale nato per gli allegati della posta elettronica (MIME significa Multipurpose Internet Mail Extension) che è diventato standard universale per Internet. I browser moderni sono in grado di rappresentare direttamente o mediante il ricorso ad applicativi esterni molti tipi di documento e non solo l'HTML. Esempi di MIME-type sono **text/plain** (testo piano), **image/gif** (immagine in formato gif), **audio/mpeg** (audio in formato mpeg), eccetera.

Al termine della comunicazione viene rilasciata la connessione TCP.

Il browser può così creare la visualizzazione del testo, eventualmente recuperando ancora in rete le immagini (o gli altri file inclusi) dagli URL indicati nell'HTML.

Figura 10: Transazioni tra browser e server web per la visualizzazione di un documento

3.1 Caratteristiche di HTTP

Il servizio web è dunque caratterizzato dal protocollo di trasmissione degli ipertesti HTTP. Le caratteristiche fondamentali di HTTP sono le seguenti:

- è un protocollo client-server: i ruoli di richiesta-risposta sono ben definiti
- è generico, cioè non dipende dal formato dei dati trasmessi: via HTTP possiamo trasmettere ipertesti in formato HTML, ma anche altri dati caratterizzati dai diversi MIME-type.

Si noti che ogni coppia di richiesta-risposta è indipendente: HTML **non tiene memoria** delle connessioni precedenti e quindi per realizzare un sistema in cui si debba tenere traccia delle attività precedenti è necessario utilizzare dei trucchi per ovviare a questa caratteristica. Una tecnica utilizzata dai siti per fare in modo che le richieste dell'utente tengano conto di ciò che è stato fatto prima (che però può creare qualche problema di sicurezza), consiste nell'uso dei cosiddetti **web cookies**, cioè piccoli file di traccia delle attività inviati dal server ai client e che vengono memorizzati sui dischi locali degli utenti. Quando un utente si ricollega al medesimo server, questi vengono reinviati ad esso che quindi riceve informazioni sulle attività svolte in precedenza.

3.2 Il browser

Il browser è il programma client del servizio WWW. Esso deve quindi essere in grado di visualizzare gli ipertesti codificati in HTML seguendo le direttive dei vari standard e di inviare le richieste codificate in HTTP ai programmi server. Esistono molti browser web, sviluppati da grandi aziende o da fondazioni (es. Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari, ecc.), che evolvono continuamente aggiungendo nuove funzionalità e migliorando l'esperienza di "navigazione" sul web. L'interfaccia di tali programmi è, comunque, più o meno sempre la stessa. Abbiamo (Figura 11) una finestra con un riquadro principale ove viene fatta la visualizzazione dell'ipertesto e varie barre, ampiamente configurabili dall'utente con menu completi delle opzioni e scorciatoie per l'utilizzo efficiente.

Il browser consente di "**navigare**" sul web, cioè di passare alla visualizzazione della pagina collegata quando si clicca su un'**àncora** (la parte dell'ipertesto attiva che di solito è evidenziata con un colore differente o una cornice) o di accedere alle pagine già visitate con appositi tasti o opzioni di menu.

Tipicamente, oltre alla visualizzazione del contenuto dell'ipertesto nella parte centrale dell'interfaccia, il browser presenta una serie di componenti standard (Figura 11): una casella di testo per inserire o semplicemente mostrare l'indirizzo (URL) della pagina del file visualizzato; un campo di testo in alto a sinistra che contiene il titolo del documento stesso; un altro campo di testo (barra di stato) in basso a sinistra, che visualizza messaggi relativi alla navigazione (per esempio l'indirizzo del link corrispondente all'ancora su cui si trova il cursore); un tasto per tornare alla pagina precedente ed uno per ricaricare (aggiornare) la pagina indicata dall'indirizzo scritto (che può essere stata modificata nel frattempo). Attraverso i menù, poi, è possibile accedere ad opzioni avanzate come stabilire la visualizzazione di default per le varie parti dell'ipertesto, mostrare la cronologia dei siti visitati, gestire il salvataggio dei cosiddetti segnalibri, vedere il codice sorgente della pagina, definire le opzioni di sicurezza, ecc. I browser non sono ovviamente tutti uguali, anche se il modo in cui supportano la navigazione e visualizzano i documenti dovrebbe seguire in maniera rigorosa le specifiche indicate dal W3C. Spesso però i browser hanno introdotto estensioni proprietarie alla codifica HTML poi passate a far parte dello standard, cioè in pratica è stata anche l'evoluzione dei browser a guidare le evoluzioni della codifica dei documenti per il web. I browser web, in genere, non si limitano a visualizzare documenti HTML ed a supportare il protocollo HTTP per trasferire i documenti. Essi sono in grado, per esempio, di utilizzare altri protocolli, come FTP per il trasferimento di

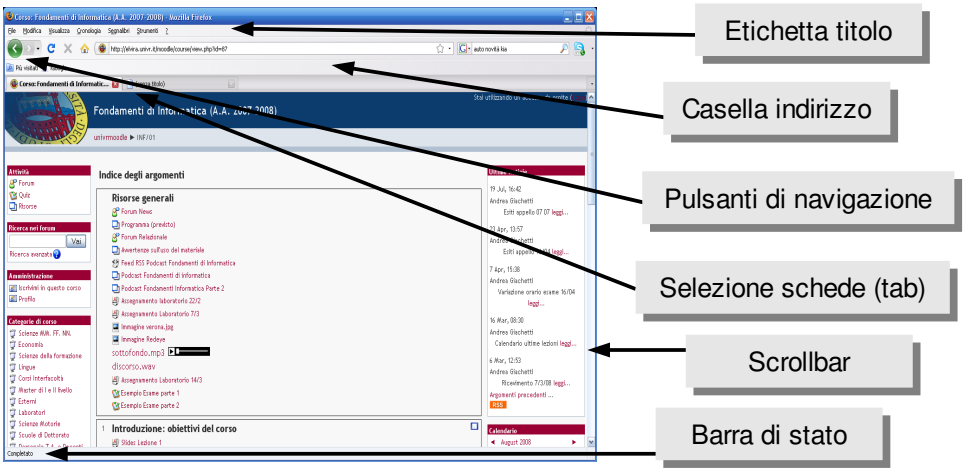


Figura 11: Interfaccia classica di browser web ed alcuni elementi caratteristici

file, possono usare protocolli criptati, e le loro funzionalità sono in continua evoluzione. Essi sono in grado di visualizzare documenti multimediali, non necessariamente integrati in ipertesti, come per esempio le immagini. Quando un tipo di documento non supportato direttamente dal browser (es. un documento pdf, un video, ecc.) viene richiesto, i browser possono richiamare applicazioni installate sul computer per gestirli. Ogni tipo di file (determinato dal MIME Type e dall'estensione) può avere associata un'applicazione **helper** (o **player**), che viene lanciata all'occorrenza per aprirlo. In caso non ve ne siano, il browser propone, in genere, all'utente il salvataggio del file sul disco locale. Se i file multimediali sono integrati all'interno dell'ipertesto (vedremo come ciò avviene nel capitolo 4.16), essi saranno invece gestiti dai cosiddetti **plugin**, componenti software creati appositamente per il browser (non applicazioni a sé stanti) che, con un meccanismo standard, integrano la propria interfaccia all'interno di quella riservata alla visualizzazione dell'ipertesto. Tra le opzioni che si trovano a disposizione sul browser, c'è anche l'attivazione dell'interprete Javascript e la visualizzazione della consolle degli errori per tale linguaggio. Vedremo in dettaglio cosa sia Javascript nel capitolo 4.15; esso è estremamente importante perché costituisce il meccanismo standard attraverso cui, in un documento HTML, che di per sé sarebbe statico e non interattivo, può essere reso dinamico attraverso programmi che vengono eseguiti dall'interprete e attraverso i quali le stesse parti del documento e del browser possono essere modificate. In pratica, questo fa sì che nelle pagine web possano essere inclusi veri e propri programmi che elaborano le azioni fatte dall'utente sull'interfaccia, eseguendo azioni programmate (ad esempio cambiare colore se il mouse passa sopra un oggetto, modificare le immagini sulla pagina, ecc.). Tutto questo senza inviare richieste al server via HTTP.

3.3 Il server web

Per realizzare un sito web statico, occorre, quindi, inserire il codice HTML degli ipertesti nella memoria di massa di un computer ove sia attivo un programma (server) che riceva le richieste dei clienti (browser) e sia in grado quindi di trovare il file richiesto e inviarlo all'indirizzo del

computer richiedente attraverso un messaggio HTTP di risposta.

Il termine server si può riferire sia alla macchina (host) su cui risiede il programma che risponde alle richieste, sia il programma (demone) che, sempre attivo sulla macchina host, riceve le richieste ed invia le risposte, utilizzando il protocollo HTTP.

Nella forma più semplice, il programma server non fa altro che ricevere gli URL delle pagine (e degli eventuali file multimediali) trovare i file corrispondenti nella cartella corrispondente dell'host server ed inviarli al client via rete utilizzando il protocollo HTTP (Figura 13).

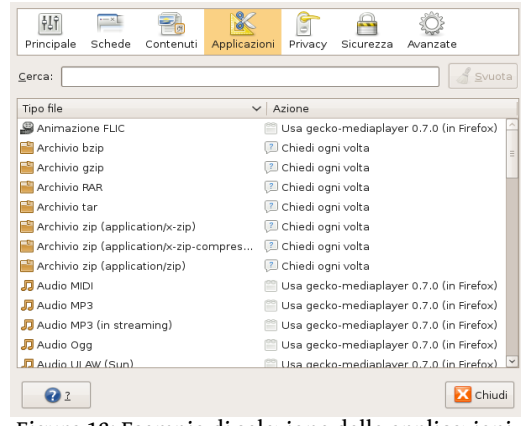


Figura 12: Esempio di selezione delle applicazioni helper per vari tipi di file sul browser Mozilla Firefox.

Chi crea i siti non deve fare altro, in questo caso, che creare i file degli ipertesti e gli eventuali contenuti aggiuntivi ad essi collegati (immagini, suoni).

Il programma server, anche per il servizio basilare che stiamo descrivendo, non si limiterà soltanto al trasferimento del file, ma dovrà anche gestire la sicurezza dei file stessi, limitandone se necessario l'accesso con password o utilizzando il protocollo sicuro HTTPS (che aggiunge la crittografia alla trasmissione) al posto di HTTP. Il server dovrà anche gestire correttamente gli accessi multipli, tenere traccia (mediante i cosiddetti log file) di tutte le attività svolte, e molto altro. Non ci soffermeremo su questi dettagli, ma ci limiteremo a ricordare che i server sia dal punto di vista dell'hardware che del software, devono essere creati utilizzando computer e applicazioni software molto affidabili e potenti per garantire un servizio efficiente. Il più noto programma di gestione per i server web è il già citato Apache, software open source che detiene quasi il 50% del mercato.

Come sa bene chi usa il web, i siti che si trovano online non sono però semplicemente

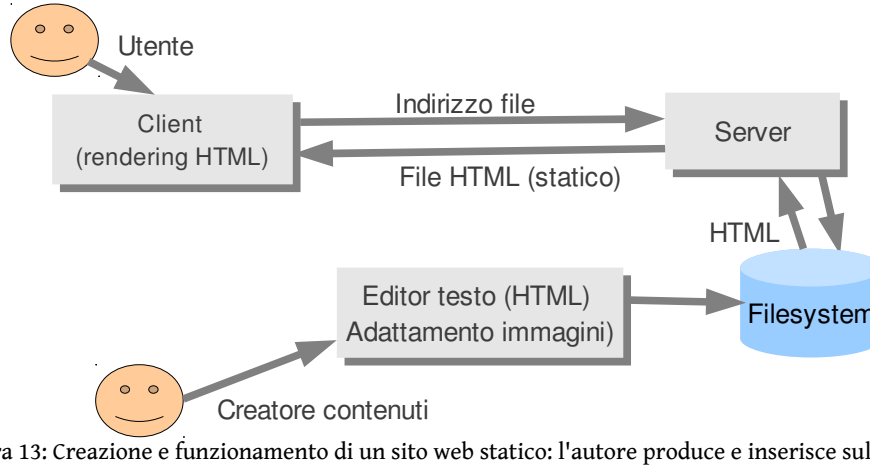


Figura 13: Creazione e funzionamento di un sito web statico: l'autore produce e inserisce sul server i documenti che verranno trasferiti immutati ai clienti.

collezioni di ipertesti più o meno complicati che integrano contenuto multimediale. Gran parte dei siti implementano **servizi** che consentono di inviare richieste e di ottenere risultati, iscriversi ad attività varie, gestire pagamenti, ecc. Questo non può certamente essere realizzato attraverso un server che si limiti a distribuire pagine HTML statiche. In questo caso il server deve consentire l'esecuzione di programmi complessi che ricevano le richieste (sempre codificate dal protocollo HTTP) contenenti i dati inviati dall'utente mediante le apposite interfacce messe a disposizione dal web (i cosiddetti **moduli** o **form** HTML) e le elaborino. Un motore di ricerca è, per esempio, un sito di questo tipo: quando si inseriscono parole chiave nella casella di testo di Google e si clicca, esse vengono inviate al server in un messaggio di richiesta HTTP. Sul server il messaggio avvia un programma che elabora i dati e porta alla generazione automatica della pagina di risposta. I programmi eseguiti "lato server" memorizzano eventuali dati da conservare in basi di dati, estraggono dalle basi di dati stesse o ricercano attraverso la rete le informazioni necessarie per soddisfare la richiesta, eseguono le elaborazioni opportune, e "rispondono" alla richiesta generando "al volo" (on the fly) un codice HTML da inviare all'utente come risposta e che verrà alla fine visualizzato dal browser (Figura 14). Questo tipo di pagine web sono dette a volte "dinamiche" in quanto non residenti staticamente sul disco rigido del server, ma generate automaticamente dall'elaborazione dati "server side" sulla base della richiesta. Non vanno confuse con le pagine "dinamiche" client side, rese tali dall'esecuzione dei programmi Javascript sul browser.

L'esecuzione di programmi che elaborano i dati inviati e generano la risposta HTML era, nelle prime applicazioni web, ottenuta attraverso un meccanismo standard detto CGI (Common Gateway Interface) mentre oggi è in genere realizzata integrando nel server stesso interpreti di **linguaggi di programmazione** specifici (ad esempio PHP, Perl, ASP), con i quali si realizzano le applicazioni dinamiche.

Per fare in modo che sia agevole memorizzare e recuperare sul sito grosse quantità di informazione attraverso sulla base di parole chiave, categorie, ecc., si installa di solito sul server anche un software di gestione di **basi di dati** (vedi paragrafo seguente) che permette ai programmi di aggiornare e recuperare i contenuti utili in maniera efficiente. Di fatto, quindi, un sito web dinamico lato server viene realizzato attraverso una serie di software differenti e non un semplice programma. Si parla in tal caso di **piattaforma** del server web; la più comune viene chiamata **LAMP**, acronimo che mette assieme tutte le componenti viste prima: un sistema operativo (**L**inux), un server web (**A**pache), un sistema di gestione di basi di dati (**M**ySQL) ed un linguaggio di scripting (**P**HP). Analoghe piattaforme commerciali combinano allo stesso modo i diversi elementi. L'invio delle informazioni ai server avviene in genere attraverso quelli che in HTML, vengono chiamati **moduli** o **form** (vedi capitolo 4.14), che inseriscono nelle pagine vere e proprie interfacce utente (menù, caselle di testo, ecc.). Quando compiliamo un form (ad esempio scriviamo la nostra richiesta a Google nella opportuna casella di testo), in pratica, anziché richiedere l'invio di un semplice file, si richiedono al server delle elaborazioni, che esso esegue al volo sulla base dei parametri inviati, sfruttando eventualmente archivi di dati collegati con il server stesso. I dati inseriti possono essere memorizzati nella base di dati e al client viene restituito comunque un documento HTML, non più, però, precedentemente memorizzato sul disco rigido, ma generato "al volo" (questo fatto fa sì, tra l'altro, che questo tipo di pagine non possa essere indicizzato dai motori di ricerca). I siti web più moderni utilizzano anche tecnologie attraverso le quali si possono inviare e ricevere dati dai server senza necessariamente ricaricare una nuova pagina web dinamicamente creata, ma aggiornandone direttamente una parte. Questo crea la possibilità di rendere le pagine web vere e proprie interfacce attraverso cui controllare una serie di attività che sono elaborate remotamente da uno

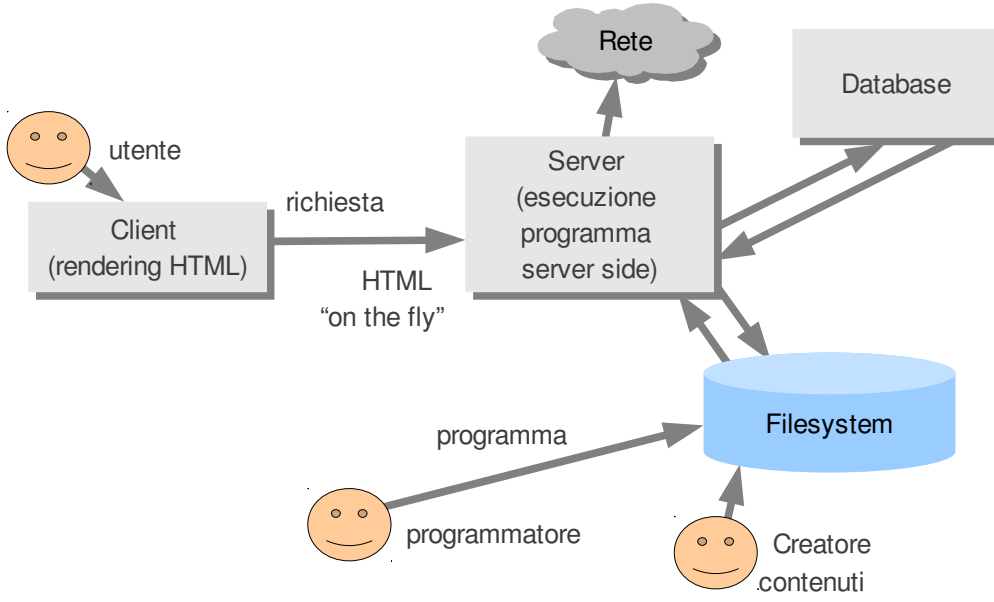


Figura 14: Esempio di sito web che usa la programmazione server side: gli autori del sito devono creare il codice per elaborare le richieste dei clienti.

o più server in rete. La principale tecnologia con cui questo meccanismo è realizzato si chiama **AJAX** (Asynchronous Javascript and XML).

Chi gestisce i siti web che usano la programmazione lato server, quindi, non deve semplicemente memorizzare gli ipertesti ed avere un programma server che li distribuisca, ma deve *scrivere i programmi* che creano le pagine utilizzando i linguaggi di scripting server side, gestire eventualmente le basi di dati, e così via.

Questo amplia notevolmente le potenzialità del web, rendendo possibile non soltanto l'accesso a collezioni di ipertesti, ma rendendo possibili servizi interattivi (ricerca, vendita, dialogo a distanza e così via). Naturalmente tutto ciò complica in modo notevole l'attività di creazione dei siti che implicherà, in questo caso, l'intervento di tecnici piuttosto esperti. In Figura 14 viene mostrato il funzionamento di un sito web che sfrutta la programmazione lato server. La sua gestione è chiaramente molto complicata e richiede una buona conoscenza delle tecnologie e della programmazione.

Chi vuole sviluppare siti complessi, però, può avere la vita molto semplificata dalla disponibilità di sistemi software completi che contengono già i programmi di gestione di vari tipi di servizi, ossia quelli generalmente forniti dalle tipologie classiche di sito web. Essi permettono di configurare e personalizzare i propri siti in maniera piuttosto semplice. Questi pacchetti software si chiamano **CMS** (**Content Management System**). Oggi gran parte dei siti professionali sono realizzati con piattaforme di questo tipo. Esempi di CMS open source di successo sono, ad esempio, Joomla, Plone e Drupal, ma ve ne sono molti altri, alcuni specifici per tipologie particolari di sito (blog, wiki, e-learning, e-commerce, ecc.).

In pratica i Content Management System evitano ad “autori” e “gestori” dei siti di dover essere anche abili programmatori, mettendo a loro disposizione una serie di programmi pronti utili a generare i servizi presenti nelle varie tipologie di siti web che devono semplicemente essere

configurati con le opportune opzioni di layout, grafica, ecc.

3.4 Basi di dati e gestione dell'informazione

Un sito attraverso il quale si accede a servizi complessi (ad esempio l'acquisto di beni, la fruizione di servizi pubblici, l'archiviazione di foto o video, l'aggiornamento continuo di notizie, e così via), deve mantenere al suo interno un grosso archivio di dati strutturati (ad esempio la lista degli utenti con username e password, il catalogo dei prodotti, gli indici per l'archiviazione di foto e video, eccetera) accessibili da differenti applicazioni (ad esempio il sistema di iscrizione, quello di login, ecc.). Su tali dati devono essere tipicamente effettuate delle operazioni caratteristiche come inserimento e cancellazione di dati, ricerca. La realizzazione di archivi elettronici efficienti e affidabili è uno dei principali settori di ricerca e sviluppo dell'informatica. I sistemi di gestione dei dati sopra citati prendono il nome di **Database Management System** (sistemi di gestione di basi di dati) spesso abbreviato semplicemente in **database** (anche se, ad essere precisi, database o base di dati sarebbe il nome riferito ai dati archiviati e non al sistema informatico). In essi si cerca di organizzare i dati in modo tale che siano integri, sicuri, non duplicati, collegabili tra loro attraverso le relazioni che li legano. I database più utilizzati sono quelli di tipo **relazionale**, appunto, dove si organizzano i dati e le loro relazioni attraverso **tabelle**.

Non ci dilungheremo qui sulle tecnologie delle basi di dati, per le quali rimandiamo ad un testo specifico, in ogni caso diciamo che il programma di gestione del database che viene utilizzato dai server web è un'altra applicazione di tipo server, che gestisce il collegamento tra dati memorizzati fisicamente sulla macchina e modello concettuale degli stessi. L'applicazione è, pertanto, in grado di fornire i servizi di memorizzazione e ricerca ad applicazioni esterne dialogando con esse in un opportuno linguaggio (si parla in questo caso di linguaggi di **query**, come il noto SQL, Structured Query Language).

Quando, per esempio, richiediamo dei dati memorizzati su un sito (per esempio “fornisci la lista degli iscritti al corso che abbiano sostenuto almeno tre esami”), generalmente la nostra richiesta, fatta in qualche modo attraverso il browser, attiva un programma che si collega con il database ed effettua una query in linguaggio SQL che corrisponde alla richiesta stessa. Esso, riceverà quindi in risposta i dati che utilizzerà per generare la pagina web da inviare al browser. Il DBMS garantisce che l'accesso ai dati e l'aggiornamento degli stessi siano sicuri ed efficienti.

Esistono molti sistemi disponibili per la gestione di basi di dati, sia commerciali (ad esempio Oracle, Microsoft SQL Server) che open source (ad esempio MySQL o PostgreSQL, molto usati per i siti web). Tutti i principali Content Management System sviluppati per creare i siti web memorizzano i dati inseriti dagli autori utilizzando un server di basi di dati (e i programmi che generano le pagine dinamiche “interrogano” quindi il sistema di basi di dati stesso per realizzare il proprio scopo) e necessitano, quindi, della preventiva installazione di un software di questo tipo.

3.5 Tipologie di siti web

La nostra breve introduzione al web ha mostrato che si possono creare differenti tipi di siti su cui l'utente può realizzare attività molto varie. In realtà, però, l'evoluzione del mercato legata a Internet ha fatto affermare tipologie abbastanza precise di siti web di successo. Ognuna di esse è in genere caratterizzata da una modalità particolare di interazione con l'utente e per la loro

creazione sono utilizzati spesso specifici programmi di gestione (Content Management System) sul server.

Tra i tipi da citare annoveriamo sicuramente i siti di **commercio elettronico** (e-commerce) perché rappresentano ormai una modalità di acquisto sempre più utilizzata: chiaramente saranno realizzati attraverso applicazioni che implicano gestione concorrente di grosse basi di dati (cataloghi, magazzini), l'integrazione di sistemi di pagamento sicuri che implicano l'utilizzo di server HTTPS (cioè resi sicuri con con crittografia dei dati trasmessi), connessioni con database bancari, ecc. Dal punto di vista tecnico, sebbene esistano rischi di violazione dei dati riservati mediante decodifica dei messaggi criptati, con le attuali tecnologie utilizzate su questi siti, le probabilità che si verifichino problemi (es. furto su carta di credito) è estremamente bassa (è molto più probabile che i dati sulla carta o sul bancomat vengano rubati acquistando normalmente nei negozi). Ovviamente gli utenti dovranno ugualmente prestare una certa attenzione alle operazioni svolte: consigliamo quindi di utilizzare siti conosciuti, verificare l'uso di protocolli sicuri durante le operazioni di pagamento (controllare sulla barra degli indirizzi del browser che l'indirizzo sia di tipo "*https://...*") e, ovviamente non rispondere mai a messaggi di posta elettronica di provenienza non sicura che chiedono dati bancari (il cosiddetto "phishing").

I siti di aziende che invece non vendono direttamente online, ma presentano solo attività e prodotti dell'azienda sono spesso definiti "**siti vetrina**". Essi si caratterizzano in genere per uno stile grafico accurato, la presenza di video od animazioni (in Flash) e la possibilità di accedere a vario materiale informativo.

Una tipologia di sito web importante è poi quella dei siti della **pubblica amministrazione.** Si parla oggi di **e-government,** perché oggi, sempre attraverso le tecnologie della programmazione server side, si possono erogare via web molti servizi al cittadino da parte di comuni, regioni, aziende concessionarie, università, società di servizi, trasporti, ecc. Se le prime forme di utilizzo del web per la gestione di pratiche burocratiche mettevano a disposizione semplicemente il testo delle normative o la modulistica in forma scaricabile, oggi attraverso la programmazione lato server e i database, è possibile operare direttamente e prenotare visite, iscriversi a concorsi, pagare tasse, ecc. Al giorno d'oggi gli studenti universitari forse neppure si rendono conto che soltanto pochi anni fa tutte le pratiche relative ad iscrizioni, prenotazione di esami, e così via richiedevano di recarsi in uffici di docenti e segreterie, fare code, compilare a mano documenti. Oggi, infatti, basta spesso compiere poche azioni utilizzando i siti di ateneo per ottenere gli stessi risultati. I siti delle istituzioni (detti spesso **siti istituzionali**, come quelli di governo, regioni, autorità, ecc.) e delle varie aziende pubbliche possono, chiaramente fornire diversi livelli di servizio a seconda dell'avanzamento delle procedure di informatizzazione della struttura, quello che va, però, detto è che dal gennaio 2006 è in vigore in Italia il "Codice dell'Amministrazione digitale" che impone che i siti rispettino principi di usabilità ed accessibilità (cap. 3.6).

Il web è poi oggi una delle principali fonti dove andiamo a cercare informazioni sugli avvenimenti che accadono nel mondo. I siti di **news** o i **quotidiani online** hanno quindi molto spesso soppiantato i mezzi cartacei e stanno rivoluzionando il mercato dell'informazione (tanto che si dibatte molto, specialmente nel mondo anglosassone, sul futuro dei quotidiani cartacei e sul modo di far pagare all'utente l'informazione online).

Al di là della presenza e continuo aggiornamento degli articoli, dell'inclusione di immagini e filmati, i siti di news presentano alcune caratteristiche peculiari, come la presenza di parti interattive come sondaggi, forum, **blog** (vedi cap. 3.8) redatti dai giornalisti. Inoltre vengono

fornite anche informazioni sugli aggiornamenti mediante la tecnologia dei **feed RSS** vedi cap. 3.8) e si tendono sempre più spesso ad includere, nelle homepage dei quotidiani online, collegamenti a vari tipi di siti di servizi in modo tale che esse possano essere utilizzate pagina di inizio per la navigazione dell'utente.

I siti che vorrebbero essere utilizzati a questo scopo prendono in genere il nome di "**portali**", nel senso che dovrebbero agire come porta di ingresso ai servizi del web. Molto spesso i siti che vorrebbero essere utilizzati in questo modo vengono creati dagli stessi provider dei servizi di connettività (es. Lycos, Libero, ecc.). Oggi questi portali tendono ad essersi evoluti in pagine personalizzabili che integrino diverse applicazioni web (es. agenda, news, lettore RSS, ecc.), come per esempio iGoogle. Spesso ci si riferisce a siti che integrano applicazioni prese da varie parti con il termine **mashups**.

Più che dai portali, però, la navigazione viene iniziata dagli utenti a partire dai cosiddetti "**motori di ricerca**" (comunque inclusi sempre dai portali stessi), una delle tipologie di sito di maggior successo. I motori di ricerca sono, in pratica, cataloghi del web che permettono di trovare informazioni e servizi utilizzando l'immissione di parole chiave. Come facciano a trovare le risposte più vicine alle intenzioni dei clienti che hanno effettuato le richieste sarà discusso meglio nel capitolo 3.9 . In ogni caso, tutti coloro che utilizzano il web ne fanno certamente uso e tutti sicuramente conoscono l'azienda Google, che, partendo da un efficacissimo motore di ricerca, ha creato un vero e proprio impero sul web e oggi fornisce svariati tipi di servizi avanzatissimi per la gestione via interfaccia web di dati di ogni genere. Altri tipi di sito ormai utilizzato da moltissimi utenti come forma primaria di accesso alla rete sono i cosiddetti **social network**, che in pratica vincolano l'accesso alle risorse in rete alle segnalazioni di un gruppo ristretto di utenti collegati ("amici"). Anche di essi parleremo in modo più esteso nel capitolo 3.8 .

Tra i molti servizi di ricerca, una particolare importanza stanno assumendo quelli che utilizzano mappe e metodi di **georeferenziazione** (utilizzo dei cosiddetti Sistemi Informativi Geografici o GIS, che collegano informazione a coordinate geografiche). Su alcuni siti (come Google Maps, Maporama, ViaMichelin, PagineGialle, ecc.) è, infatti, possibile vedere mappe, mappe fotografiche, localizzare servizi sul territorio in base alla posizione richiesta o visualizzata, disegnare itinerari ottimali per i propri percorsi. Infine dobbiamo sottolineare che il web non sta diventando centrale solamente per quanto riguarda l'informazione, ma anche per quanto riguarda l'**intrattenimento**: molte stazioni radiotelevisive diffondono le loro trasmissioni in diretta attraverso una tecnologia detta streaming, che vedremo, oppure mettono online registrazioni delle stesse (podcasting). Allo stesso modo, approfittando delle varie tecnologie per la programmazione client e server side molti siti mettono a disposizioni giochi interattivi con cui gli utenti possono cimentarsi (siti di **gaming**).

3.6 Problematiche di utilizzo del web: usabilità ed accessibilità

Un sito web rappresenta un'applicazione multimediale interattiva che può essere utilizzata in maniera efficace da una vasta classe di utenti per ottenere differenti scopi. Questo implica particolari attenzioni nella fase di progettazione dei siti stessi, che devono tenere conto delle esigenze degli utenti desiderati.

Nell'ambito dell'informatica, si è sviluppata una vera e propria disciplina detta **interazione**

uomo-macchina, che studia le modalità di interazione appunto tra uomo e i sistemi automatici usati per i più diversi scopi e le metodologie per garantire la massima **usabilità** dei sistemi stessi. L'usabilità viene definita (secondo uno standard definito dall'ISO, cioè l'organizzazione internazionale per la standardizzazione), come *l'efficacia, l'efficienza e la soddisfazione con le quali determinati utenti raggiungono determinati obiettivi in determinati contesti*.

Che le azioni che svolgiamo su un sistema interattivo abbiano un obiettivo prefissato è abbastanza intuitivo. In ogni caso gli studiosi di interazione uomo-macchina modellano l'interazione con una serie di passi con cui si ideano, eseguono e valutano le azioni, in modo da scomporne ed analizzarne i processi. Nel modello di interazione più famoso (di Norman) questi passi sono: formulazione dell'obiettivo, formulazione dell'intenzione, identificazione dell'azione necessaria, esecuzione dell'azione, percezione dello stato del sistema a seguito dell'azione, interpretazione dello stato del sistema e valutazione del risultato rispetto all'obiettivo.

Senza entrare nel dettaglio delle metodologie di progettazione utili a introdurre e verificare nelle interfacce web **efficacia** (l'accuratezza e completezza con cui l'obiettivo viene raggiunto), **efficienza** (cioè le risorse spese per ottenerlo) e **soddisfazione** (comfort e accettabilità del sistema), possiamo comunque ottenere alcune **linee guida** per migliorare l'usabilità dei sistemi, semplicemente analizzando le caratteristiche dell'uomo e le possibilità messe a disposizione dalle interfacce utente nel nostro caso realizzabili mediante le tecnologie web.

Facciamo qualche esempio per chiarire meglio le idee. Attraverso lo studio del sistema visivo umano possiamo capire come usare l'impaginazione ed il colore per rendere più efficace l'interazione. Gli studi mostrano che la percezione del colore non è costante, ma è ridotta per le frequenze del blu: quindi una buona norma di progettazione consisterà nell'*evitare di scrivere il testo rilevante in tale colore*. Ancora: più dell'8% della popolazione maschile (molto meno per le donne) è affetta da daltonismo, ovvero non è in grado di distinguere correttamente le tinte (la patologia più frequente impedisce la distinzione del rosso dal verde). La norma di progettazione che ne deriva è quindi *evitare di utilizzare contrasti di tonalità*, meglio evidenziare i contenuti con contrasti di luminosità. Colori molto saturi e ravvicinati possono, poi, creare fastidio a causa delle difficoltà di messa a fuoco contemporanea di segnali luminosi a frequenza diversa: anche questi andrebbero quindi evitati.

L'uomo, come hanno mostrato gli psicologi della "Gestalt" nel secolo scorso, tende poi naturalmente a raggruppare gli oggetti sulla base di determinati principi (vicinanza, somiglianza, continuità) e conviene usare questi principi per distribuire le componenti della propria interfaccia (ad esempio collegando, riquadrando parti con significato comune, usando colori o caratteri coerenti con il significato, ecc.).

Lo studio dei processi mnemonici, invece, rende evidente che la memoria a breve termine dell'uomo si limita a circa 7 unità base o "chunk" (ad esempio gruppi di 2-3 numeri, parole). Questo vuol dire che occorre evitare di costringere l'utente a ricordare un numero superiore di dati. Pensate, ad esempio, al classico messaggio di errore che compare quando dimenticate di compilare qualche campo in un modulo di iscrizione online. Se il messaggio di errore vi elenca i dati dimenticati, ma vi costringe a ricordarli a memoria quando ritentate l'iscrizione, è decisamente poco usabile. Meglio, allora, un sistema che segnali gli errori sullo stesso modulo errato, senza imporre sforzi mnemonici. I siti migliori utilizzano, infatti, quest'ultima opzione.

Un sito usabile deve anche rendere facilmente comprensibile quali siano le azioni che si possono fare su di esso e quali risultati ci si devono attendere. Quindi è buona norma seguire le

convenzioni in voga sulla posizione di testo, link e contenuti (ad esempio gli utenti delle pagine web usano leggere per lo più il testo evidenziato in alto e a sinistra, trascurando il resto ove di solito ci sono le informazioni importanti), utilizzare i colori standard per evidenziare i link, fornire le parti di interfaccia di quella che si chiama una buona "**affordance**" (cioè la proprietà di rendere chiara la propria funzione: il disegno di un tasto in rilievo invita a cliccarci sopra!).

Gli studiosi di interazione uomo macchina hanno poi analizzato molto bene i processi che portano a commettere errori nell'uso dei sistemi interattivi identificando alcune strategie per prevenirli, migliorando così l'utilizzabilità delle interfacce.

Gli errori umani possono sostanzialmente essere di due tipi, i cosiddetti **slip** o **lapsus** (sviste), cioè errori di esecuzione di un'azione da parte di un utente che pure sapeva cosa doveva fare e i cosiddetti **mistake** o errori concettuali, dovuti ad una scarsa conoscenza del sistema. Per quest'ultimo tipo di errore i progettisti possono svolgere un'azione preventiva rendendo più semplici i sistemi e cercando di rendere maggiormente fruibile la documentazione degli stessi.

Per evitare le sviste i progettisti possono fare molto. Ad esempio, si può evitare di accostare molto i tasti cliccabili, specie se danno origine ad azioni irreversibili e differenti, in modo che non vengano azionati per sbaglio. Un altro espediente può essere poi limitare la libertà dell'utente al minimo indispensabile. Se si realizza un'interfaccia in cui egli deve inserire il nome della sua città, si può evitare di richiedere la digitazione del nome sulla casella di testo, inserendo invece un menu a discesa con la lista dei comuni su cui marcare quello di residenza. Se l'utente deve inserire la data gli si può evitare di scriverla coi tasti, facendo invece cliccare un punto su un calendario. Questo tipo di funzioni che obbligano l'utente a una compiere azioni semplici si chiamano, appunto, **funzioni obbliganti** e riducono significativamente le possibilità di errori. In ogni caso, è poi buona norma cercare di limitare i danni rendendo il più possibile reversibili le azioni svolte, inserendo richieste di conferma e così via. Per maggiori informazioni riguardo l'usabilità si rimanda al bel saggio divulgativo "La caffettiera del masochista" di Donald Norman e per studi sull'usabilità dei siti web al sito di Jacob Nielsen *http://www.useit.com*.

Un ulteriore problema riguarda poi la cosiddetta **accessibilità** dei siti web. Per accessibilità si intende la possibilità da parte dei siti di essere facilmente utilizzabili da tutte le diverse categorie d'utenza. Qui il problema sta quindi nella necessità di rendere i contenuti fruibili da utenti con disabilità fisiche (difetti di vista o udito, deficit motori) o intellettive, così come da parte di categorie particolari come ad esempio bambini, anziani, persone con collegamento Internet lento o PC o browser vecchi.

L'accessibilità da parte di queste varie categorie è ottenibile anche mediante **tecnologie assistive** che supportino diverse modalità di input/output alternative per chi non può usufruire dei canali standard. Ad esempio, chi non vede può utilizzare lettori audio di schermo o dispositivi braille, chi ha problemi motori può utilizzare lo sguardo per puntare e cliccare con i sistemi cosiddetti di "eye tracking", o utilizzare riconoscitori vocali, mentre chi ha problemi di udito dovrebbe ricorrere a sistemi di descrizione o sottotitolazione per il contenuto audio.

Non solo è evidentemente buona norma fare sì che anche utenti diversamente abili o non dotati di calcolatori di ultima generazione possano fruire comunque dei contenuti di un sito, ma è anche obbligatorio per legge se questo sito ricade in particolari categorie (ad esempio pubblica amministrazione, concessionarie pubbliche, educazione, ecc.). Esiste infatti una apposita legge, la n. 4 del 9/1/2004 nota come "legge Stanca", dal nome dell'allora ministro e pubblicata nella

Gazzetta Ufficiale il 17 gennaio 2004, che definisce i criteri per garantire l'accessibilità ai siti e i soggetti che vi si devono uniformare (pubblica amministrazione, enti pubblici economici, aziende concessionarie di servizi pubblici, aziende municipalizzate regionali, enti di assistenza e di riabilitazione pubblici, aziende di trasporto e di telecomunicazione a prevalente capitale pubblico). Informazioni dettagliate a riguardo possono essere reperite sul sito ***http://www.publiaccesso.gov.it*** ove si trovano tutti i riferimenti normativi compreso il DPR del regolamento di attuazione e i decreti ministeriali di specificazione.

Nell'ambito della comunità web in generale, per garantire l'accessibilità si seguono linee guida come le **WCAG, Web Content Accessibility Guidelines** della WAI, Web Accessibility Initiative (sezione del World Wide Web Consortium).

Esse stabiliscono, ad esempio, che, per quanto riguarda le pagine web, un buon supporto per l'accessibilità consiste nella correttezza della codifica, che consente una buona interpretazione ai diversi browser, anche quelli basati su sistemi assistivi (per esempio lettori di schermo o barre braille per non vedenti).

Un'altra buona norma consiste nella linearizzabilità del contenuto, cioè nel fare in modo che le parti della pagina possano essere rappresentate in maniera sequenziale prescindendo quindi dall'impaginazione, dato che la rappresentazione sequenziale è l'unica supportata da un lettore di schermo o da una barra braille. I documenti, poi, dovrebbero essere il più possibile leggeri (per non escludere, ad esempio, chi ha collegamenti Internet con scarsa banda) ed adattabili a diverse modalità di visualizzazione. I testi dovrebbero essere chiari e non ambigui, con possibile supporto multilingue. Vanno evitate indicazioni non traducibili cambiando modalità di rappresentazione (ad esempio “clicca qui” sopra un pulsante, che non ha senso se non si vede la rappresentazione spaziale della pagina). Anche i riferimenti linguistici e culturali che non siano universali per i fruitori del sistema andrebbero evitati.

3.7 Il web per l'utente: navigazione standard e interazione su siti attivi

Una volta comprese le problematiche dell'interazione uomo-macchina, il successo immediato che ha avuto il web fino dalle sue origini diventa facilmente spiegabile considerando l'efficacia della sua esperienza di interazione. Allo stesso modo se ne possono capire i limiti e le problematiche.

L'interazione standard sul web, costituita dal passare da una pagina all'altra cliccando su determinate parti del documento di origine dette **àncore,** oppure utilizzando gli elementi dell'interfaccia del browser (casella indirizzo, tasti “indietro” e “home”, cronologia, preferiti), viene detta **navigazione**. La possibilità di muoversi non linearmente all'interno della collezione di dati concede molta libertà all'utente, che può selezionare meglio il contenuto di suo interesse. Questo metodo ha però dei problemi. Uno consiste nel fatto che, utilizzando il protocollo HTTP, ogni richiesta di pagina è un'azione indipendente; quindi, nell'interazione web standard non c'è “memoria” delle attività precedenti (questo limita il tipo di attività che si realizzano sul sito alla semplice consultazione di documenti, se non si utilizzano trucchi o tecnologie diverse per ovviare al problema). Un altro problema consiste nel rischio dell'utente di “perdersi” all'interno della collezione di pagine, fenomeno chiamato dagli esperti di interazione uomo-computer “**lost in hyperspace**”. Pensiamo, infatti, ad un sito complesso e strutturato, come quello di un'università. A partire dalla pagina principale (**homepage**) potrei accedere, per esempio, ad una pagina sui corsi, da qui ad una sui docenti e così via. E poi? Come faccio in un

qualunque istante a capire dove mi trovo all'interno della rete costituita dalle varie pagine del sito e dai loro collegamenti reciproci? Come faccio a tornare al punto di partenza o arrivare a un punto di mio interesse?

Per ridurre questa problematica i siti ben progettati possono adottare due soluzioni: la prima è quello di strutturare in modo opportuno i siti, in genere attraverso una **struttura gerarchica**, attraverso cui la "rete" dei collegamenti sia modellata come un albero: i collegamenti vanno da livello superiore a livello inferiore. La seconda, spesso legata alla prima è quella di inserire nelle pagine degli aiuti per la navigazione, ad esempio mappe, che riproducano la struttura gerarchica del sito, percorsi guidati, per consentire all'utente di svolgere compiti predeterminati e link di aiuto all'interno delle pagine per muoversi nel modo desiderato (es. menù, toolbar, oppure le molto diffuse "briciole di Pollicino" (Figura 15), cioè quelle sequenze in genere inserite in alto in cui si "vede" il percorso eseguito all'interno del sito e cliccando si può tornare ad una pagina della sequenza eseguita).

Un altro problema nel passare tra una pagina è l'altra può essere dovuto alla lentezza nel caricamento delle pagine stesse, che ovviamente dipenderà sia dalle caratteristiche dei documenti richiesti, sia dal tipo di connessione a disposizione. Se non si stanno svolgendo particolari compiti, l'esperienza di navigazione non subisce grossi fastidi se le pagine sono caricate nel giro di qualche secondo. Problemi di lentezza del caricamento possono essere causati da immagini o contenuti audio/video pesanti (che occupano, cioè, molto spazio in memoria). Trucchi per ridurre tali problematiche consisteranno, per gli autori, nel limitare le dimensioni delle immagini o riutilizzare più volte le stesse immagini sul sito (quando un'immagine viene trasferita rimane poi nella cache del browser, cioè in un'area del disco ove il programma client memorizza temporaneamente i documenti recenti scaricati, e non viene più trasferita se nuovamente richiesta).

Completamente differente è l'interazione dell'utente quando abbiamo a che fare con siti "attivi" o "dinamici". Come già visto, questi sono i siti dove, anziché sul solo collegamento ipertestuale, l'interazione si basa su vari "eventi" generati dall'utente sull'interfaccia, come premere tasti, inserire testo, ecc. Questi "eventi" saranno gestiti in due possibili modi: l'esecuzione di un programma (script) scaricato insieme alla pagina senza inviare informazioni al server (pagine attive client side), oppure inviando richieste al server, che le elaborerà inviando al client un nuovo documento ipertestuale (pagine attive server side). Esiste anche un modo ibrido di realizzare pagine dinamiche, realizzato includendo nella pagina componenti che dialogano con server remoti senza che la struttura della pagina visualizzata cambi (si pensi ai componenti inserite nelle pagine che visualizzano finestre di chat, mappe navigabili, giochi, ecc.). Un enorme vantaggio di queste modalità di interazione sta nel consentire la realizzazione sul browser web di compiti complessi, impossibili con la semplice navigazione ipertestuale. Attraverso le tecnologie del web "attivo" si realizzano quelli che possono essere definiti **siti "operativi"** ove lo scopo dell'utente non è di visualizzare pagine, ma quello di **realizzare un obiettivo** attraverso azioni, esattamente come quando si utilizza un programma in esecuzione locale. Diventano in questo caso critiche le prestazioni dei sistemi: in questa interazione, infatti, spesso si aspettano risposte in tempo reale dai server remoti, impossibili se la rete non è

Figura 15: Le cosiddette "briciole di Pollicino" inserite in molti siti come ausilio alla navigazione.

abbastanza veloce. Inoltre le tecnologie attraverso cui sono realizzati alcuni effetti (es. Javascript, Flash, Java, ecc.) sono spesso recenti e non ben standardizzate, per cui possono esservi problemi di funzionamento per alcuni utenti non dotati di connessione veloce e hardware e software aggiornati. Un problema non indifferente nell'aggiunta di componenti attive nelle pagine è che queste non consentono di utilizzare le modalità di navigazione standard (ad esempio l'uso del tasto "indietro" o la cronologia). Questo può provocare disagio negli utenti, dato che, comunque, la navigazione resta sempre la modalità di interazione da privilegiare sul web e quella a cui siamo più abituati. I siti moderni cercano di conciliare le due tipologie di interazione, in modo da essere il più possibile utilizzabili in modo semplice ed intuitivo, pur aggiungendo le moderne funzionalità interattive.

Le indicazioni del W3C spingono in genere verso l'uso di metodologie standard nella realizzazione dei siti, che preservino, ove possibile, la navigazione classica, dato che essa rende anche più semplice supportare l'accessibilità dei siti.

3.8 Web 2.0 e siti "user centered"

Uno dei principali utilizzi del web è sempre stato quello consistente nell'inserimento, da parte dell'utente, di informazioni personali, scritti, foto, insomma il proprio profilo e le proprie creazioni, con lo scopo di metterle a disposizione di chi le voglia vedere, farsi conoscere o comunicare in modo indiretto con amici lontani. Si parla in tal caso di siti o pagine **personali**. Agli inizi della diffusione popolare dell'uso di Internet, però, le persone in grado di sfruttare questa opportunità erano relativamente poche. Questo perché, innanzitutto, occorreva creare le pagine scrivendone direttamente il codice HTML, cosa che, sebbene non eccessivamente complicata (vedi cap. 4), richiede un po' di studio. Inoltre non era possibile inserire contenuti audio e video a causa della tecnologia acerba e delle reti lente. Con la banda larga a disposizione e la possibilità di gestire i contenuti mediante tecnologie avanzate, si sono sviluppati molti servizi che permettono agli utenti di diventare attivi creatori di contenuto web senza senza dover gestire la vera e propria costruzione del sito personale. Questo cambio di prospettiva è quello che in genere viene definito il passaggio al "**web 2.0**", uno slogan un po' fuorviante (2.0 in gergo informatico significa la seconda release ufficiale di un software, in realtà non c'è stata nessuna variazione improvvisa nella tecnologia del web), ma che spesso si sente nominare riferendosi a varie evoluzioni e mutamenti nei servizi web rispetto ai pionieristici anni '90. Vediamo in dettaglio alcuni esempi di queste nuove tipologie di servizio ed evoluzioni tecnologiche e cosa consentono di realizzare a chi li utilizza.

Dal sito personale al "blog"

Come detto, gli utenti che volevano realizzare un cosiddetto **sito personale** su cui inserire testo e foto per rendersi rintracciabili in rete, esprimere idee, descrivere la propria attività o la propria famiglia, dovevano scrivere personalmente gli ipertesti, trasferirli sul server e ripetere l'operazione per ogni aggiornamento del contenuto. Oggi per fare operazioni simili, gli utenti ricorrono in genere ai cosiddetti **web log**, più comunemente chiamato "**blog**". In gergo informatico un "log" file è un file che tiene traccia di tutte le attività di un dato sistema o software. L'idea è, in questo caso, fare qualcosa di simile, cioè una sorta di diario, pubblicato in rete. In pratica, i provider di servizi di blogging (ad esempio Splinder, Blogger, Wordpress) mettono a disposizione uno spazio per l'utente per scrivere testi od inserire contenuti multimediali utilizzando interfacce semplici e forniscono modalità predefinite di interazione

(commenti dei lettori, riferimenti a siti esterni, ecc.). I sistemi informatici che li gestiscono sono degli appositi Content Management System che tengono traccia perenne di tali interventi nei propri database. L'utente non si deve preoccupare della gestione della grafica, del database od altro, ma solo di personalizzare l'aspetto e di inserire il contenuto mediante interfacce "user friendly", simili a un text editor tipo Word (Figura 16). Questo ha consentito a molti giornalisti, personaggi di cultura e così via di aprire spazi di intervento e dibattito in rete in maniera molto semplice ed immediata e permette a chiunque di fare altrettanto.

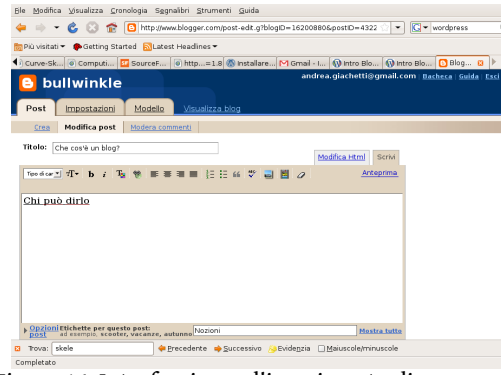


Figura 16: Interfaccia per l'inserimento di un nuovo "post" in un blog

La peculiarità del blog rispetto ad altri meccanismi simili di pubblicazione è la strutturazione abbastanza standardizzata dell'interazione autore/lettori: l'utente può in genere inserire informazioni personali e alcuni campi (es. lista dei siti consigliati, ecc.) da rendere disponibili al lettore che vengono in genere visualizzati in un formato standard (Figura 17).

I contenuti poi inseriti (post), vengono in genere visualizzati in colonna dal più recente al più vecchio. Tutti gli interventi, anche i più remoti, vengono conservati e restano accessibili nel tempo a chi voglia commentarli e collegarli alle proprie pagine attraverso link permanenti (permalink) ovvero indirizzi di URL in cui sarà sempre presente il post considerato. Ogni post può di solito essere commentato dai lettori e si possono tracciare con meccanismi appositi (trackback, pingback) le citazioni fatti da altri blog ai propri interventi. A parte i blog di personaggi noti, spesso collegati a testate giornalistiche o gruppi editoriali, od alcuni rarissimi casi di blog che hanno un certo seguito o influenza sull'opinione pubblica, il pubblico dei blog è in genere composto da comunità di persone che si scambiano idee e che condividono esperienze o commenti. La struttura del blog si presta comunque anche a diffusione di esperienze pratiche, istruzioni per l'uso di sistemi informatici, informazione su novità su vari argomenti di interesse.

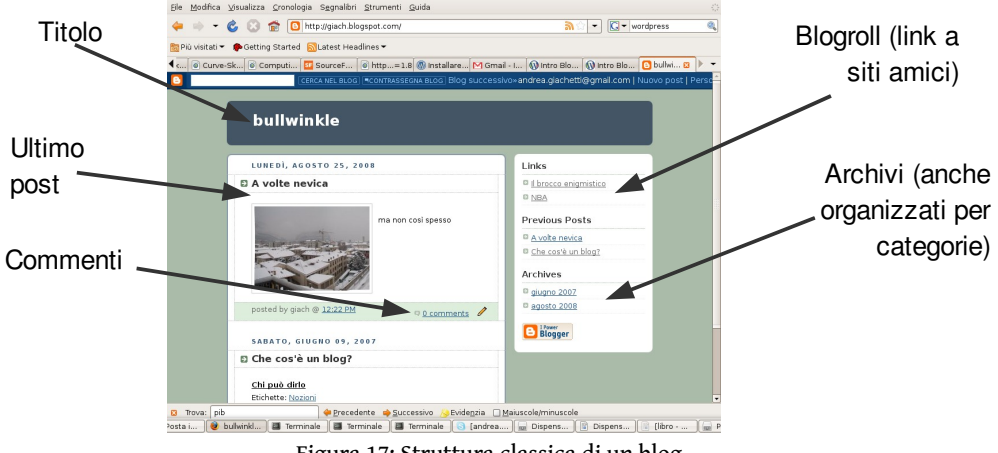


Figura 17: Struttura classica di un blog

Tecnologie di syndication

Figura 18: L'icona che indica il link ai web feed

Gli utenti che vogliono consultare gli aggiornamenti tanto dei blog che dei siti di informazione più "istituzionali" dovrebbero in teoria consultare periodicamente i siti per vedere gli aggiornamenti, cosa piuttosto difficile e che richiede tempo. Una tecnologia che sta prendendo piede per ovviare a questi problemi è quella dei metodi di **syndication**, cioè di distribuzione di contenuti. In pratica consistono nella pubblicazione di cosiddetti **feed**, liste di brevi schede testuali sugli aggiornamenti del sito contenenti i relativi collegamenti al contenuto completo della fonte di informazione. I servizi di feeds, realizzati con formati appositi basati su XML (es. RSS o Atom), consentono all'utente iscritto di accedere con mezzi diversi (esistono opportuni programmi detti "feed reader" per consultare la lista degli aggiornamenti a cui si è iscritti, oppure esistono modalità per la loro consultazione all'interno dei browser, con dispositivi portatili, ecc.) a queste liste ed essere informati del sommario degli aggiornamenti senza doversi collegare con i siti. Lo stesso meccanismo è utilizzato per gestire gli aggiornamenti dei cosiddetti "podcast" cioè periodici audio o video che si sono diffusi con l'avvento dei lettori portatili di file multimediali.

Community e social network

Altre tipologie di siti web attraverso cui gli utenti non esperti possono facilmente creare o mettere in rete contenuti sono le cosiddette **community** o i **social network**.

In genere si tratta di siti che consentono all'utente di iscriversi inserendo un proprio profilo consultabile all'esterno e svolgere attività eterogenee, dalla semplice gestione di album di file multimediali a comunicazione con amici, gestione di diari, ecc., consentendo quindi modalità di interazione più elastiche dei semplici blog. Alcuni di questi siti sono orientati alla diffusione/scambio di particolari contenuti (es. YouTube per il video, Flickr per le foto digitali), mentre altri, i cosiddetti **social network**, sono più genericamente dedicati alla creazione di reti di comunicazione tra gruppi di amici che possono così interagire in modo differente (messaggi, bacheche, ecc.)

I social network stanno godendo di un vero e proprio boom di iscrizioni, basti pensare che i due social network più famosi, cioè i siti Facebook e Myspace, vantano centinaia di milioni di iscritti nel mondo, e, anche per questo godono di grande risonanza mediatica (ma forse, viceversa, il numero di iscrizioni è causato dalla risonanza mediatica).

Si parla di reti sociali perché la principale peculiarità di questi servizi, rispetto agli altri sistemi assistiti di pubblicazione di contenuto, consiste nella definizione, per ogni utente, di una rete di connessioni che lo lega ad altri iscritti al sito. Questa funzionalità consente la creazione di reti di amici o di persone che condividono interessi o passioni e facilita lo scambio di dati o la comunicazione a distanza tra gruppi. In alcuni casi i social network sono specifici per l'ambito lavorativo, come nel caso di LinkedIn, e possono essere utili a contattare colleghi o per ricerche/offerte di lavoro.

Ci sono molte discussioni sull'utilità o la serietà di queste reti, anche perché, nei casi di maggior successo, come Facebook, i profili personali inseriti nei siti sono spesso fittizi (basta un email per iscriversi e basta un click di conferma o un email per fare amicizia o iscriversi a gruppi od attività), il numero di persone con cui si interagisce realmente resta comunque basso e l'uso di

questi strumenti crea una certa dipendenza.

Sicuramente il motivo del grande successo è dovuto ad un colossale fenomeno di imitazione collettiva ed all'ottima usabilità di alcuni strumenti di messaggistica, in realtà, però, essi non propongono nulla di particolarmente nuovo o peculiare dal punto di vista delle tecnologie e delle interfacce di comunicazione (si tratta di chat, bacheche condivise, giochi in rete, ecc.). Anche alcune delle funzionalità specifiche che vengono esaltate nei social network (ad esempio la ricerca di persone note) si possono, ovviamente, realizzare anche con altri tipi di sito (motori di ricerca, pagine bianche, ecc.). Certamente se una gran massa di utenti si iscrivono a tale sito, risulterà comunque più facile trovarle. Uno dei fattori di maggior successo di Facebook è poi, probabilmente, l'immediatezza con cui si realizza la comunicazione uno a molti, peraltro già tipica dei blog e dei siti personali. Del resto un altro caso di "social network" di recente successo è "Twitter", che permette in maniera ancora più elastica e veloce di inviare brevi testi ai propri contatti e che per questo motivo ha riscosso molto successo.

Un altro degli aspetti attraenti di queste tipologie di siti di successo consiste nell'integrare i normali servizi di comunicazione e di pubblicazione di materiale su Internet in un unica interfaccia personalizzata. Questo può risultare abbastanza comodo, ma va tuttavia segnalato un rischio: quello di affidare **tutti i propri dati sensibili** e le proprie comunicazioni ad un **singolo soggetto privato** che le può utilizzare a proprio piacimento (lo scopo dichiarato dei gestori dovrebbe essere quello di "vendere" le informazioni per consentire alle aziende di offrire pubblicità mirata ai gusti dell'utente). Occorre sempre ricordare che pubblicando materiale su siti esterni, si accettano dei contratti di utilizzo e di eventuale tutela della privacy che andrebbero considerati con attenzione. Più informazioni si inseriscono in rete più è difficile tutelare la propria privacy, dato che se non si mettono filtri, le informazioni scambiate sono accessibili a tutti (curiosi, spie, datori di lavoro, ecc.). Il successo di questi mezzi, comunque ha creato abbastanza resistenze all'uso improprio delle informazioni, basti pensare che, quando Facebook nel 2008 ha cambiato le condizioni di proprietà, dichiarando che i contenuti inseriti dall'utente diventavano di proprietà dell'azienda, ha dovuto fare velocemente retromarcia. Inoltre sulla piattaforma sono stati inseriti "filtri" che consentono all'utente di limitare l'accesso ai contenuti (vedi Figura 19). In ogni caso è ancora bene ricordare che di tutte le proprie azioni sulla piattaforma rimane traccia sui server aziendali, sicuramente protetta, ma comunque teoricamente recuperabile.

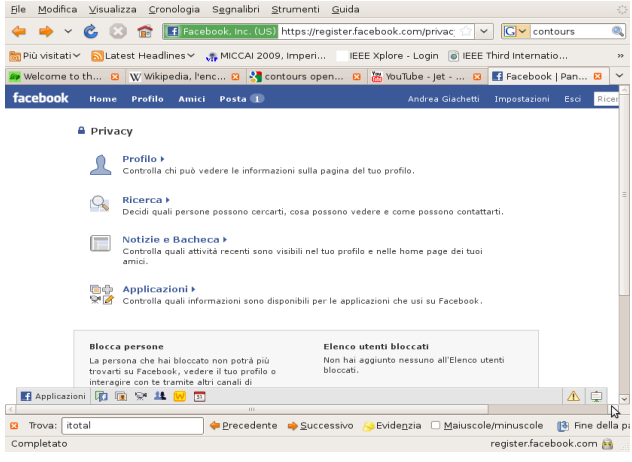


Figura 19: Menu del social network Facebook per gestire le opzioni relative alla privacy dell'utente.

Altro fattore di rischio consiste nel fatto che accedere alle informazioni sulla rete esclusivamente in modo mediato da un unico gestore privato e solo tramite le segnalazioni di "amici" limita molto sia l'esperienza di navigazione e sottopone l'utente a più o meno inconsapevoli censure.

Un ultimo mito da sfatare riguardo queste forme di pubblicazione di contenuti online è che attraverso di

essi si possano facilmente fare conoscere le proprie idee o le proprie opere artistiche (musica, foto). Di fatto, data anche la quantità di iniziative ed utenti, così come è difficile catturare l'interesse di sconosciuti con mezzi tradizionali, lo è altrettanto su queste comunità. A volte viene fatto credere che questo sia più probabile del reale (avrete magari letto sulla stampa notizie su intellettuali diventati noti per un blog, gruppi pop-rock che si sono fatti conoscere su MySpace o YouTube). Salvo rare eccezioni, la realtà è in genere differente: certo, la persona o il gruppo avranno utilizzato tali mezzi per pubblicare le proprie opere, ma, probabilmente, hanno utilizzato altri canali per far si che milioni di persone le abbiano scelte/trovate.

✔ *Nota: social network e memoria*

Forse uno degli aspetti meno percepiti e più rischiosi nell'utilizzo delle bacheche dei social network, è il fatto che in tali siti l'inserimento dei contenuti testuali o delle immagini è fatto per un'interazione immediata con i propri contatti, senza pensare che si tratta comunque di dati che rimangono memorizzati permanentemente nei server dei gestori. È noto che molte società controllino profili e caratteristiche dei candidati a posizioni lavorative su tali siti (secondo uno studio del 2009 del sito CareerBuilding.com, negli Stati Uniti le aziende che usavano questa pratica erano già il 45%), e che quindi quanto inserito per scherzo o dialogando con gli amici può creare seri problemi in futuro. Spesso capita anche di sentire di personaggi pubblici o politici in imbarazzo per foto o frasi scritte con tali mezzi. La "memoria" della rete è un problema serio, e lo è oggi in modo particolare appunto perché utilizziamo questi mezzi per colloquiare in tempo reale senza considerare la persistenza delle informazioni. Alcune aziende hanno anche approfittato di questo fatto e vendono a caro prezzo servizi che cercano di cancellare tutte le tracce lasciate sui server dal cliente. Ma è buona norma, ricordare, quando si pubblica qualcosa in rete, che esso rimarrà memorizzato da qualche parte e che anche se il servizio di pubblicazione è ad accesso limitato, le protezioni poste su tali contenuti possono essere violate.

Wiki

Un ultimo fenomeno che vogliamo qui affrontare è quello dei sistemi di creazione collaborativa di contenuti, detti anche "wiki". Occorre evitare la confusione tra "wiki" (un tipo di strumento) e Wikipedia, che è un particolare sito creato con questo tipo di strumento di creazione e di cui, data l'importanza assunta, ci occuperemo brevemente.

Wiki, dall'Hawaiano "wiki wiki" che significa veloce, è un termine che indica un sistema che permette agli utenti di modificare collaborativamente un contenuto in modo tale che l'inserimento sia agevole (uso di interfacce user friendly), le modifiche siano tracciate e memorizzate e si possa agevolmente ripristinare lo stato del documento precedente se necessario. In pratica questo corrisponde all'usare un particolare Content Management System che garantisca tutto questo (insieme con la sicurezza, la gestione di eventuali conflitti sui file e così via). Il programma che viene usato per generare un wiki viene detto "motore wiki".

Per quanto riguarda la celeberrima **Wikipedia,** essa costituisce un esempio molto bello di utilizzo della creazione collaborativa. Si tratta di un'enciclopedia online che chiunque voglia può contribuire a creare. Basta infatti iscriversi e si possono scrivere o aggiornare le voci dell'enciclopedia. Tutto questo è realizzato tramite un sistema di gestione che prende in genere il nome di "motore wiki" e che nel caso di Wikipedia è il software open source MediaWiki. Questo sistema è basato sulle classiche tecnologie PHP, mySQL cui abbiamo fatto cenno, che

caratterizzano la stragrande maggioranza dei Content Management System (anche i motori wiki rientrano chiaramente in tale categoria).

Come per il caso del software libero (vedi Appendice 1), anche nel caso della creazione collaborativa di contenuti i risultati del modello di sviluppo hanno superato probabilmente ogni aspettativa. Infatti il controllo esercitato dalla comunità che collabora alla realizzazione è molto efficace nell'evitare che si inseriscano dati errati o parziali e gli episodi di boicottaggio o di inserimento di dati tendenziosi sono molto minori di quanto si possa credere. In definitiva, il meccanismo di controllo di massa fa sì che, almeno per quanto riguarda argomenti non controversi, l'enciclopedia sia molto più completa, aggiornata ed equilibrata di molte tra quelle commerciali ed ormai è utilizzata anche troppo come fonte di informazione da parte di giornalisti della carta stampata (non è così raro trovare sugli articoli pubblicati “copia e incolla” di voci wikipediane). Solo per argomenti controversi o con risvolti politici ci possono essere controversie o vandalismi, per evitare questi ultimi si utilizzano, ad esempio, vincoli per riservare la modifica ai soli redattori autorizzati.

L'uso dei motori wiki non è ovviamente limitato a Wikipedia e alle enciclopedie, ma può risultare molto utile per diverse applicazioni, basti pensare alla documentazione di progetti aziendali o di progetti di software libero, oppure alla creazione di archivi di informazione su specifici argomenti (viaggi, letteratura, ecc.). Anche il recentemente discusso sito di pubblicazione di file “segreti”, Wikileaks, è, come suggerisce il nome, gestito in questo modo.

Metafora del sistema operativo, desktop remoti e condivisi

L'evoluzione di Internet ha quindi portato ad una serie di cambiamenti nello stile di interazione con le pagine ed all'affermarsi di tipologie di siti che permettono agli utenti di condividere informazioni, di inserire contenuti multimediali personali, di interagire in maniera più semplice ed efficace e di costruire vere e proprie opere collaborative e pubbliche. Come abbiamo detto questo non corrisponde tanto ad un'evoluzione tecnologica quanto ad un cambio nel modo di partecipare alla creazione di contenuti in rete. L'unico importante avanzamento tecnologico che ha mutato l'aspetto dei siti è legato alle tecniche avanzate come **AJAX,** Asynchronous Javascript and XML, che permettono in pratica lo scambio di informazioni da una pagina web ai server senza ricaricare una nuova pagina. Grazie a tecniche di questo tipo e alla disponibilità di connessioni veloci anche per gli utenti domestici, i siti dinamici cioè le applicazioni web sono diventate interattive ed il browser è in pratica diventato un interfaccia di un enorme **sistema operativo** che mette a disposizione risorse di calcolo e file remoti ed offre in più la comunicazione con gli altri utenti. Con queste tecnologie il browser web diventa come un desktop di un computer il cui archivio dati e le cui risorse di calcolo dono distribuite in tutto il mondo. Vi è quindi la possibilità di spostare online anche le classiche attività lavorative che si svolgono quotidianamente sul proprio PC come scrivere, elaborare dati, archiviare file, ecc. Colossi come Google e Microsoft offrono infatti anche applicazioni web analoghe ai pacchetti Office utilizzabili online, mentre sono attivi sistemi per l'archiviazione on line e condivisione di file. Questo presenta ovviamente vantaggi (possibilità di lavorare da qualsiasi terminale) e svantaggi (necessità di connessione sicura e veloce), in ogni caso apre notevoli opportunità specie per il lavoro in collaborazione e l'e-learning. Un altro tipo di strumento che si sta affermando online per il lavoro è infatti quello degli editor o dei desktop condivisi su cui possono lavorare più utenti remoti in modo coordinato.

3.9 Gestione e ricerca dell'informazione sul web, il web semantico

Abbiamo visto che il web odierno si caratterizza per l'enorme mole di dati archiviati sotto forma di file (pagine) e di tabelle di database sui vari server accessibili, in modo più o meno filtrato, dagli utenti. E' evidente che la vastità di tale struttura rappresenta un'enorme potenzialità, ma anche un serio problema. Oggi in rete possiamo trovare quasi tutte le informazioni di nostro interesse, ma spesso non siamo in grado di raggiungerle.

Come si fa a recuperare l'informazione che ci serve? Come fare a definire cosa è rilevante e cosa non lo è? E' possibile catalogare e strutturare l'informazione del web in modo efficace?

Abbiamo visto che molti siti gestiscono i loro archivi di dati utilizzano le cosiddette basi di dati relazionali: in tal caso è relativamente semplice fare in modo che si possano trovare i dati utili all'interno del sito strutturando in modo corretto il database e usando i linguaggi di interrogazione degli stessi per recuperare i dati.

Questo ovviamente non è possibile quando vogliamo recuperare informazioni dall'intera rete, cioè da siti indipendenti ognuno con le sue pagine ed eventualmente con le proprie basi di dati.

I **motori di ricerca**, come abbiamo visto, cercano una soluzione mediante tecniche di **information retrieval** (disciplina che si occupa, appunto, del reperimento automatico di informazione da collezioni e archivi digitali). Essi basano la ricerca su parole chiave cercando nella collezione dei documenti quelli ritenuti più pertinenti a tali parole.

Per fare questo devono creare degli **indici** del web, realizzati mediante software (detti **crawler** o **spider** o **robot**) che navigano automaticamente tra le pagine raccogliendone i contenuti, che vengono poi analizzati da altri software che ne estrapolano una rappresentazione compatta. In pratica essi elaborano il testo, estraendo le radici delle parole non triviali (escludendo cioè le cosiddette **stop words** come articoli e congiunzioni). Attraverso successive analisi su posizionamento, ruolo e frequenza delle parole cercano poi di stabilire la rilevanza di esse per indicizzare il documento.

Per generare la risposta alle richieste verranno cercate le pagine indicizzate che hanno la maggiore “rilevanza” rispetto alle parole chiave inserite. Questa rilevanza è calcolata mediante opportuni algoritmi che sono un po' il segreto del successo del motore stesso: in qualche modo il successo di Google è stato dovuto all'efficacia dei suoi algoritmi (il cosiddetto PageRank). Tre gli altri “motori” usati sul web possiamo citare AltaVista, utilizzato da Yahoo e “Bing” di Microsoft, che ha soppiantato il precedente “Live Search”.

Il limite di questo metodi di ricerca di informazioni consiste nel fatto che la rilevanza viene calcolata a partire semplicemente dal contenuto testuale dei documenti e dal numero e contenuto dei riferimenti ai documenti trovati su altre pagine (Google attribuisce maggiore rilevanza alle pagine maggiormente citate pesate dall'importanza della pagina citante). Non c'è quindi nessuna comprensione del “significato” del documento stesso per l'utente. Inoltre non c'è accesso alle informazioni nei database dei server, per cui gran parte dell'informazione contenuta in rete non viene utilizzata.

Per i più diffusi motori di ricerca, poi, il risultato finale è una semplice lista di pagine e non un'informazione elaborata che cerchi effettivamente di rispondere alla domanda posta dall'utente o estrapoli la parte informativa dalle varie fonti facendone un sunto. Quest'ultimo

scopo (oltre all'utilizzo dell'informazione nei database) è quello che si è prefisso Stephen Wolfram, che ha creato il nuovo motore di ricerca chiamato Wolfram Alpha che oltre ad utilizzare informazioni recuperate nei database, anziché riportare come risposta dei link ordinati per rilevanza, cerca di rispondere alla domanda posta attraverso complesse elaborazione dei dati recuperati dalla rete. In alcuni casi il risultato è sorprendente, ma chiaramente i campi di applicazione sono per ora modesti ed il compito è notevolmente arduo.

I limiti dell'indicizzazione dei motori di ricerca sono chiaramente legati, in genere, all'assenza di informazioni semantiche sui documenti accessibili in rete. Per trasformare la mole di dati distribuita in rete in una "base di conoscenza" da cui trarre le informazioni mediante semplici richieste in linguaggio naturale, occorrerebbe rivedere la struttura dei documenti stessi, aggiungendo opportunamente etichette e relazioni strutturate tra i vari oggetti. Questa è l'idea del cosiddetto "**Web semantico**" proposta da Tim Berners-Lee, cioè lo stesso creatore del Web. Essa impone che i documenti online siano opportunamente associati a **metadati** cioè informazioni accessorie che ne determino il contesto semantico (annotazioni semantiche) e possano essere usate per creare relazioni complesse tra i documenti stessi. Il web semantico si dovrà basare su linguaggi di descrizione di documenti come XML e di opportune **ontologie** cioè rappresentazioni standard dei concetti e delle relazioni dei domini di interesse.

Il web semantico è oggi agli albori ed è argomento di ricerca, anche se sicuramente l'uso di metainformazione e descrittori associati ai contenuti in rete è già comune ed ha già in parte mitigato i problemi relativi alla catalogazione e ricerca di documenti eterogenei. I metodi di ricerca che si usano, per esempio, all'interno di blog, social network, sistemi di condivisione di foto e video tipo Flickr, YouTube, ecc, di fatto utilizzano come chiave per le ricerche i cosiddetti "tag" inseriti dagli utenti per "marcare" i propri contenuti.

L'etichettatura fatta dagli utenti delle risorse (che a volte viene definita col termine di "**social bookmarking**") può essere vista come un'alternativa interessante a quella automatica realizzata dai motori di ricerca, con il vantaggio di utilizzare appunto etichette relative al significato attribuito ad esse, ma con il limite della necessità di fidarsi delle etichette e delle segnalazioni di utenti che non si conoscono.

Uso efficace dei motori di ricerca

Facciamo infine una breve parentesi "pratica" dando qualche suggerimento utile per "trovare" più facilmente l'informazione di interesse utilizzando i motori di ricerca. Visto che motori come Google si basano sulla ricerca di parole chiave, quelle le cui radici vengono memorizzate nei database (quindi escluse le cosiddette stop words come congiunzioni, articoli e simili), inserendo una sola parola, si otterrà un risultato ragionevole solamente se la parola stessa è molto specifica. In caso contrario, occorrerà specificare meglio la ricerca utilizzando, per esempio, più parole. Se ci si aspetta che il documento da cercare contenga varie parole nomi, meglio inserirli tutti. Le parole possono essere anche legate da connettivi logici. Ad esempio si può specificare che il documento contenga una determinata parola, ma non ne contenga un altra. In genere questo si ottiene anteponendo a quest'ultima parola il simbolo -. Utilizzando Google o Bing, è possibile, per specificare meglio la ricerca, accedere ad un menu apposito (ricerca avanzata, vedi Figura 20) che permette di specificare anche altre opzioni riguardo al documento da ricercare (es. lingua, formato del file, regione di provenienza, data, ecc.). Si possono anche regolare le caratteristiche del filtro (SafeSearch) che elimina i contenuti che possono essere ritenuti inadatti ai minori.

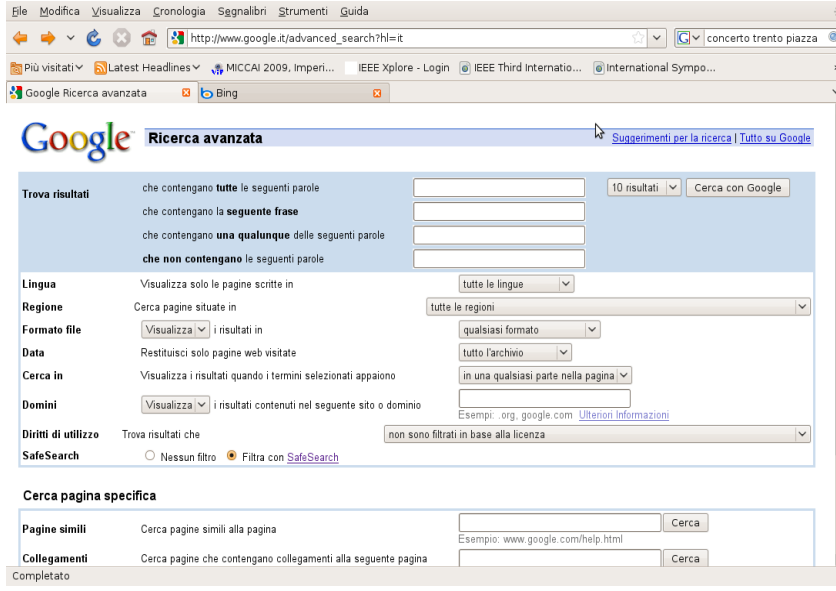


Figura 20: La pagina per la ricerca avanzata di Google.

Per quanto riguarda i risultati, occorre poi tenere conto delle caratteristiche dei motori al fine di averne una corretta interpretazione. Innanzitutto ricordiamo che una pagina sarà trovata dal motore se conterrà il termine (i termini) inserito nella casella di ricerca possibilmente in evidenza o ripetutamente, e se altri siti sono collegati a tale pagina. Se si usano più termini, questi saranno in genere considerati come collegati da un AND logico (cioè verranno trovate le pagine che sono rilevanti rispetto a tutti i termini inseriti. Per considerare un'intera frase come chiave di ricerca occorre inserirla in genere tra virgolette. Se i termini non sono particolarmente specifici, non è affatto detto che il sito che contiene l'informazione da noi cercata sia in testa alla colonna. A volte può, quindi, valere la pena di soffermarsi a vedere un po' di risultati verificando magari anche il dominio del sito e l'estratto del contenuto, anziché cliccare sul primo risultato trovato. Anche per quanto riguarda l'autorevolezza delle fonti si possono fare considerazioni abbastanza simili: se si cercano nomi o parole che generano molto traffico è probabile che effettivamente i siti più rilevanti siano trovati nelle prime posizioni, dato che saranno “linkati” da molti siti autorevoli, che fanno guadagnare molta rilevanza almeno con il PageRank di Google. Viceversa, se si tratta di argomenti che non sono molto seguiti o su cui cui non esistono portali affidabili di riferimento, non è detto che l'algoritmo funzioni altrettanto bene, dato che in rete ci sono una marea di siti molti dei quali contengono ben poco contenuto informativo (o contenuti discutibili) e sono generati soltanto per attrarre traffico e pubblicità e che possono confondere molto i sistemi automatici. Per esempio, se provate a cercare un regista o uno scrittore, un personaggio storico, della fiction, ecc., facilmente troverete ai primi posti della ricerca siti affidabili con notizie utili: Wikipedia, portali del cinema o della letteratura, fan club ufficiali, ecc. (vedi Figura 21) Se cercate però una ricetta di cucina o un venditore di auto, ci sono molte più difficoltà in quanto in genere verranno innanzitutto spesso presentati due colonne di risultati di cui una riservata ad inserzionisti a pagamento, inoltre i risultati porranno nelle prime posizioni spesso siti inaffidabili o “siti civetta”, realizzati al solo scopo di attrarre con l'inganno traffico per realizzare qualche soldo di pubblicità (Figura 22). Meglio allora

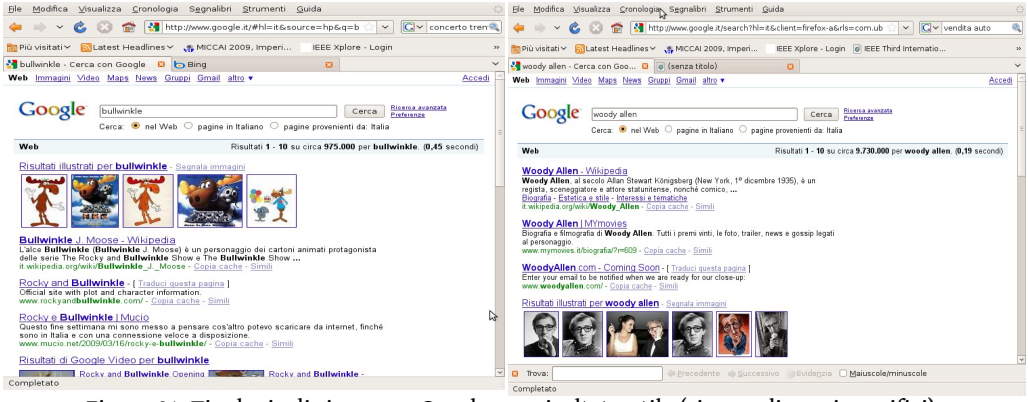


Figura 21: Tipologie di ricerca su Google con risultato utile (ricerca di nomi specifici).

consultare un buon libro di cucina o le pagine gialle! Anche per questo un altro suggerimento abbastanza ovvio è quello di utilizzare i motori di ricerca generalisti solo se non si hanno indicazioni più dettagliate su dove cercare l'informazione: se per esempio vogliamo cercare documenti multimediali come immagini o video possiamo subito passare ai motori di ricerca specifici, se cerchiamo un'informazione enciclopedica possiamo rivolgerci a Wikipedia, se ci interessano informazioni su un film si possono utilizzare database specifici (come l'Internet Movie Database) o altri siti specializzati. Sempre meglio perdere un po' di tempo a priori per pensare a come e dove potremmo trovare l'informazione che ci serve piuttosto che impigrirsi digitando sempre la parola su Google, che è sicuramente uno strumento utile, ma di cui è prudente non abusare. In effetti la stessa Google ha ormai diversificato la sua offerta affiancando al motore di ricerca generalista, strumenti atti a fornire informazioni e servizi specifici per i diversi interessi. Ad esempio, per la ricerca della bibliografia scientifica, si può ricorrere a “Google Scholar” che indicizza specificatamente articoli accademici, se si cercano notizie, si può ricorrere a “Google News”, e così via. Le problematiche legali dell'uso di motori di ricerca o altri siti similari che si collegano a contenuti protetti da copyright sono spesso discusse sui media, anche riguardo a presunte cause legali tra grandi aziende. Occorre, tuttavia, tener conto del fatto che i motori di ricerca non forniscono all'utente copia dei contenuti dei siti, ma solo il collegamento alle pagine ospitate dal reale proprietario. Inoltre gli standard web prevedono che sia semplice, per chi vuole, impedire ai motori di ricerca di indicizzare i propri contenuti: basta infatti includere nelle cartelle dei propri server dei file specifici ed i file non saranno indicizzati dai programmi (robots) dei motori di ricerca.

3.10 Gestione pratica di un sito Web

Abbiamo visto, dunque, cos'è un sito web e quali sono le tecnologie con cui si possono realizzare i vari tipi di sito. Abbiamo anche visto che, molti siti moderni, quelli che spesso consideriamo come il “Web 2.0” consentono agli utenti di inserire contenuti personali su proprie aree personali accessibili da tutti, utilizzando le varie piattaforme dei social network, blog, eccetera. Per molte delle attività che possono interessare il grande pubblico (diffondere testi o foto, presentare la propria persona e le proprie attività) questi spazi gestiti da aziende esterne sono più che sufficienti.

Tuttavia, se di vuole creare un proprio sito completamente autonomo da tali piattaforme e tali

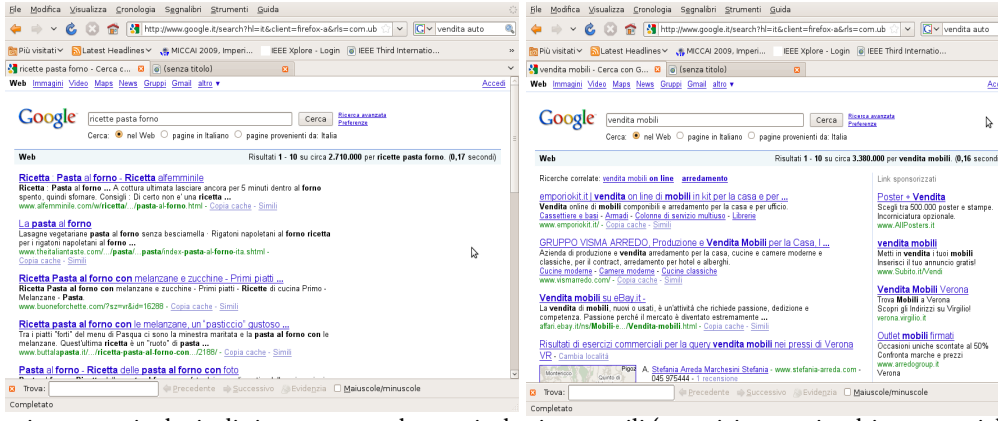


Figura 22: Tipologia di ricerca su Google con risultati poco utili (generici o con risvolti commerciali): molti dei risultati sono siti "civetta" o segnalazioni a pagamento (colonna a destra).

aziende, avere il proprio indirizzo, avere piena libertà di gestione dei contenuti, è necessario procedere autonomamente, creando i propri ipertesti e rendendoli pubblici inserendoli in un'area pubblica di un server HTTP.

Nell'ipotesi più semplice, quindi, per realizzare un sito autonomo, occorre avere uno spazio a disposizione su una macchina ospite (host) accessibile in rete e su cui sia attivo un programma server HTTP e copiare sull'area del suo disco accessibile dall'esterno il codice delle pagine HTML da noi create, insieme con i dati aggiuntivi collegati (file immagine, suoni, ecc.). Siccome l'utente privato in genere non dispone della possibilità di gestire un host direttamente connesso alla rete ed installarvi il programma server, la soluzione che si adotta è tipicamente quella di affittare dello spazio da un cosiddetto **provider** di **spazio web**.

Se non si richiede che questo spazio sia indirizzato da uno specifico dominio, è possibile anche ottenere questo spazio gratuitamente (esistono molti provider gratuiti di spazio web, in Italia quello con più clienti è www.altervista.org). Basta collegarsi al sito del provider, iscriversi, inserendo una casella e-mail valida, e scegliere le opzioni. Notare che i siti che offrono spazio non sono tutti uguali: alcuni inseriscono automaticamente pubblicità sulle pagine, alcuni no; alcuni supportano la programmazione server side, altri no, insomma occorre scegliere il provider adatto alle proprie esigenze. Se si è disposti a pagare, si possono avere servizi migliori: evitare la pubblicità indesiderata, associare un dominio di propria scelta, avere assistenza tecnica e così via.

Per creare comunque il proprio sito web, una volta iscritti, basta accedere all'area personale del sito del provider ed usare gli strumenti forniti per trasferire le pagine create dal proprio PC al server oppure quelli, spesso presenti, per creare pagine in modo assistito lavorando direttamente sull'interfaccia web. Se si trasferiscono i file dal proprio PC, l'opzione migliore è poi quella di utilizzare un client FTP (il protocollo di Internet per il trasferimento file, utilizzando l'indirizzo indicato dal provider).

Creare ipertesti scrivendo codice HTML non è particolarmente complicato, e lo vedremo in sufficiente dettaglio nel capitolo 4 . Se si vogliono creare siti piuttosto complessi o se addirittura si vogliono realizzare i siti "attivi" visti in precedenza che contengano cioè programmi in grado di realizzare compiti vari, le cose diventano certamente molto più complicate. Per questo, come visto in precedenza, le persone che semplicemente vogliono pubblicare propri contenuti e

utilizzare modalità standard di interazione sono passate dalla creazione di siti personali alla gestione di blog o di profili sui social network. Tuttavia, se si vuole creare un sito indipendente dotato di caratteristiche "attive" e di grafica sofisticata senza dover scrivere codici complessi e senza ricorrere a consulenze professionali, esiste una soluzione, cioè quella di gestire personalmente, sul proprio spazio affittato dal provider, i sistemi di gestione utilizzati per le tipologie di sito classico, cioè i cosiddetti Content Management System di cui già abbiamo parlato nel capitolo 3.3.

3.11 Gestione di Content Management System

Abbiamo già detto che i Content Management System sono pacchetti software che, installati sulla macchina che opera come server web, consentono di realizzare tipologie standard di siti dinamici che utilizzano la programmazione server side, senza dover scrivere codici complessi, ma configurando i programmi, selezionando le opzioni per grafica e servizi ed inserendo i contenuti mediante interfacce semplici.

Questi programmi possono, in genere, essere installati da qualunque utente sulle proprie aree web ed utilizzati nello stesso modo in cui li utilizzano i professionisti per creare i propri siti. Questa operazione richiede in genere un po' di esperienza e conoscenze di base sulle tecnologie web, HTML, e altre cose che vedremo nel capitolo seguente, ma è certamente alla portata anche di persone che non fanno gli informatici di professione.

I pacchetti di Content Management più noti, come Joomla, Drupal, Wordpress o Plone, sono software open source e possono essere scaricati gratuitamente dai rispettivi siti web. Se si seguono le istruzioni e si affitta uno spazio web da un provider che rispetti i requisiti tecnici richiesti dall'applicazione (ad esempio il supporto PHP o la presenza di un database MySQL), è possibile quindi per chiunque installare il sistema di gestione sul proprio spazio web e gestire il sito esattamente come fanno i webmaster dei più noti siti istituzionali o commerciali che, in gran parte, utilizzano questi strumenti per realizzare i loro prodotti.

Molti provider di spazio web propongono anche come opzione l'utilizzo di tali sistemi per creare i propri siti senza doversi preoccupare della loro installazione e gestione (l'utente accede direttamente al pannello di controllo di un CMS installato sul server del provider. Non è intenzione di questo testo addentrarsi nella descrizione dei singoli programmi o della loro configurazione e personalizzazione, quello che vogliamo qui sottolineare è che, per chi vuole fare un uso "professionale" del web gestendo siti dedicati (per esempio il sito della propria scuola o della propria società) senza ricorrere a consulenze di tecnici specializzati, l'utilizzo di questi software è sicuramente la soluzione più semplice, ed è effettivamente anche la più praticata. Non occorrono, infatti, competenze informatiche eccezionali per il loro uso, ma solo un po' di dedizione per imparare ad usare le interfacce dei sistemi e, soprattutto, molto lavoro per la creazione dei contenuti. Una cosa che spesso chi vuole realizzare e commissionare siti web dimentica, infatti, è che il spesso il lavoro maggiore legato ad un sito non è tanto la sua creazione, quanto il suo aggiornamento costante, che è assolutamente necessario se si vuole che esso sia visitato ed utilizzato.

Il web di oggi pullula di siti (ma anche di pagine di social network, blog, forum e quant'altro), che non vengono mai aggiornati e non vengono mai visitati. Quando si crea uno spazio in rete, insomma, è necessario pianificare adeguatamente le risorse necessarie al suo mantenimento, altrimenti lo sforzo iniziale risulterà del tutto inutile.

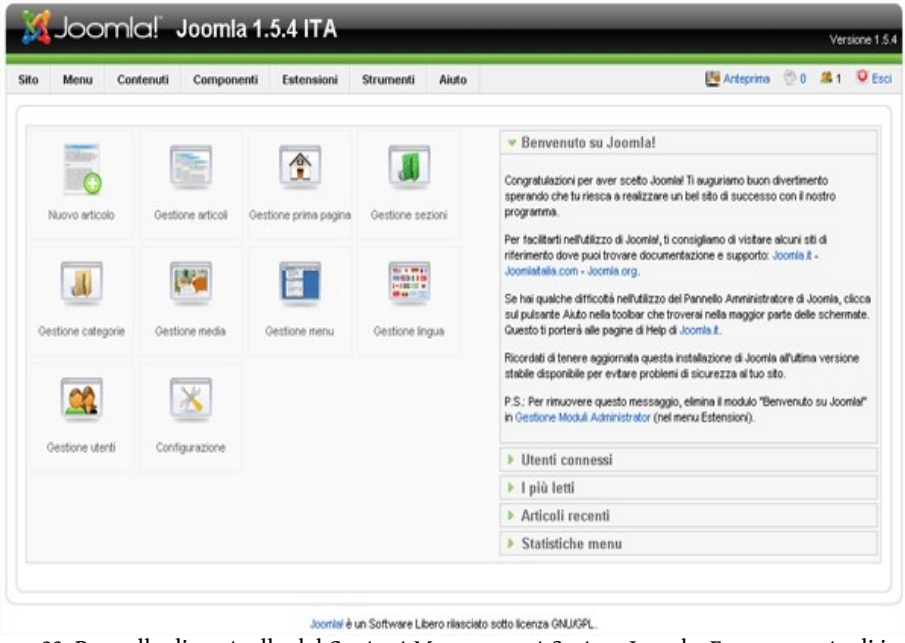


Figura 23: Pannello di controllo del Content Management System Joomla. Esso consente di inserire e gestire in modo intuitivo i contenuti di un sito dinamico.

4 Creare contenuti per il web: (X)HTML e CSS in dettaglio

4.1 HTML: storia ed evoluzione

Anche se, come abbiamo visto, non è necessario conoscere il meccanismo di codifica degli ipertesti per inserire contenuto sul web, vale la pena di andarlo a vedere da vicino, sia perché è un ottimo esempio per avvicinarsi alla scrittura di codice informatico, sia perché ci consente anche di capire meglio come funzioni il browser e come si inseriscano i contenuti multimediali nelle pagine web.

Del resto, anche se si utilizzano i Content Management System come amministratori o semplici utenti per creare i propri contenuti web, è decisamente utile sapere come è fatto il markup (linguaggio di marcatura) che genera le pagine a basso livello, anche perché molte opzioni di configurazione del layout grafico disponibili in tali sistemi implicano l'utilizzo di codice scritto in HTML o CSS (vedi nel seguito).

L'invenzione del web, come abbiamo visto è strettamente correlata alla realizzazione di un meccanismo per codificare gli ipertesti, cioè per creare dei testi con annotazioni che ne determinino la visualizzazione sul browser ed il "significato" da attribuire ad ogni parte. Quando Tim Berners-Lee creò il meccanismo al CERN di Ginevra, negli anni '90, creò, quindi, anche il linguaggio per la codifica, che chiamò **HTML (HyperText Markup Language)**.

Esso è un linguaggio di codifica che si dice di **marcatura** o **markup** proprio perché non dà "istruzioni" su operazioni da eseguire come i linguaggi di programmazione, ma semplicemente marca le parti di testo dando loro determinati significati.

Il linguaggio venne creato sulla base di un altro linguaggio dagli scopi più generali detto SGML (Standard Generalized Markup Language). Tratti caratteristici del linguaggio sono la sua codifica come testo, che lo rende indipendente da hardware e sistema operativo e, appunto, l'introduzione dei collegamenti ipertestuali che possono essere aggiunti alle parti del documento.

Il linguaggio originale ideato da Berners Lee, dopo la diffusione del web e lo sviluppo del mercato dei browser, andò incontro ad una rapida evoluzione, tuttora in corso.

L'idea fondamentale, come vedremo meglio in seguito, era di utilizzare dei marcatori (tag) che identificassero determinati "elementi" del documento da interpretare in maniera peculiare (p.es. titoli, paragrafi, tabelle) e da visualizzare con particolari attributi, modificabili.

Il linguaggio originale aveva 22 elementi di cui 13 sono rimasti fino ad oggi. A metà anni novanta, Netscape e Microsoft, che producevano i browser più diffusi, avevano aggiunto al linguaggio varie opzioni aggiuntive proprietarie, che rendevano più attraenti i siti ma anche più difficile il lavoro di chi li realizzava, data la necessità di evitare le incompatibilità o di realizzare versioni multiple per adattarsi ai diversi browser. Per questo l'organismo di standardizzazione del web, **W3C**, fondato nel 1994 dallo stesso Berners-Lee, cominciò a definire in modo sempre più completo lo standard cui i browser avrebbero dovuto uniformarsi, includendo in esso tutte le modifiche utili e di successo introdotte con gli elementi proprietari. Le evoluzioni ufficiali dello standard furono: l'HTML 2.0 (1995), l'HTML 3.2 (1997), che incluse i principali elementi proprietari di Netscape, l'HTML 4.0 (1997) che introdusse la

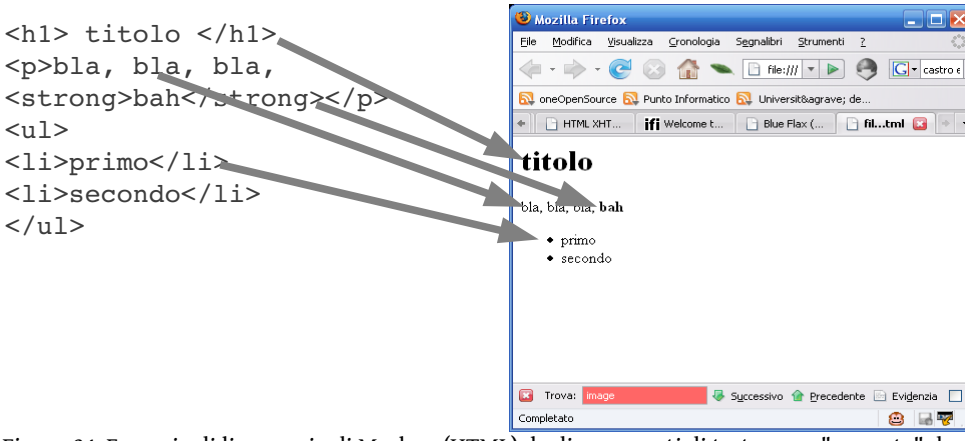


Figura 24: Esempio di linguaggio di Markup (HTML): le diverse parti di testo sono "marcate" da appositi elementi (tag) in base ai quali sono visualizzati in modo diverso nella finestra del browser

dichiarazione di tipi diversi di documento (strict, transitional, frameset) all'inizio del codice delle pagine e, pur includendo altri elementi ed attributi proprietari dei browser per la gestione degli effetti grafici cominciò a definirli "deprecated", cioè disapprovati. Infatti si stava allora introducendo un nuovo modo di attribuire l'aspetto grafico alle pagine, con l'uso dei fogli di stile a cascata (Cascading StyleSheets, CSS), che vengono codificati con un linguaggio a parte e collegati ai documenti HTML o XHTML (vedi cap. 4.11). L'ultima versione ufficiale di HTML (4.01) risale al 1999, dopodiché si tese a sostituire il linguaggio HTML con una nuova versione (XHTML) molto simile ma riscritta rispettando i vincoli più stretti del meta-linguaggio XML, in modo da rendere anche più facile l'interpretazione dei documenti stessi (HTML, ponendo meno vincoli di correttezza del codice è molto più difficile da interpretare correttamente). La versione attuale di XHTML è la 1.1 ed è il linguaggio cui si dovrebbero conformare i siti attuali, mentre evoluzioni sia del vecchio HTML (5.0) che di XHTML (2.0) sono ancora in fase di discussione da parte dei rispettivi "Working Groups".
L'esistenza di tutte queste versioni ed evoluzioni potrebbe creare grossa confusione; di solito però non ce ne rendiamo conto quando utilizziamo il web, in quanto i browser moderni sono in grado di interpretare abbastanza correttamente tutti i differenti linguaggi in modo da garantire la possibilità di usufruire di tutto il materiale attuale e passato.

4.2 Elementi di base di un documento HTML o XHTML

Le pagine HTML o XHTML, (dato che le cose che diremo valgono in genere per entrambi useremo spesso indifferentemente i termini o useremo la scrittura (X)HTML per indicare entrambi) sono quindi composte da testo con elementi marcatori, come dice il termine stesso "markup" dell'acronimo. Sono quindi abbastanza semplici da comprendere e codificare.

La differenza tra un linguaggio di marcatura ed uno di programmazione è quindi evidente: nel linguaggio di marcatura non diamo una serie di istruzioni alla macchina per risolvere un problema, ma vogliamo dare solo indicazioni sulla struttura del testo contenuto in un documento per strutturarne la visualizzazione.

Le principali forme di marcatura che caratterizzano le parti del documento si chiamano "**elementi**", "**attributi**" e "**valori**".

Gli elementi definiscono il “significato” da attribuire alle varie sezioni del testo (ad esempio che una parte corrisponde a un titolo, una a una tabella, ecc.). Ogni elemento è contraddistinto da un cosiddetto “tag” di apertura, con il nome dell'elemento circondato dai simboli < e >, ad esempio `<html>` e da un corrispondente tag di chiusura, identico a parte il simbolo “/”, es. `</html>`. Tutto ciò che è compreso tra i due tag (testo o altri elementi) viene interpretato come facente parte della struttura indicata dal nome.

All'interno del tag di apertura, si possono indicare eventuali attributi e valori che possono caratterizzare l'elemento (es. dimensione di una tabella, indirizzo di un collegamento ipertestuale, URL di un immagine da inserire, ecc.). Vedremo meglio in seguito la sintassi, ma in pratica ogni elemento è caratterizzato da un insieme ben definito di attributi che possono assumere soltanto determinati valori.

Non tutti gli elementi da marcare, però, implicano la presenza di testo o altri elementi all'interno, esistono anche i cosiddetti “elementi vuoti”, come per esempio “img”, che contiene attributi e valori che indicano file e formattazione di un immagine da inserire e non include testo:

```
<img src=”pippo.jpg” alt=”pippo” width=”200” />
```

Nel primo HTML gli elementi vuoti avevano solo il tag di apertura e non quello di chiusura. In XHTML la chiusura dei tag è invece obbligatoria, e deve essere effettuata scrivendo esplicitamente il tag di chiusura (`<img ... ></img>`) o utilizzando la forma abbreviata equivalente `<img ... />`, come nell'esempio sopra.

Un altra regola resa più rigorosa in XHTML è quella relativa all'annidamento: teoricamente se un tag si apre all'interno di un altro, vuol dire che è in relazione di discendenza da esso (“figlio”) e deve essere del tutto contenuto in esso, quindi il tag di chiusura del figlio deve essere inserito prima di quello del padre. HTML tuttavia tollerava annidamenti scorretti, del tutto proibiti in XHTML. Altre differenze tra HTML e XHTML sono l'obbligo in XHTML di utilizzare le lettere minuscole per i nomi degli attributi (`<html>`, non `<HTML>`!), l'obbligo delle virgolette per i valori degli attributi, l'obbligo di inserire un valore per ogni attributo (cosa non necessaria in HTML, ove si potevano avere casi come `<option selected>test</option>`).

Al di là delle lievi variazioni e della differente rigidità sintattica, l'utilizzo dei principali elementi è sostanzialmente invariato nelle ultime versioni di HTML e XHTML.

Gli elementi di (X)HTML possono essere **inline** o **block-level**. Gli elementi inline denotano oggetti che si inseriscono all'interno del flusso del testo (senza andare a capo) e devono quindi contenere all'interno solo testo od altri elementi inline, come `<em></em>` (testo enfatizzato) e il già visto `<img>`:

```
...<em>pippo</em>...<img src=”pippo.png” alt=”pippo”/>...
```

Gli elementi block-level invece denotano oggetti a sé stanti, separati dal flusso del testo, che in assenza di opzioni di posizionamento vengono separati andando a capo prima e dopo, es.: `<p></p>` (paragrafo), `<div></div>` (divisione di testo), `<table></table>` (tabella) ecc. Una categoria a parte di elementi riguarda le liste, che si comportano in modo particolare e

che vedremo nel seguito.

Nei capitoli seguenti vedremo, a scopo di esempio (rimandiamo invece ai vari manuali che si trovano anche online per chi volesse conoscere bene tutti gli elementi/attributi/valori del linguaggio), i principali elementi del markup e come questi vengano utilizzati per inserire tutti i tipi di contenuto ed ottenere i vari effetti che siamo abituati a vedere nelle pagine web. Prima, però, vediamo meglio la strutturazione del codice di una singola pagina.

4.3 Struttura di un documento HTML

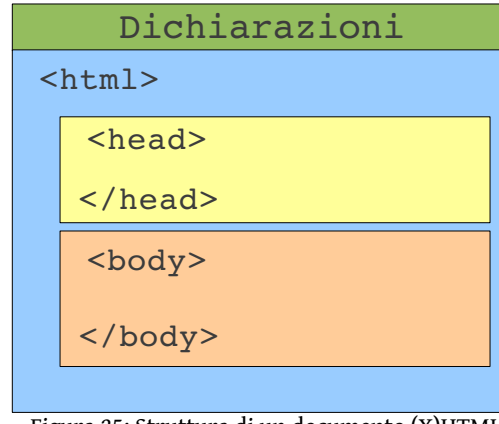


Figura 25: Struttura di un documento (X)HTML

Ogni elemento del documento deve rispettare quindi delle regole di sintassi ben precise, ma anche l'intero documento deve essere organizzato secondo una struttura definita più o meno rigidamente dalle specifiche del linguaggio. L'organizzazione base della pagina (X)HTML è quella in Figura 25. All'inizio del documento possono (devono nel corrente XHTML) essere presenti delle dichiarazioni, che servono ad indicare al browser a quali specifiche il documento si conforma. La loro struttura può risultare effettivamente un po' complessa, ma in realtà non è necessario ricordarsi a memoria i vari tipi di dichiarazioni di tipo per creare pagine web corrette, in quanto si possono tranquillamente copiare da modelli, manuali o dal sito ufficiale del W3C. La dichiarazione di tipo è stata introdotta dalla versione 4.01 di HTML ed in XHTML è obbligatoria. Per quanto riguarda XHTML (stessa cosa in HTML 4.01), esistono tre differenti tipi di documento ammissibile:

- strict (sintassi rigida)
- transitional (sintassi che tollera l'uso di elementi e attributi non più in uso, ed è la più diffusa)
- frameset (con supporto dei cosiddetti frame, sistema per gestire l'inclusione di pagine multiple in unica visualizzazione, ormai sostanzialmente in disuso).

A ciascuno di essi, sia per XHTML che per HTML 4.01 corrisponde una specifica dichiarazione di tipo. In Figura 26 sono mostrate le tre diverse dichiarazioni di tipo in XHTML. In tali dichiarazioni la prima riga è opzionale ed indica che il documento risponde alle specifiche XML (quindi sarà XHTML) e che il tipo di codifica del testo è UTF-8 (Vedi Appendice 2). La dichiarazione di tipo contiene il collegamento all'URL del W3C ove si trova la definizione ufficiale delle specifiche di linguaggio (l'estensione .dtd significa “Document Type Definition”). La dichiarazione di tipo influenza la visualizzazione del documento sul browser, dato che l'interpretazione della stessa seguirà le regole specifiche del DTD. Se questo non è indicato, il browser usualmente interpreta le pagine in una modalità particolare (quirks) in cui tenta ugualmente di visualizzare il contenuto usando le regole dei vecchi browser. Copiata la dichiarazione di tipo a cui ci si intende uniformare, per il resto un documento (X)HTML contiene una serie di elementi definiti come abbiamo visto nel paragrafo precedente, annidati in

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Opzionale: dichiara che il documento è XML e la codifica UTF-8

PUBLIC corrisponde al fatto che la DTD è definita universalmente e non localmente

Figura 26: Le tre dichiarazioni di tipo in XHTML 1.0

una struttura ad albero. La struttura deve contenere obbligatoriamente alcuni elementi base: tutti i tag, infatti, sono contenuti all'interno dell'elemento HTML (aperto all'inizio e chiuso alla fine del documento). All'interno dell'elemento HTML si aprono e si chiudono due elementi che definiscono le due principali sezioni del documento stesso: l'intestazione `<head></head>` che conterrà, ad esempio, metainformazione sulla pagina, il titolo della stessa, eccetera, e il corpo della pagina `<body></body>`, che contiene in pratica tutti gli elementi che definiscono la pagina ipertestuale. Un esempio può essere visto in Figura 27.
L'annidamento dei tag HTML definisce naturalmente una struttura ad albero ove la radice è HTML, mentre gli altri elementi sono in rapporto di discendenza (Figura 28). Occorre quindi attenzione al fatto che molti attributi di visualizzazione vengono ereditati dai genitori in questa discendenza. Anche se versioni passate di HTML accettavano annidamenti errati degli elementi è comunque buona norma stare molto attenti a chiudere gli elementi stessi nel corretto ordine (cioè chiudere prima quello aperto dopo). Come accennato in precedenza, i browser sono molto tolleranti non solo all'uso di vecchie versioni di (X)HTML, ma anche a errori di codifica, molto frequenti in XHTML "strict". In caso di errore, infatti, i browser non si bloccano, ma passano a

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1
transitional.dtd">
<html>
<head>
<title>La mia pagina</title>
</head>
<body>
<img src="foto.jpg" alt="faccia"/>
<p>testo<em> evidente</em></p>
</body>
</html>
```

Figura 27: Esempio di semplice codice HTML con alcuni elementi standard.

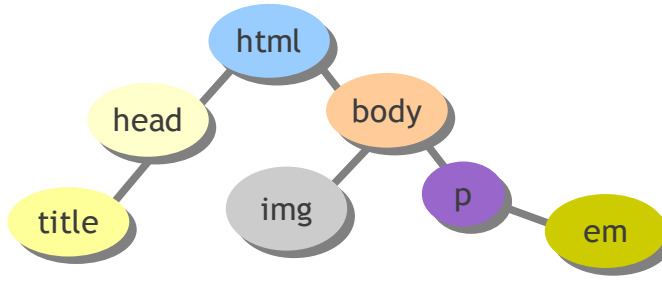


Figura 28: La struttura ad albero determinata dall'annidamento degli elementi del codice nella figura precedente.

una modalità di interpretazione più tollerante (quirks mode) per visualizzare il contenuto. Questo non significa, tuttavia, che non si debba stare attenti alla correttezza del codice che si realizza, particolarmente importante se si vogliono realizzare contenuti web facilmente accessibili. Per verificare se il proprio codice è corretto, è possibile ricorrere a strumenti di controllo automatici detti validatori. Essi controllano se il codice inserito è conforme al DTD relativo alla tipologia di documento creata, segnalando gli eventuali errori. Il più noto è ovviamente quello ufficiale del W3C, accessibile al sito: ***http://validator.w3.org/.***

Gli elementi principali del markup

Abbiamo detto che l'elemento radice **<html>** contiene sempre due sottoelementi principali, **<head>** e **<body>**. Descriviamoli quindi in maggior dettaglio. Head rappresenta l'intestazione del documento, quindi conterrà metainformazione riguardante lo stesso e non gli elementi che ne caratterizzano il contenuto.

Esso può contenere annidato l'elemento **<title>:** questo conterrà il titolo del documento, usualmente visualizzato dai browser come titolo della finestra. Anche se generalmente non notato durante la navigazione, questo titolo assume una certa importanza in quanto apparirà poi nei menù cronologia e preferiti del browser e può essere usato dai motori di ricerca. Un altro elemento annidabile in head è **<meta>**: come suggerisce il nome, questo serve a inserire informazione riguardante il documento, come la codifica dei caratteri (se non dichiarata in XML) o per aiutare o impedire l'indicizzazione sui motori di ricerca.

L'elemento **<body>** contiene invece la descrizione di tutto ciò che verrà visualizzato sulla finestra del browser: titoli, paragrafi, tabelle, caselle di testo, immagini e oggetti multimediali. I titoli di paragrafo, che appaiono tipicamente su righe separate e con caratteri di dimensione grande, sono inseriti gli appositi elementi del markup, da annidare nel **<body>.** Questi elementi sono **<h1></h1>** per titoli di livello 1 (principali)**,<h2></h2>** per il secondo livello e così via. La Figura 30 mostra un semplice documento HTML con titoli e testo. Per inserire un testo sotto il titolo, l'elemento block level che si utilizza è ** **, che definisce un blocco separato di testo da inserire nel flusso della pagina (paragrafo). Se non si utilizzano specifici attributi (che erano stati poi introdotti nel markup per variare lo stile) e non si utilizzano i fogli di stile per la formattazione degli elementi, l'apparenza dei caratteri dei vari elementi è stabilita dalle opzioni del browser. Non è detto, quindi, che la visualizzazione sia identica in tutti i browser ed è possibile variarla agendo sulle opzioni di menù dei vari Explorer,

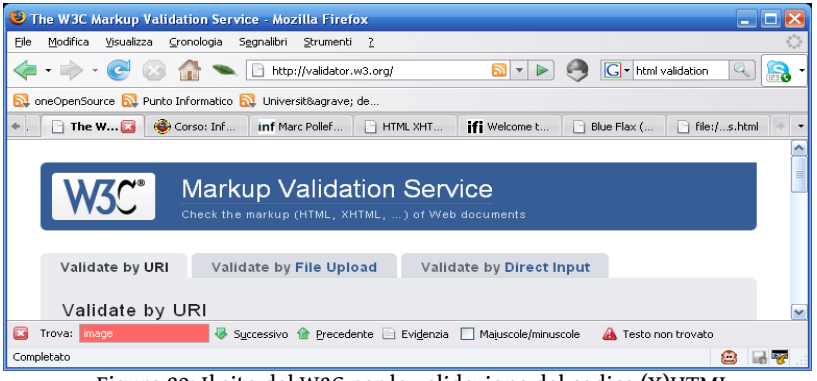


Figura 29: Il sito del W3C per la validazione del codice (X)HTML.

Firefox, ecc. Questo è ragionevole, considerando che HTML è, appunto, un linguaggio di marcatura, che non si dovrebbe occupare della parte di "rendering" grafico, ma solo della strutturazione logica del documento. Andando avanti con gli elementi più usati, occorre dire che gli interpreti HTML non considerano spazi bianchi e a capo, per ottenere i quali occorre quindi ricorrere a codici. Per inserire uno spazio bianco è necessario utilizzare una cosiddetta "sequenza di escape", inserendo i caratteri **` `**. Per andare a capo col testo occorre inserire l'elemento vuoto **`<br/>`**. Per inserire una linea per dividere parti di testo si può usare l'elemento **`<hr/>`**. `<hr/>` è in realtà un elemento "deprecated" anche se ancora molto usato. Esso ha attributi che permettono di modificarne l'aspetto, es. `"size=n"`, che permette di indicarne lo spessore in n pixel o `noshade="noshade"` che rimuove l'effetto di ombreggiatura. Siccome come vedremo è possibile applicare a sezioni di documento specifiche istruzioni di stile con i CSS, è possibile poi inserire degli elementi che hanno il solo scopo di includere queste sottoparti. Essi sono **`<div>....</div>`** per blocchi di contenuto (block level) separati e successive e **`<span>...</span>`** per blocchi di contenuto inline dentro altre sezioni. Per poterli selezionare ad essi si possono applicare specifici attributi come l'identificativo (id) e la classe (class). Un altro attributo generico degli elementi è name, che definisce una stringa di testo che verrà anche visualizzata quando il cursore passa sopra all'elemento.

```
<h1> Fondamenti di informatica
</h1>
<p> In questo corso studieremo
molto</p>
<h2> Perché? </h2>
<p> Per passare l'esame</p>
```

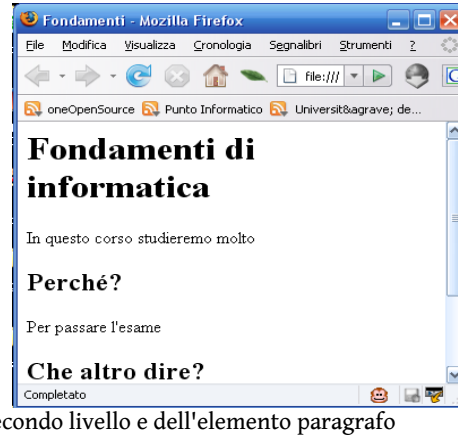


Figura 30: Esempio di uso di titolo di primo e secondo livello e dell'elemento paragrafo

È noto che una buona pratica nello sviluppo di qualunque tipo di codice consiste nel commentare lo stesso in modo da renderlo più comprensibile a se stessi ed agli altri. Nei linguaggi di programmazione più noti, ad esempio, esistono speciali combinazioni di caratteri che delimitano parte del codice che non deve essere compilato o interpretato, ma serve solo come nota a margine dell'autore. Anche in (X)HTML esistono i commenti e la sintassi è la seguente:

```
<!-- commento.... -->
```

Attenzione, però: a differenza di quanto accade nei programmi compilati, i commenti aggiunti all'(X)HTML sono visibili all'utente finale (attraverso l'opzione visualizza sorgente pagina del browser), per cui è opportuno evitare eccessi.

4.4 Inserire immagini

Nei documenti (X)HTML è possibile inserire immagini attraverso l'elemento **<img>**. La sintassi è semplice: si tratta di un elemento vuoto, che si apre e si chiude allo stesso tempo e in cui l'immagine da inserire è indicata mediante l'URL nell'attributo obbligatorio **src.** L'immagine può essere ridimensionata a piacere attraverso gli attributi dimensionali **width** e **height**. Le dimensioni si indicano in pixel o in percentuale sull'elemento che contiene il tag. Si noti che le dimensioni non sono obbligatorie: in caso siano assenti l'immagine verrà visualizzata con il numero di pixel originale. L'uso dei parametri width ed height può essere utile anche se essi coincidono con i valori del file collegato, in quanto velocizzano in tal caso la visualizzazione sul browser. L'uso di valori di dimensione inferiori a quelli dell'immagine originale è invece poco efficiente: varrebbe la pena, in tal caso, creare una versione ridotta dell'immagine per ridurre il tempo di trasferimento del file. Altri attributi di img sono border, che permette di inserire un bordo intorno all'immagine stessa e **alt**, obbligatorio, il cui valore deve essere una descrizione alternativa dell'immagine ad uso dei browser testuali o vocali. La descrizione viene, in genere, visualizzata anche come testo popup (cioè a comparsa) quando il puntatore staziona

```
<body>
<img align=”right” text-align=”top”
src="mare.jpg" width="320" heigth="427"
alt="mare"/> Ho tanta voglia di andare al
mare!
</body>

<body>
<img align="right" src="mare.jpg"
width="100"
alt="mare"/>
 Ho tanta voglia di andare al mare!
</body>
```

Figura 31: Esempi di inserimento di immagine in documento (X)HTML

sopra l'immagine. In Figura 31 si vede un esempio dell'uso di img con vari attributi. La posizione (es. left, center, right) si può controllare con align e vi sono opzioni anche per lo scorrimento del testo, tuttavia esse sono in disuso (e "deprecated") essendo meglio usare le regole di stile CSS. Per l'elenco completo delle opzioni si rimanda ai manuali.

4.5 Formattazione del testo

Formattazione è una parola usata per definire il modo in cui si attribuisce lo stile grafico a un (iper)testo. Come abbiamo detto, durante l'evoluzione di (X)HTML si è cercato di separare lo stile dal contenuto mediante l'uso dei fogli di stile (Cascading StyleSheets o CSS) che vedremo in seguito e che rappresentano sicuramente il modo migliore di gestire la formattazione. All'inizio però per creare gli effetti grafici di impaginazione e modifica del testo si usavano appositi elementi e coppie attributi/valori HTML. Essi possono naturalmente ancora oggi essere usati, anche se alcuni di essi sono stati dichiarati "deprecated" dal W3C, cioè ne viene ufficialmente sconsigliato l'uso. Vediamone i principali.

Per modificare l'aspetto del testo, si può usare l'elemento `<b>`, come ad esempio in `<b>testo</b>` per usare il grassetto (bold), oppure `<i>` per il corsivo (italic) (`<i> testo </i>`). In alternativa esistono anche `<strong></strong>` e `<em></em>` che generano effetti analoghi.

Per variare le dimensioni esistono gli elementi `<big>...</big>` e `<small>...</small>`, che se ripetuti moltiplicano il loro effetto. L'effetto di questi elementi può essere visto in Figura 32. Il tipo di carattere, cioè il cosiddetto **font**, visualizzato dipende se non specificato dalle opzioni del browser. Oggi, come vedremo, si possono scegliere con grande libertà i font, ma all'inizio non vi era questa possibilità. Per modificare il tipo di carattere senza i CSS, si possono usare alcuni elementi per modificare la tipologia del testo o per scegliere il carattere. Il carattere standard di visualizzazione della pagina web è di tipo **con grazie** a **spaziatura variabile** (come il noto Times o come quello con cui è scritto questo testo. I caratteri senza grazie sono quelli privi degli abbellimenti ortogonali ai margini detti appunto grazie, come ad esempio il carattere Arial).

Alcuni elementi che modificano il tipo di carattere sono i tag `<tt></tt>`, `<code></code>`, `<kbd></kbd>` o `<samp></samp>` essi fanno sì che il testo al loro interno sia rappresentato con **spaziatura fissa**, cioè in cui tutte le lettere occupano lo stesso spazio in larghezza come nelle macchine da scrivere o nei vecchi display (infatti si usano in genere per rappresentare i codici in linguaggi di programmazione). Un carattere di questo tipo nei word processor moderni è il Courier. Se si vuole inserire del testo con spaziatura fissa e che inoltre non trascuri spazi ed a capo come normalmente avviene nel codice HTML, si può usare il tag `<pre> </pre>` che sta per "testo preformattato", di cui cioè HTML non codifica la formattazione.

Altri effetti di modifica del carattere sono creati da `<sup> testo </sup>` che trasforma il testo contenuto in apice (superscript) e `<sub> testo </sub>` , che trasforma il testo contenuto in pedice (subscript).

Più di recente era stato inserito anche l'elemento `<font>` i cui attributi permettono di scegliere dettagliatamente tipo, dimensione e colore dei caratteri. La sintassi dell'elemento è, per esempio:

```
<font face="tipo di carattere"
     color="colore" size="dimensione"> testo </font>.
```

Tale elemento è però deprecato dal W3C e se ne sconsiglia l'uso (gli elementi deprecati dovrebbero essere eliminati poi definitivamente dal linguaggio). Come vedremo il modo più corretto e potente di modificare la formattazione degli elementi di (X)HTML è quello utilizzare le regole di stile ed i fogli di stile a cascata, che consentono piena libertà di scegliere differenti caratteri per le differenti parti del documento.

4.6 Attributi e valori

Abbiamo visto quindi che per definire le caratteristiche di un elemento utilizziamo delle coppie attributo-valore ed abbiamo visto vari esempi. Gli attributi e i possibili valori da essi assunti sono specifici del linguaggio e definiti esattamente dalle specifiche W3C (accessibili tramite il sito *www.w3c.org*).

A seconda del tipo di elemento gli attributi possono assumere valori espressi in differenti modi: **numerici, percentuali, testuali**. Gli attributi che indicano dimensione, come width e height, possono essere espressi da un numero che indica il valore in pixel o da una percentuale dell'elemento contenitore. Un attributo come src nell'elemento img deve avere come valore la stringa che rappresenta un URL valido di un'immagine (l'indirizzo esatto cioè di dove si trova l'immagine collegata sul web). Molti attributi infine possono assumere solo in insieme finito di stringhe (es. “left”, “center”, “right” per align) rigidamente determinata dal DTD del linguaggio. Ricordiamo che una delle innovazioni di XHTML rispetto a HTML consiste nell'obbligo di valorizzare tutti gli attributi, cioè non possono più esistere attributi senza valori associati. Siccome in realtà alcuni attributi non erano stati pensati per assumere differenti valori, in alcuni casi esiste un unico valore ammissibile (es. `noshade="noshade"`).

```
<html><head>
<title> Fondamenti </title>
</head>
<body>
<h1> Fondamenti di informatica </h1>
<p> In <b>questo corso</b> studieremo
molto per passare <i>l'esame</i></p>
<h2> Che altro dire? </h2>
Che però è meglio non <big>esagerare
<big>troppo</big></big>
Meglio <small>poco</small> ma bene,
purché non
<small><small>pochissimo</small>
</small>
<body></html>
```

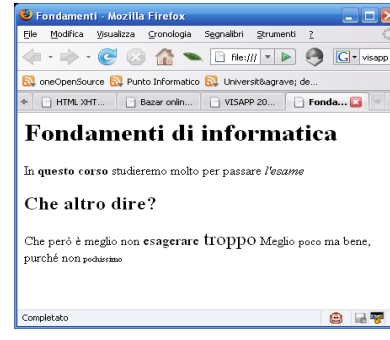


Figura 32: Esempio di codice (X)HTML con tag di formattazione

4.7 Collegamenti ipertestuali

Veniamo ora alla caratteristica peculiare dell'(X)HTML: la possibilità di creare ipertesti, cioè inserire nelle pagine le cosiddette **àncore**, parti cliccando sulle quali si passa alla visualizzazione di un altro documento sul browser.

Quest'altro documento è tipicamente altra pagina web, o una diversa posizione sulla medesima pagina, ma non necessariamente. Dato che il browser è in realtà in grado di visualizzare vari tipi di file o di fare avviare un'altra applicazione per il PC attraverso cui aprire un tipo di file non direttamente gestito dal browser stesso (applicazioni helper), il file collegato può essere di vari tipi (es. immagini, video, pdf, script).

L'elemento che crea il collegamento è **<a>** (dal termine àncora, che si riferisce appunto al punto di collegamento tra i documenti). Per inserire un collegamento, si usa l'attributo href, il cui valore è l'URL della pagina web collegata, es. `<a href=" http://www.univr.it/">`, se il riferimento è assoluto. Se si collega un file che risiede nella stessa cartella pubblica della pagina di provenienza si può usare quello che viene chiamato collegamento relativo, omettendo cioè l'indirizzo del sito, es: `<a href="altrapagina.html">`. Quando si creano documenti (X)HTML occorre sempre fare attenzione alla correttezza di tutti i riferimenti, assicurandosi, se il riferimento è relativo, che esistano i file corrispondenti sul filesystem del server, o, se è assoluto, che l'URL sia corretto. Se non si specifica nulla, dopo il clic sull'àncora il nuovo documento viene visualizzato sulla stessa finestra del browser dove si trovava il documento di origine. E' però possibile fare in modo che appaia in una differente finestra con l'attributo **target**. E' possibile anche indicare una cartella base in cui trovare i file collegati mediante l'elemento `<base>`. Tutta la parte di documento contenuta tra tale apertura di elemento e la successiva chiusura `</a>` agiranno come collegamento ipertestuale: il testo sarà evidenziato e diventerà cliccabile e lo stesso accadrà con le immagini (vedi Figura 33). L'evidenziamento del testo avviene in genere mediante colorazione blu e sottolineatura. Quello delle immagini avviene attraverso la colorazione del bordo, per cui se il bordo viene eliminato indicando `border="0"`, il link non sarà evidenziato, a meno di non passare sull'immagine stessa col cursore. Infatti un altro modo di evidenziare le ancore implementato sui browser consiste nel modificare l'aspetto del puntatore (ad esempio trasformandolo in una piccola

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1
transitional.dtd">
<html>
<head>
<title>La mia pagina</title>
</head>
<body>
<a href="http://www.regione.sardegna.it">
<img align="right" src="mare.jpg" width="100"
alt="mare"/> vado al mare </a>
<a href="home.html"> o resto a casa? </a>
</body>
</html>
```

Figura 33: Inserimento di collegamenti ipertestuali. Con il link relativo dell'esempio, occorre che il file "home.html" si trovi nella medesima cartella del file di origine.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1
transitional.dtd">
<html>
<head>
<title>Ricette</title>
</head>
<body>
<p>Ottimi primi sono la
<a href="zuppa_fagioli.html"
target="primi"> zuppa di
fagioli</a> oppure gli
<a href="carbonara.html"
target="primi"> spaghetti alla
carbonara</a>, ....</p>
<p>Tra i secondi segnaliamo la
<a href="cotoletta.html"
target="secondi"><em>Cotoletta
alla Viennese</em></a>, o la <a
href="coda.html"
target="secondi"><em>coda alla
vaccinara</em></a>....</p>
</body> </html>
```

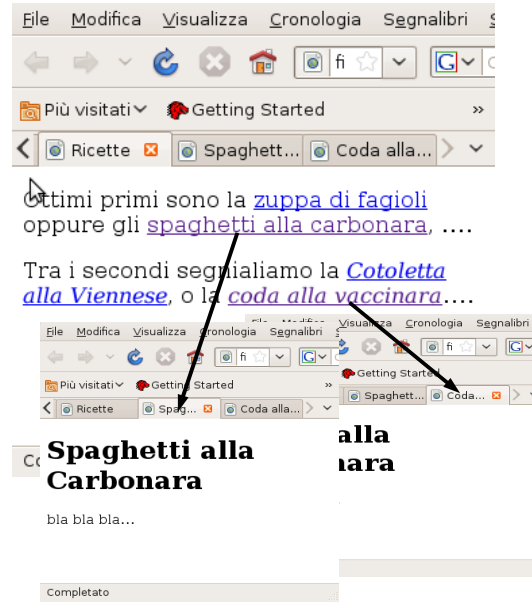


Figura 34:Uso dell'attributo target per aprire i collegamenti in differenti finestre o schede del browser. A sinistra parte del codice della pagina, a destra in alto il risultato.

mano). Si noti poi che il colore dei testi àncora e l'aspetto dei puntatori possono essere modificati dai creatori dei contenuti web. Tra le opzioni dei browser c'è comunque in genere quella di forzare l'aspetto di tali elementi. Per collegare una differente sezione della stessa pagina, si può inserire un elemento a vuoto con attributo **name**, che agirà da riferimento, es:

```
<a name="qui"></a>
```

Se nella stessa pagina o in un'altra si trova un elemento del tipo:

```
<a href="pagina.html#qui"> </a>
```

il browser visualizzerà la pagina indicata a partire dal punto indicato. Se l'url nel riferimento non è una pagina html, il browser agirà in maniera differente a seconda delle opzioni scelte dall'utente (in genere selezionabili con il menu "opzioni" del browser. Ad esempio, se l'URL si riferisce ad un indirizzo email, come:

```
<a href="mailto:andrea@univr.it"> scrivimi</a>
```

in genere verrà aperta un'applicazione esterna per scrivere un messaggio di posta elettronica, come si vede in Figura 35, se si collega un file di tipo diverso, se il browser è in grado di gestirlo tramite un plugin (cioè un programma che integra la visualizzazione del file all'interno dell'interfaccia del browser) o un'applicazione esterna, verrà aperto, altrimenti verrà richiesta all'utente l'autorizzazione a salvare il file stesso sul filesystem locale. Un ulteriore attributo che si può inserire sull'elemento ancora è accesskey, che crea la possibilità di accedere al contenuto del collegamento anche con una combinazione di tasti (alt+control insieme alla cifra indicata), es:

```
<a href="mailto:andrea@univr.it">
Scrivimi!<img src="mail_icon.png"></a>
```

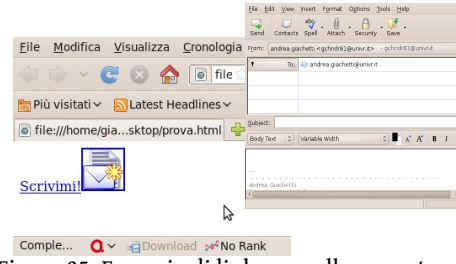


Figura 35: Esempio di link con collegamento a e-mail: in genere fa aprire il programma standard di posta elettronica con l'indirizzo precompilato.

```
<a href="index.html#pippo" accesskey="p">Pippo</a>(Alt-p,Ctrl-p)
```

Infine, per consentire la navigazione sequenziale tra i collegamenti, si può inserire l'ordine di scorrimento tra i collegamenti ottenibile premendo ripetutamente il tasto tab mediante l'attributo `tabindex`.

Una forma di link ipertestuale più complicato realizzabile attraverso le immagini è quello relativo alle cosiddette "mappe" cliccabili, cioè immagini su cui posizionando il cursore in posizioni differenti e premendo il tasto sinistro del mouse si attivano differenti link. Esistono diverse tecniche per realizzare questo effetto, alcune anche server side, cioè che trasmettono le coordinate del punto cliccato al server, sul quale sarà memorizzata la corrispondenza punto-destinazione del link. Esiste, però, anche un elemento apposito HTML per realizzarle client side ed è `<map>`. Attraverso di esso è possibile definire regioni tramite poligoni ed associare a ciascuno un evento differente.

4.8 Elenchi e tabelle

Una struttura standard dell'HTML è quella degli **elenchi**. Il loro utilizzo è piuttosto rilevante, dato che è sicuramente una buona norma presentare le informazioni rilevanti di una pagina utilizzando pochi punti evidenziati. Prove sperimentali mostrano infatti che i lettori delle pagine web tendono a non leggere con attenzione tutto il testo, ma soltanto gli inizi delle frasi evidenziate.

In (X)HTML si possono utilizzare elenchi di due tipi: non ordinati e ordinati. Quelli non ordinati sono i cosiddetti elenchi puntati. Si realizzano aprendo e chiudendo l'elemento `<ul>` e associando ad ogni punto un elemento `<li></li>`, come nell'esempio che segue:

```
<ul><li>Introduzione</li>
<li>...</li>
 ...
</ul>
```

```
<h1> Cose da fare </h1>
<ul>
<li> Dormire </li>
<li> Mangiare </li>
<ul>
<li> Pranzo </li>
<li> Cena </li>

<ol>
<li> Minestra</li>
<li> Frutta </li>
</ol>
</ul>
</ul>
```

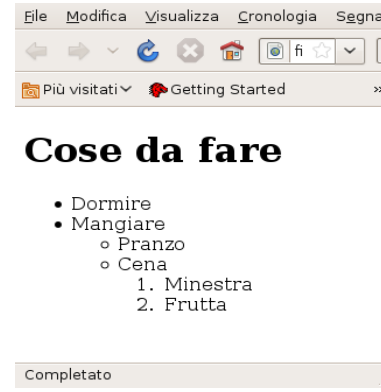


Figura 36: Esempio di pagina con elenchi puntati e numerati

Per avere un elenco numerato, basta utilizzare `<ol>` (ordered list) invece di `<ul>`. Sul browser gli elenchi appaiono visualizzati come in Figura 36. Gli elenchi, come si vede, possono essere annidati creando livelli gerarchici, con i livelli successivi al primo che verranno visualizzati spostati di una tabulatura a destra e diversi punti.
La numerazione degli elenchi ordinati può essere variata con gli attributi start (che stabilisce il numero di partenza) e value (su attribuisce un numero particolare).

Altra struttura particolare in HTML sono le **tabelle**. Esse sono importanti in quanto consentono di inserire dati di tipo tabulare in modo semplice. Agli albori del web in realtà, le tabelle venivano anche usate per creare layout di pagina, cioè disporre testo e figure nelle varie caselle per dare una struttura spaziale al documento stesso. Oggi si sconsiglia di creare il layout in questo modo, dato che attraverso le regole di stile è possibile posizionare arbitrariamente le varie parti della pagina ove si vuole (questo non era possibile con gli attributi e valori del "vecchio" HTML).

La sintassi dell'elemento **table** è abbastanza semplice: per creare una tabella si apre e chiude un tag `<table>`. All'interno di esso l'elemento `<tr>...</tr>` serve per inserire una nuova riga, mentre elementi `<td>...</td>` inseriti all'interno di `<tr>`. Per intestazioni di colonne al posto di `<td>` si può usare `<th>`. Le tabelle consentono ampie possibilità di variazione di aspetto, anche per questo sono state e sono ancora molto usate per gestire il layout di pagine. Gli attributi possono essere inseriti a vari livelli: sull'elemento table si possono inserire cellspacing, che consente di inserire la distanza (in pixel) tra una cella e l'altra, oppure tra una cella e il bordo (default 1) e cellpadding, che permette di variare la distanza tra il contenuto della cella e il bordo, inserendo un valore in pixel o anche in percentuale.

Alcuni attributi possono invece essere inseriti sia su table, che sulle sottoparti td e tr (cioè le righe e le semplici caselle). Essi sono, ad esempio, il valore di border (lo spessore del bordo in pixel), width, la larghezza percentuale o assoluta (pixel) dell'elemento, align, che può assumere i valori "left", "center" e "right" per definire la giustificazione del testo inserito nelle caselle, bgcolor, che consente di indicare il colore dello sfondo, bordercolor che permette di definire il colore del bordo o anche background, che permette di inserire un'immagine di sfondo mettendone l'url come valore assegnato all'attributo stesso. Qualche esempio di effetto dell'inserimento di attributi con valori specifici è visualizzato in Figura 37.

```
<table width="75%" border="1" align="center"
bgcolor="#00FF00">
 <tr>
  <td width="50%" bgcolor="#FF0000">
   <font color="#FFFFFF">prova</font>
  </td>
  <td width="50%">
  </td>
 </tr>
</table>
```

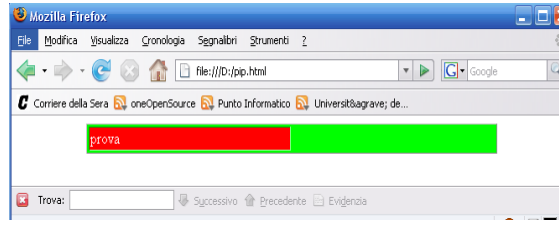
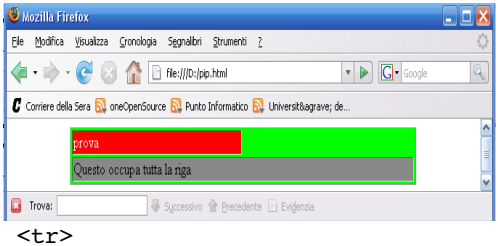
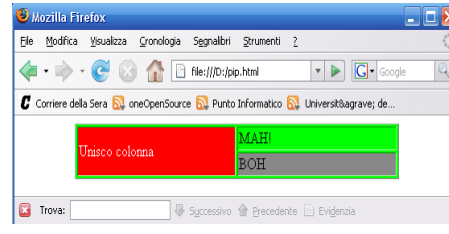


Figura 37: Esempio di tabella con uso di semplici attributi

Per creare strutture un po' più complicate è possibile anche unire celle tra loro. Questo lo si può fare attraverso due attributi, rowspan e colspan, cui si assegna il numero di righe o colonne da occupare da parte della casella. Questo effetto è particolarmente interessante perché non è ottenibile in maniera alternativa con le regole di stile. Esempi di unione di colonne e righe sono riportati in Figura 38.




```
<tr>
<td width="50%" bgcolor="#FF0000">
 <font color="#FFFFFF">
  prova</font>
</td>
<td width="50%">
</td>
</tr>
<tr>
<td colspan="2" bgcolor="888888">
Questo occupa tutta la riga
</td></tr>
```




```
<tr >
<td rowspan="2" width="50%"
  bgcolor="#FF0000">
<font color="#FFFFFF">Unisco
  colonna</font>
</td>
<td width="50%"> MAH!  </td>
</tr>
<tr><td bgcolor="888888"> BOH
</td>
</tr>
```

Figura 38: Unione di caselle: attraverso l'attributo colspan viene creata una casella che occupa due colonne (sinistra), mentre con rowspan si uniscono due righe (a destra)

4.9 Formattazione e regole di stile

Abbiamo visto che, inizialmente, in HTML si creava la cosiddetta formattazione del documento (aspetto grafico, scelta colori e caratteri, disposizione spaziale delle parti o layout, ecc.) soltanto

mediante attributi e valori. Questo non è un modo ideale in quanto codifica simultaneamente contenuto logico ed aspetto grafico. Inoltre le possibilità di modifica dell'aspetto grafico erano limitate: la visualizzazione degli elementi era perlopiù standard e fissata dal browser.

Da HTML 4.01 si è cercato di separare il contenuto dal layout e di ampliare le possibilità di variazioni stilistiche. Il risultato sono state le cosiddette regole di stile, che vengono in genere codificate in documenti separati dalla pagina stessa detti fogli di stile o **fogli di stile a cascata** (Cascading Style Sheets o CSS) .

La prima versione di questa codifica risale al 1996, la versione attuale 2.1 è del 2004 e si sta lavorando alla versione 3.0.

Questa codifica separata fornisce numerosi vantaggi: innanzitutto consente di controllare in maniera più semplice l'aspetto grafico della pagina, generando codice più compatto. In un sito si possono inserire molte pagine che fanno riferimento ad un solo foglio di stile e questo stile può essere variato senza dover toccare il codice delle pagine. Si possono anche utilizzare fogli di stile differenti per supportare differenti strumenti di visualizzazione che necessitano di adattamenti di blocchi e caratteri alle diverse dimensioni (monitor, stampanti, palmari, ecc.). In tale modo si può anche supportare l'accessibilità (creando fogli appositi per lettori braille, ecc.), sempre senza riscrivere i documenti.

Vediamo quindi la codifica dei CSS in maggior dettaglio. La sintassi delle regole di stile è separata da quella dell'HTML/XHTML e viene collegata al documento HTML/XHTML in uno dei seguenti modi:

- Mediante inserimento “inline” all'interno della pagina, inserendo l'apposito elemento `<style>` dentro `<head>`, come nell'esempio sotto:

```
<html>
<head>
<title>Inserire i fogli di stile in un documento</title>
<style type="text/css">
body { background: #FFFFCC; }
</style>
</head> <body> …
```

- Utilizzando uno o più documenti di stile associati (fogli di stile) contenenti le regole e collegati alla pagina in uno dei due seguenti modi:

```
<html>
<head>
<title>Inserire i fogli di stile in un documento</title>
<style type="text/css">
<style>
@import url(stile.css);
</style>
```

oppure:

```
<html>
<head>
<title>Inserire i fogli di stile in un documento</title>
<link rel="stylesheet" type="text/css" href="stile.css">
</head><body>...
```

Questi ultimi metodi sono vantaggiosi in quanto realizzano propriamente uno dei principali vantaggi dell'uso dei fogli di stile, cioè la separazione tra documento che contiene struttura logica e contenuto testuale e documento che contiene lo stile di rappresentazione. Questo fa sì che si possa modificare lo stile di tutte le pagine che importano lo stesso file, modificando solo quest'ultimo. L'altra possibilità cui avevamo accennato, cioè quella di inserire differenti stili per differenti mezzi di visualizzazione può essere infine realizzata con l'attributo "media", inserito in `<style>` o `<link>`, es:

```
<link rel="stylesheet" type="text/css" media="screen"
href="screen.css" />
<link rel="stylesheet" type="text/css" media="print, tv,
aural" href="print.css" />
```

In questo caso ho due fogli di stile e se il documento HTML è visualizzato sullo schermo, userò il primo, se è stampato, il secondo. I possibili valori per l'attributo "media" sono: "screen" (desktop e laptop), "handheld" (PDA e smartphone), "print" (stampanti), "braille" (browser braille), "embossed" (stampanti braille), "projection" (proiezioni), "speech" o "aural" (sintetizzatori vocali), "tty" (telescriventi), "tv" (televisioni), "all" (tutti i dispositivi). Il supporto all'interpretazione dell'attributo media, però, non è ancora largamente diffuso.

E' possibile definire fogli di stile alternativi anche senza che questi siano selezionati automaticamente in base al tipo di terminale. Basta infatti definire per ciascun file di stile importato un'etichetta con l'attributo "title". Molti browser di ultima generazione (ad esempio Mozilla Firefox) consentono in questo caso di scegliere il foglio di stile da usare tra i diversi definiti dall'autore visualizzando le etichette in un menu (menù visualizza/stile pagina).

Si parla di fogli di stile a cascata in quanto, essendo possibile definire regole di stile in vari modi e posizioni, in decodifica il browser scorre la lista delle regole per ciascun elemento della pagina ed a parità di specificità della validità della regola, considera valida l'ultima. Si noti che le regole di stile possono essere inserite anche localmente a livello del singolo elemento, inserendo nell'elemento stesso l'attributo `style="..."` seguito dalla definizione della regola che si applicherà quindi al solo elemento che la contiene. Se non si utilizzano i fogli di stile quest'ultima modalità è appunto quella che sostituisce la formattazione ottenuta con gli attributi vecchio stile di HTML, in genere dichiarati "deprecated" nelle ultime versioni del linguaggio e quindi da evitare.

Cerchiamo ora di chiarire cosa siano le regole di stile che troveremo, ad esempio, nei file style.css degli esempi visti prima. La sintassi è molto semplice, ed è riportata in Figura 39: prima di tutto si inserisce un selettore, che determina a quali parte del documento si applica la regola, e poi un numero variabile di coppie proprietà-valore, che attribuiscono le proprietà di

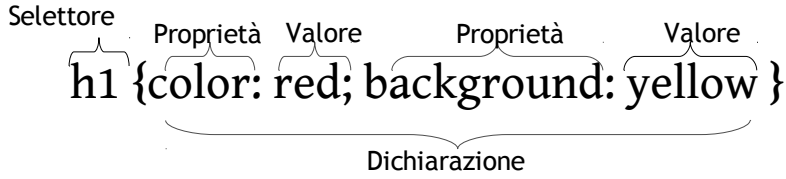


Figura 39: Esempio di regola di stile

formattazione a tali parti.

Le proprietà che si possono stabilire con queste regole sono moltissime, abbiamo detto che con i CSS sono infatti state anche date molte più libertà di variare la rappresentazione grafica del documento rispetto all'HTML semplice. Non è interesse di questo testo vederle tutte, rimandiamo a un manuale o alle specifiche W3C per il dettaglio, ricordiamo come esempio le più comunemente utilizzate: background (colore sfondo), border (bordo), color (colore testo), margin, padding, text-align, le proprietà relative ai caratteri (font-family, font-size, font-weight), ecc. Ogni attributo, come per gli attributi dei tag HTML, ha definito il tipo di valori che può assumere (numeri, percentuali o stringhe di caratteri predefinite).

I **selettori** possono selezionare una semplice classe di elementi tra quelli che abbiamo visto. E' possibile selezionare le varie parti dei documenti con grande libertà. Vediamo qualche esempio.

In primo luogo si possono selezionare **tipi di elementi:** ad esempio `h1{color:red;}` farà sì che tutti gli elementi h1 avranno testo di colore rosso. Se si vogliono assegnare le stesse regole a più tipi di elemento con un unica regola, si possono raggruppare i nomi in un unico selettore mediante la virgola. Ad esempio `h1, h2, h3 {color:red;}` farà sì che tutti gli elementi di tipo h1, h2 e h3 abbiano colore di testo rosso. Un altro tipo di selettore può essere quello riferito a una pseudoclasse di elementi, cioè classi di elementi che hanno una determinata proprietà: se, ad esempio, consideriamo l'elemento `<a>` per le àncore dei collegamenti ipertestuali, possiamo distinguere tra collegamenti visitati, non visitati, àncore su cui si trova il cursore, attivate, ecc. Nel linguaggio dei CSS si possono specificare questi stati con il simbolo : e la proprietà. Quindi, se nel mio foglio di stile scrivo:

```
a:link {color:red}, a:visited {color:yellow},
a:hover {color:green} ,a:active {color:blue}
```

renderò rispettivamente rossi i colori dei link non visitati, gialli i colori dei link già visitati, verdi i colori dei link su cui passa il cursore, blu i colori dei link attivati.

Per applicare regole a gruppi di elementi di un certo tipo, si può usare la selezione di **classi** di elementi. Basta assegnare la classe agli elementi con l'attributo class=”nomeclasse” e selezionare poi la classe “nomeclasse” con l'apposito selettore, es:

```
.nomeclasse {background=yellow; color=red;}
```

assegnerà colore rosso e sfondo bianco a tutti gli elementi in cui sarà indicato l'attributo class con valore nome, es:

```
<h1 class="nomeclasse"> Il linguaggio XHTML</h1>
```

Con questa sintassi le classi sono generiche, cioè applicabili a tutti i tipi di elemento, comunque le regole possono essere create anche per un tipo particolare mettendo prima del punto il tipo, es:

```
h1.nomeclasse {background=yellow; color=red;}
```

In tal caso, se provassimo a inserire il nome "nome" a un elemento di tipo differente, per es:

```
<p class="nomeclasse"> non funziona </p>
```

la regola non sarebbe applicata.

Simile a quello delle classi di elementi è l'uso del selettore di identificativo, che si ottiene appunto inserendo un identificativo univoco a un elemento con l'attributo `id="nomeid"`, es se nel file .css:

```
#frasecelebre {background=yellow; color=red;}
```

sarebbe interessato l'elemento così marcato, che sarà unico:

```
<p id="frasecelebre"> Chi tardi arriva
male alloggia </p>
```

Anche qui è possibile specificare nella regola il tipo di elemento in cui deve essere inserito l'id, ad esempio:

```
p#frasecelebre {background=yellow; color=red;}
```

farà in modo che la regola valga solo per gli elementi p con id uguale a "frasecelebre" .

La differenza rispetto a class è che l'id può essere applicato ad un solo elemento in tutta la pagina HTML e, quindi, non va bene se si vuole applicare la stessa regola a diversi elementi. Per questo vengono in genere utilizzati per riferirsi a sezioni ben definite di un modello di pagina: ad esempio, nei modelli classici dei Content Management Systems abbiamo sezioni caratterizzate da id come `div#page`, che indica il box che contiene l' intera pagina html, `div#header` che indica il box che contiene la testata, `div#navbar` che identifica la barra di navigazione e così via.

Esistono poi svariate altre modalità di selezione. Ad esempio si possono concatenare i selettori: se scriviamo:

```
selettore1 selettore2 { regola },
```

la regola si applicherà agli elementi selezionati da selettore2 solo se contenuti in un elemento di tipo selettore1.

Ad esempio:

```
p em {background=yellow; color=red;}
```

applicherà la regola a tutte le parti di testo enfatizzato contenute in un elemento paragrafo. Rimandiamo anche qui a un manuale, anche online, per le specifiche di tutti i tipi di selettori. Quello che qui interessa è semplicemente rimarcare il fatto che, attraverso i fogli di stile, è davvero possibile selezionare in modo completamente libero le parti dei documenti in base a nome, categoria o ruolo ed assegnare ad esse le caratteristiche di aspetto volute.

4.10 Formattazione del documento: il testo

Una delle caratteristiche controllabili per quanto riguarda lo stile dei nostri elementi (X)HTML è senza dubbio il testo. In origine HTML non dava la possibilità di modificare le caratteristiche del testo; abbiamo visto che in seguito, però, erano stati inseriti alcuni semplici elementi per la modifica: `<em></em>` per enfatizzare, `<b></b>` per il grassetto `<big></big>` `<small></small>` per modificare le dimensioni. Oppure, l'elemento `<pre> </pre>`, che fa in modo che il testo contenuto all'interno venga rappresentato così come scritto sull'editor, con anche spazi e a capo di solito ignorati ed usando un carattere a spaziatura fissa.

Come abbiamo detto, però, le regole di stile non soltanto sono ufficialmente il metodo da adottare a tale fine secondo le direttive del W3C, ma sono anche molto più potenti e complete, come evidenziano gli esempi che seguono.

Il tipo di carattere (font) si può scegliere con la dichiarazione: `{font-family:"nomefamiglia"}`, es:

```
{font-family:"Arial Black"}.
```

Dato che sui browser usati dagli utenti non tutti i tipi di font possono essere disponibili, si possono indicare più alternative, come in:

```
{font-family:"Arial Black", "Helvetica Bold", sans serif}.
```

In tal caso il primo tipo di font utile trovato verrà visualizzato.

Si noti che al posto della famiglia precisa si possono usare le espressioni Serif (con grazie), sans serif (senza grazie), monospaced (spazi uguali).

L'attributo font-style permette di controllare lo stile del testo, ad esempio `font-style:italic` crea il corsivo; `font-weight` invece controlla lo spessore, `font-weight:bold` crea il grassetto, ma possiamo anche usare `font-weight:lighter` `font-weight:bolder` per modificare in modo incrementale o addirittura usare un valore numerico.

Le dimensioni si controllano con `font-size:valore;` i valori ammessi sono xx-small, x-small, small, medium, large, x-large, xx-large o valori numerici, con varie unità di misura supportate, es. `font-size:12pt` (punti tipografici = 1/72 pollice) 15px (pixel), ma si possono anche indicare le unità `cm` (centimetri), `in` (pollici) o valori percentuali del carattere standard, es. `font-size:90%.` Utilizzando `line-height:valore;` si controlla l'interlinea del testo inserendo un valore percentuale (%), in proporzione (em) o assoluto (a). E' possibile anche controllare lo spazio tra i caratteri (word-spacing) o l'indentazione (spazio dal bordo) dell'inizio testo (text-indent). L'allineamento del testo all'interno di un blocco (quindi

solo per gli elementi block-level) si controlla con `text-align` che può assumere valori `left, right, justify.`

Come abbiamo già visto, il colore del testo si controlla con `color:nomecolore;` quello dello sfondo con `background:valore;` qui il valore oltre che di colore può anche essere `transparent,` cioè si può rendere lo sfondo trasparente. Un'ulteriore possibilità è quella di mettere come sfondo dell'elemento un'immagine, in tal caso si indica: `background:url(indirizzoimmagine)` con eventuale indicazioni su come l'immagine vada scalata o ripetuta `( repeat, no-repeat, repeat-x, repeat-y, fixed, scroll )`

✔ *Nota: valori utilizzabili per i colori*

Abbiamo visto molti esempi, ma non abbiamo descritto in dettaglio come effettivamente si controllano i valori dei colori degli elementi, per cui rimediamo.

Esistono in HTML/CSS 16 nomi di colori predefiniti che consentono l'uso della semplice sintassi `color="nome"`. Questi sono: `aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow.`

Per controllare invece le sfumature della tinta in dettaglio, possiamo utilizzare due differenti notazioni: la prima è quella esadecimale, che si può ottenere scrivendo, ad esempio, `color="#RRGGBB"` negli elementi HTML e `color:#RRGGBB` nelle regole CSS, ove RR, GG, BB sono, appunto, numeri esadecimali di due cifre che codificano numeri da 0 a 255 per le componenti rispettivamente di rosso, verde e blu. Pur essendo poco intuitivo (le cifre esadecimali vanno da 0 a F) il metodo è il più usato; ad esempio `color=#00ff00` vuol dire verde saturo, `color=#400040` è un violetto poco saturo (rosso+blu) eccetera. Nelle regole di stile si può comunque usare la notazione RGB anche in decimale, utilizzando `color=rgb(r,g,b),` con valori 0-255 o percentuali, come in `color:rgb(200,0,255)` o `color:rgb(10%,50%,100%).`

✔ *Nota: la leggibilità del testo sul web*

Al di là del come si possa tecnicamente variare l'aspetto del testo delle pagine, ha senso chiedersi se esistano delle regole sensate per scegliere nei nostri documenti tipologia, colori o dimensioni dei caratteri in modo da renderli più efficaci e gradevoli, cioè qual è la scelta che migliora l'**usabilità** del sito. Uno dei criteri per rendere più efficiente l'utilizzo dei siti web è certamente quello di rendere più leggibili le informazioni utili. A questo scopo si sono fatti vari studi sperimentali misurando, ad esempio, la velocità di lettura (cronometrando l'esecuzione di opportuni task). Si è notato che tale velocità dipende da tipo e dimensione dei caratteri. I tipi di caratteri si dividono in due principali famiglie: con grazie (es Times) e senza grazie, cioè abbellimenti ai bordi (es. Arial). Esistono poi i vari stili (*corsivo*, **grassetto**). Si è visto che i font senza grazie sono più leggibili sullo schermo (a differenza che sulla carta) e che sullo schermo il corsivo è meno leggibile (per via dell'aliasing, cioè degli artefatti che si creano a causa della dimensione finita dei punti dello schermo o pixel). La dimensione più grande migliora la leggibilità e valgono ovviamente le considerazioni viste prima per i colori (l'azzurro è meno leggibile, i contrasti di luminosità vanno preferiti a quelli di tinta). Il minuscolo è più

leggibile del maiuscolo (la ragione sta nel fatto che in realtà non leggiamo lettera per lettera le parole, ma di solito le riconosciamo dalla struttura grafica complessiva, più riconoscibile nel minuscolo per via delle diverse altezze dei caratteri. Altra cosa da notare: sul web non leggiamo tipicamente tutti i contenuti, ma esistono delle tendenze consolidate a cercare le informazioni interessanti in determinate regioni. Jacob Nielsen (www.useit.com) ha studiato il fenomeno, trovando con esperimenti di eye tracking, che lo sguardo degli utenti sull'interfaccia del browser web si concentra, di solito, in un pattern ad F nella parte in alto a sinistra (Figura 40). E' buona norma, quindi, inserire nelle pagine in tali posizioni ed evidenziare, ad esempio, con elenchi puntati, le informazioni più importanti. Si può osservare che la maggior parte dei siti web è strutturato proprio in questo modo.

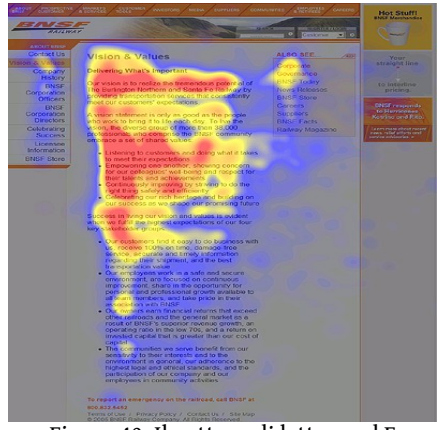


Figura 40: Il pattern di lettura ad F verificato negli esperimenti di Eye Tracking (da www.useit.com) . Le zone evidenziate sono quelle in cui si sofferma di più lo sguardo degli utenti.

4.11 Gestione del layout con le regole di stile

L'altra cosa importante che le regole di stile permettono facilmente di realizzare è la gestione dell'aspetto grafico (spazi, organizzazione, caselle, ecc.) che si indica generalmente con il termine **layout**.

La gestione degli spazi è importantissima per rendere efficace il contenuto ed evidenziare le parti importanti. Nell'(X)HTML standard le possibilità di gestione del layout erano limitate, non si potevano posizionare le parti a piacimento, tanto che per creare struttura, colonne, spazi per menu, eccetera si dovevano usare spesso le tabelle, seppure impropriamente.

L'uso dei CSS consente, invece, un'organizzazione completa e flessibile, separando, tra l'altro l'impaginazione dal contenuto e dando così la possibilità di cambiare il layout senza toccare il contenuto. Esso consente anche di creare i cosiddetti layout “liquidi”, cioè che cambiano quando si ridimensiona la finestra del browser.

La gestione dell'impaginazione è semplice: il documento (X)HTML, infatti, è strutturato in sezioni logiche cui si possono assegnare vari tipi di attributi. Se si vuole raggruppare una parte di documento e rendere la stessa posizionabile ovunque sull'interfaccia, basta includerla all'interno di un elemento block generico come `<div></div>`.

Si crea così un modello di documento a “box”, dove ogni sezione (div) è, appunto, come una scatola che include contenuti, e di cui si possono controllare posizionamento, spazio circostante, bordi esterni, margini e così via.

Le “scatole” si possono posizionare all'interno della pagina in due modi: nel flusso del documento (vengono, cioè, rappresentate dal browser dall'alto in basso nell'ordine in cui sono presentate, come accade col semplice HTML) o posizionandole liberamente in coordinate esatte rispetto al genitore (posizionamento assoluto), alla finestra del browser (posizionamento fisso) o relativamente alla posizione determinata dal flusso (posizionamento relativo). Le scatole si possono anche sovrapporre, e la visibilità sarà gestibile con un attributo con cui specificare la profondità (z-index).

A queste “scatole”, quindi si possono attribuire posizione e dimensioni a piacere. Ad esempio possiamo determinare se l'elemento è visibile o meno e di che tipo è con la proprietà “display” (che può assumere i valori: none, block, inline).

Possiamo posizionare poi i blocchi in maniera arbitraria, utilizzando la proprietà `“position:tipo”`, seguita dallo spostamento nelle due direzioni indicato dalle proprietà `top:valore, left:valore, bottom:valore` o `right:valore` che indicherà una distanza (valore) espressa usualmente in pixel o percentuale dall'alto, sinistra, basso o destra rispetto alla posizione di riferimento. Questa posizione di riferimento dipenderà dal tipo di posizionamento richiesto: `position:absolute;` sistemerà il blocco alla distanza indicata dal punto in alto a sinistra dell'interfaccia, `position:relative;` sistemerà il blocco alla distanza indicata dal punto in cui sarebbe stato visualizzato se non ci fosse stata indicazione di posizionamento, cioè alla posizione corrente del flusso di decodifica del codice HTML.

Anche le dimensioni (proprietà `width` ed `height`) possono essere scelte a piacimento, indicate o in valore assoluto (con le differenti unità di misura utilizzabili) o in percentuale, l'uso di valori percentuali per posizione e dimensione consente di creare i cosiddetti layout “liquidi” in cui tutte le componenti vengono scalate in proporzione quando si ridimensiona la finestra.

4.12 Regole di stile, ereditarietà, cascata

Abbiamo visto che il documento (X)HTML definisce una struttura gerarchica, dove gli elementi che includono altri elementi sono in rapporto padre-figlio con questi (es. un paragrafo che include un testo in grassetto, una sezione `<div>` che include titoli, eccetera). Come si comportano le regole di stile in relazione a queste “discendenze”?. Dipende dalle proprietà. Alcune di esse sono ereditate dal genitore, altre no. Di massima non sono ereditarie quelle che gestiscono la formattazione del “box” dell'elemento (bordi, sfondo, ecc.). Mentre lo sono, ad esempio, quelle relative al testo. Se il colore del testo è definito a livello di `<body>` di colore rosso con la regola `body {color:red;}`, allora anche le sottoparti, se non sono inserite altre regole più specifiche per esse, avranno il testo di colore rosso.
Per attribuire il corretto stile a ciascun elemento di un documento complesso, il browser deve svolgere un processo non banale. L'interprete del markup deve trovare i valori per le proprietà di stile (es. famiglia, dimensioni e stile dei caratteri, colore del testo, ecc.) di ogni elemento. Per fare questo viene utilizzato il cosiddetto meccanismo della **cascata** (di qui il nome “fogli di stile a cascata”). Esso ordina in base alla loro specificità le diverse regole attribuite a ciascun elemento (tale specificità dipende, ad esempio, dall'ordine di caricamento dei fogli di stile che contengono le regole), e determina quale applicare. Se nessuna regola specifica è disponibile, viene verificata l'ereditarietà. Per le proprietà non ereditate si usano i valori predefiniti. Anche qui rimandiamo ai numerosi tutorial online o manuali per approfondimenti pratici e dettagli (vedi sitografia).

4.13 Intermezzo pratico: creare un documento HTML

Visto che le informazioni contenute nei precedenti capitoli dovrebbero mettere teoricamente in grado il lettore di realizzare le proprie pagine web, cerchiamo anche di dare qualche piccolo consiglio per creare **praticamente** i propri ipertesti. Dato che un documento HTML è un semplice file di testo, per crearlo è sufficiente un qualsiasi programma di text editing (p.es.

"blocco note" o "wordpad" per Windows, "gedit" o "kedit" per Linux, ecc.) senza caratteristiche aggiuntive. I file vanno salvati come testo semplice (cioè semplici sequenze di codici binari corrispondenti nella codifica ASCII o Unicode dei simboli alfanumerici, vedi Appendice 2), con l'unico accorgimento di rimuovere l'eventuale estensione .txt a volte aggiunta dai programmi Microsoft. Sebbene esistano in commercio editor specifici per la creazione di pagine web, in genere di tipo WYSIWYG (What You See is What You Get, cioè che consentono di lavorare sulla pagina così come viene visualizzata, come gli editor di testo impaginato tipo MS Word), ne sconsigliamo l'uso. Questi programmi in genere producono codice HTML quasi illeggibile e in genere poco corretto e rendono difficile trovare errori e malfunzionamenti (basta vedere il codice HTML generato da Word per farsi un'idea). Inoltre, allo stato attuale delle tecnologie, se vogliamo creare contenuto web senza capire come esso venga generato, ci conviene allora direttamente lavorare sulle interfacce dei Content Management System online in modo da avere immediato feedback sul funzionamento della pagina stessa sul server. Apriamo dunque un editor di testo e inseriamo il codice. Un codice semplice per una pagina web può essere quello di Figura 41. Se vogliamo salvarlo e vedere come verrà interpretato dal browser, non dovremo fare altro che salvarlo come testo semplice (Figura 42), e mettere nel nome del file l'estensione .html. Si noti che nell'interfaccia di Microsoft Windows, spesso le estensioni dei file sono nascoste e i file di testo semplice sono salvati automaticamente con estensione .txt. Questo deve essere evitato (basta modificare le preferenze delle cartelle). Altra cosa da notare, il testo semplice può essere codificato in modo differente. Il metodo migliore e oggi generalmente utilizzato è la codifica UTF-8 (Unicode), che supporta alfabeti di lingue differenti, se ci sono differenti codifiche o il browser non ha il corretto supporto ci potrebbe essere qualche problema nella visualizzazione dei caratteri speciali (vedi Appendice 2). A questo punto il documento può essere caricato sul browser per vederlo. Questo può essere fatto o selezionando il menu apri file sulla barra dei menu del browser, o inserendo nella barra degli indirizzi dello stesso, al posto

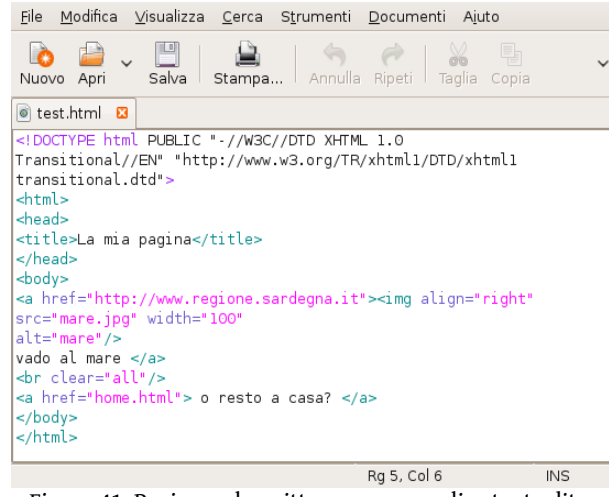


Figura 41: Pagina web scritta con un semplice text editor.

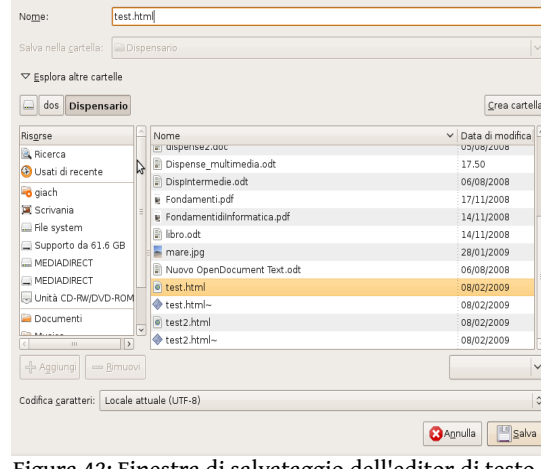


Figura 42: Finestra di salvataggio dell'editor di testo. Si noti la possibilità di scegliere la codifica dei caratteri.

dell'URL in rete, un indirizzo locale sul filesystem della macchina, preceduto dalla stringa "file://". Ancora, nei sistemi operativi moderni, che associano l'estensione del file al programma che gestisce tale tipo di file, il browser verrà automaticamente lanciato, caricando e visualizzando tale file, cliccando due volte sull'icona che lo rappresenta. Questa icona in Windows diventerà automaticamente quella del browser web non appena il file è stato salvato con l'estensione .html.
La pagina verrà, quindi, visualizzata correttamente. Ovviamente, i file collegati alla pagina web con indirizzo relativo dovranno effettivamente essere presenti nella cartella corrispondente per essere caricati e visualizzati dal browser. Lavorare sul codice testuale è, tutto sommato, molto più semplice e permette risultati più puliti e controllati che usare un editor HTML complesso. Ovviamente se poi vogliamo rendere le pagine visualizzabili in rete, dovremo trasferire le stesse dalla cartella del nostro PC ad una cartella pubblica su un server web, cioè su una macchina con indirizzo IP statico, sempre accesa e su cui è in esecuzione il programma server. In genere si affitta lo spazio sul server da un provider di servizi di hosting (vedi capitolo 3.10).

4.14 Form e programmazione server side

Anche se abbiamo fin qui descritto i documenti (X)HTML classici come semplici testi formattati con immagini e collegamenti ipertestuali, fin dagli albori del web, in realtà, i siti hanno supportato meccanismi di interazione più complessi, a cominciare dalla possibilità di inviare dati e richieste al server utilizzando delle apposite componenti di interfaccia.

In modo tale si può realizzare il cosiddetto web dinamico "server side" in cui a seguito dell'invio del messaggio di richiesta al server, viene generata in modo automatico una pagina di risposta dipendente dai dati inviati (ne abbiamo parlato nel capitolo 3.3), meccanismo che permette di realizzare svariati tipi di servizio. L'esempio più semplice di tale meccanismo è il **motore di ricerca**, in cui si inseriscono delle parole in una casella e si riceve una pagina (generata automaticamente da un programma eseguito sul server) con i collegamenti ai documenti ritenuti affini alle parole inserite. Ma siamo abituati ad utilizzare interfacce di questo tipo, ad esempio quando dobbiamo fare delle iscrizioni, compilare sondaggi, ecc.: in tutti i casi inviamo dati al server che li elabora, probabilmente li memorizza e comunque crea alla fine una pagina web contenente una risposta che verrà trasmessa al client per il rendering.

Dal punto di vista del codice (X)HTML, le interfacce per inviare le richieste al server si realizzano attraverso l'elemento `<form></form>` (in Italiano possiamo tradurre **form** con "**modulo**").

All'interno dei moduli, con l'annidamento di opportuni elementi HTML, si possono inserire vari elementi, come pulsanti, menu, campi testuali, ecc.
Si realizzano in questo modo interfacce classiche di tipo "**WIMP**" (Windows, Icons, Menu, Pointer), come quelle delle normali interfacce a finestre, ma sempre rappresentate all'interno della pagina web sul browser.

L'elemento HTML **form** contiene attributi: uno di essi (method) si riferisce al tipo di elaborazione realizzato coi dati e che tipicamente assume il valore "post". L'attributo "enctype" può specificare il tipo di codifica. Con l'attributo "action" si indica l'URL del programma (script) che risiede sul server ed elabora i dati, oppure, se si vuole che i dati vengano semplicemente inviati via email, si inserisce un indirizzo. Nel primo caso il codice sarà simile a quello qui sotto:

```
<form method="post" action="script.cgi">
...contenuto... </form>
```

nel secondo sarà invece simile a :

```
<form method="post" enctype="text/plain"
action="mailto:indirizzo@server.com">
...contenuto... </form>
```

Perché venga eseguita l'"action" dovrà essere presente nel form un pulsante "submit" che genera l'evento. Al clic su di esso (può essere anche un'immagine) vengono inviati i dati via HTTP ed il browser si mette in attesa della risposta. Gli altri elementi del form o modulo possono, come detto, essere caselle in cui inserire testo, menu di selezione, pulsanti di scelta. Per inserire **caselle di testo**, si utilizza l'elemento input con attributo `type="text"`, per esempio:

```
<input type="text" name="nomecasella", size="x",
maxlength="y" />
```

Qui "nomecasella" identifica il dato e viene usato dallo script di elaborazione. Se si vuole mettere un valore predefinito che compaia sulla finestra si deve inserire anche `value="valore_di_default"`. Se si inserisce `size="x"`, si fissa a x caratteri la dimensione della casella. Se non indicato è 20 caratteri. Con `maxlength="y"` si fissa a y il massimo numero di caratteri inseribili.
Spesso le caselle di testo vengono usate per l'autenticazione inserendo password, ed in questo caso si vuole che i caratteri inseriti nella casella non siano decifrabili. Questo effetto si realizza cambiando l'attributo type in `type="password"`:

```
<input type="password" name="nome", size="x", maxlength="y" />
```

Per inserimento di testo su più righe è utilizzato invece l'elemento **textarea**:

```
<textarea name="nome" rows="r"   cols="c">...</textarea>
```

Qui `rows="r"` determina il numero r di righe (default 4), `cols="c"` genera c colonne (default 40). Per selezionare tra opzioni predefinite abbiamo i **menu a discesa**, quelli in cui, cliccando appaiono le opzioni. Essi si realizzano con l'elemento `<select>`, con annidati elementi "option" che rappresentano le scelte possibili. Vediamo un semplice esempio:

```
<select name="nome" >
<option value="Scelta1">Scelta 1</option>
<option value="Scelta2">Scelta 2</option>
</select>
```

Se si vuole inserire un opzione di default, preselezionata al caricamento della pagina si usa l'attributo `selected="selected"`:

```
<option value="Scelta2" selected="selected">Scelta 2</option>
```

Se si vuole poter scegliere più opzioni si usa:

```
<select name="nome" multiple="multiple">
```

Altro elemento è costituito dai **radio button**: pulsanti multipli di cui solo uno può essere attivo (pulsanti di opzione). Il nome radio deriva dal fatto che simili pulsanti fisici erano utilizzati per il cambio delle stazioni nelle vecchie radio. Per realizzare un insieme di pulsanti di questo tipo si usa elemento input per ogni opzione, con tipo "radio" e con lo stesso nome e diversi valori, come nell'esempio seguente:

```
<input type="radio" name="nomeseriepulsanti"  value="1"/>
<input type="radio" name="nomeseriepulsanti"  value="2"/>
```

Per associare un nome sull'interfaccia conviene poi aggiungere accanto l'etichetta del pulsante (non necessariamente uguale al valore). Per inserire del testo come etichetta all'interno di un modulo ed identificarlo, si può usare l'elemento **label**:

```
<label for="nome"> Testo </label>
```

Per attivare il pulsante per default si aggiunge al pulsante `checked="checked".`

Le **checkbox** (caselle di controllo) sono simili ai radiobutton possono essere attivate contemporaneamente:

```
<input type="checkbox" name="nomeseriecaselle"  value="1"/>
<input type="checkbox" name="nomeseriecaselle"  value="2"/>
```

Spesso si vogliono elaborare anche dati che l'utente non deve controllare o inserire, ma sono magari ricavati precedentemente dal server. Si usano input non visualizzati nella pagina (i cosiddetti **campi nascosti**), che vengono inviati con l'invio della richiesta insieme agli altri. La sintassi è:

```
<input type="hidden" name="nome" value="valore"/>
```

Il modulo non avrebbe senso senza inserire il **pulsante di invio**. Esistono differenti modalità per realizzarli. Di norma si usa l'elemento input con attributo "type="submit", es:

```
<input type="submit" value="etichetta"/>
```

Qui "etichetta" è quello che apparirà scritto sul pulsante. Un'alternativa è utilizzare l'elemento <button> che permette di inserire all'interno un'immagine o grafica:

```
<button type="submit" name="invia" value="valore"/>
    etichetta (testo a lato)
<img src="immagine.gif" alt="immagine_invio"> </button>
```

Oltre al tasto di invio, può essere comodo inserire un tasto per cancellare tutti i valori inseriti nel form. Per realizzare questo, basta utilizzare l'elemento input con attributo `type="reset"` (o `button type="reset"`).

Esistono altri possibili elementi ed opzioni per i form, ma non ci dilungheremo ulteriormente, rimandando come sempre alla manualistica per eventuali approfondimenti coloro che vogliono o devono realizzare praticamente le pagine web. In Figura 43 viene mostrato il codice di un form ed il suo rendering sul browser.

Torniamo, invece, a parlare di cosa accade quando inviamo dati al server tramite essi. Al clic sul tasto invio il browser invia al server un messaggio codificato in standard HTTP che contiene tutti i dati in esso inseriti. Il server elabora i dati stessi utilizzando il programma indicato dall'indirizzo scelto. Questo programma eseguirà quindi elaborazioni più o meno complesse, di solito collegandosi anche a un sistema di gestione di database per la memorizzazione o la ricerca di dati. Al termine dell'elaborazione, esso genera una pagina web (creata "dinamicamente"), che viene trasmessa al client, che la visualizza. Si realizza così una sorta di

```
<form method="post"
enctype="text/plain" action=
"mailto:indirizzo@server.com">

Nome <input type="text"
name="nomecasella" size="10"
maxlength="10" />
<br/>
<textarea name="commenti" rows="3"
cols="65">Commenti?</textarea>
<hr />
<select name="nome" >
<option value="Scelta1">
Scelta 1</option>
<option value="Scelta2" selected="selected">Scelta 2</option>
</select>
<hr />
<input type="radio" name="nomeseriepulsanti" value="1"/>
<input type="radio" name="nomeseriepulsanti" value="2"/>
<hr />
<input type="checkbox" name="nomeseriecaselle" value="1"/>
<input type="checkbox" name="nomeseriecaselle" value="2"/>
<br/><br/>
<input type="submit" value="invio"/>
<input type="reset" value="reset"/>
</form>
```

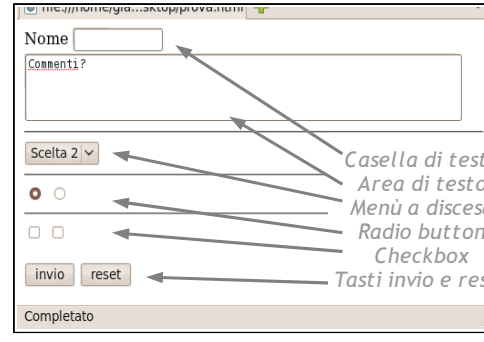


Figura 43: Codice di modulo e suo rendering grafico sul browser.

servizio interattivo, ove il sistema web client+server risponde ad un'attività dell'utente. Nell'HTML originario il meccanismo di interfaccia standard tra il server HTML e l'esecuzione dei programmi era detto CGI (Common Gateway Interface). Oggi si usano più spesso differenti tecnologie, che consistono nell'uso di programmi scritti in linguaggi che vengono interpretati da software installato sul server (es. PHP, ASP, Perl). Come si può facilmente intuire, quindi, realizzare servizi web in questo modo richiede competenze informatiche molto superiori alla realizzazione di semplici ipertesti, che possono essere realizzati semplicemente anche da persone prive di competenze informatiche molto avanzate. Per realizzare servizi relativamente classici, è possibile eventualmente usare programmi messi a disposizione da archivi online. I provider di hosting web in genere segnalano il supporto a linguaggi di scripting server side come PHP o ASP e l'eventuale possibilità di utilizzare un database.

4.15 Scripting client side e HTML dinamico

Quello di inviare dati al server che genera al volo una pagina dipendente da essi non è l'unico modo di realizzare pagine web cosiddette **dinamiche** o **interattive** (cioè che si modifichino sulla base delle azioni dell'utente). Un differente modo di interagire è il cosiddetto "**client side**", cui abbiamo già fatto cenno nel capitolo 3. In questo caso, al codice (X)HTML della pagina sono associati programmi (script) eseguibili da interpreti integrati nei browser e che possono generare la visualizzazione di contenuti ed effetti grafici dipendenti delle azioni dell'utente (cioè interattivi), senza collegarsi al server.

Ovviamente le finalità delle funzionalità interattive "server side" e "client side" sono completamente diverse: le prime sono lente (richiedono connessione e trasferimento dati), ma possono utilizzare informazioni pressoché illimitate e ovunque situate sulla rete. Sono quindi adatte per realizzare il cosiddetto feedback "semantico" (rispondere a richieste di informazioni nuove). Le seconde sono, al contrario, veloci (non c'è connessione), ma richiedono il precedente trasferimento di programmi sul client, i quali, quando eseguiti, accederanno soltanto a informazioni già presenti sul client. L'interattività "client side", è adatta, quindi, per realizzare feedback grafico, di interfaccia. Questo tipo di effetti dà origine a quello che prende il nome di HTML dinamico (DHTML). Esso permette di modificare interattivamente tutti gli elementi della pagina, considerati come oggetti con determinate proprietà. Il codice inserito nelle pagine può ad esempio associare azioni eseguite dall'utente su parti della pagina (ad esempio il clic su un testo, il passaggio del mouse su una sezione) a modifiche della pagina stessa (cambio di colore o di dimensione di un testo, caricamento di nuove immagini, calcolo del risultato di un'elaborazione dati e rappresentazione dello stesso in qualche sezione della pagina, e così via. Alcuni esempi di modifiche interattive possono anche essere realizzati con semplici regole di stile CSS attribuite alle parti della pagina, ma, in generale si utilizzano a questo scopo veri e propri programmi, in genere scritti in un linguaggio di programmazione chiamato **Javascript**.

Il codice Javascript può essere direttamente inserito all'interno di un documento HTML attraverso l'uso dell'elemento `<script>` dell'HTML. Ad esempio:

```
<script type="text/javascript" language="JavaScript">
document.write("Ciao a tutti!") ;
</script>
```

crea una pagina con scritto “Ciao a tutti” in quanto ordina all'interprete di scrivere sulla pagina del documento visualizzato dal browser, tale testo. Qui `type` indica il linguaggio utilizzato per il programma, che, in genere, è Javascript, ma non lo è necessariamente.

L'interprete Javascript è disponibile nei principali browser, se per caso gli effetti dinamici non funzionano è possibile però che esso sia disattivato (controllare in tal caso le opzioni del browser stesso). Per rendere il codice HTML contenente script compatibile con i vecchi browser che non supportano il linguaggio, è possibile “nascondere” il codice Javascript in un commento:

```
<script type="text/javascript" language="JavaScript">
<!--  Codice programma // --> end comments to hide scripts
</script>
```

In tal modo se il codice non viene interpretato, non viene neppure scritto il suo testo sulla pagina, in quanto considerato un commento. Per i browser che non supportano lo scripting, è anche possibile inserire contenuto alternativo che non viene visualizzato altrimenti:

```
<noscript>Il tuo browser non supporta gli script</noscript>
```

Scopi principali di questi “script”, sono, come detto, la scrittura dinamica di parti del documento HTML sulla base di azioni dell'utente, la modifica del layout sulla base di azioni dell'utente, il controllo e la generazione di messaggi (di conferma, di errore) .

Le modifiche interattive della pagina possono essere realizzate grazie al fatto che i programmi Javascript possono accedere alle varie parti della pagina e del browser visti come variabili del programma attraverso un meccanismo detto **Document Object Model** o **Browser Object Model**.

In pratica le parti del documento sono accessibili attraverso una gerarchia di “oggetti” del linguaggio (un oggetto in programmazione è caratterizzato da valori di dati che possono essere variati e funzioni caratteristiche che ne definiscono il comportamento).

Tutti gli elementi inclusi nell'HTML sono accessibili dalla radice “document”. Gli altri oggetti nella finestra del browser sono accessibili a partire dalla radice “window”. Così se, per esempio, si vuole scrivere qualcosa sulla pagina, si può, come nell'esempio precedente, utilizzare la funzione write dell'oggetto document. Il tutto si scrive, nella sintassi di Javascript nel seguente modo: `document.write("ciao!").`

Il linguaggio mette a disposizione poi molte funzioni che permettono di agire sugli oggetti della pagina. Alcune di esse permettono, come i selettori delle regole CSS, di selezionare l'elemento della pagina desiderato su cui applicare le modifiche. Ad esempio la riga di programma:

```
document.getElementById('testo').value='ciao'
```

seleziona dalla pagina HTML l'elemento con id=”testo” e assegna il valore ciao. Quindi, se nella pagina ci fosse, ad esempio la seguente sezione:

```
<div id=”testo”>Pippo</div>
```

all'esecuzione del codice la pagina viene modificata: al posto di “Pippo” comparirebbe scritto “ciao”. Allo stesso modo si possono variare le proprietà di formattazione con apposite funzioni. Per esempio, possiamo nel codice Javascript collegato a una pagina HTML definire la funzione:

```
function ChangeFont(n)
{ document.getElementById("id").style.fontSize = n; }
```

Anche non conoscendo la sintassi del linguaggio in dettaglio, si può intuire che, quando questa funzione viene eseguita, le dimensioni del carattere nell'elemento identificato dall'identificatore “id”, vengono cambiate. Analogamente, quindi, esisteranno funzioni che mi permettono di modificare tutti gli oggetti nella pagina eseguendo un codice opportuno.

Resta da capire come si possa fare eseguire il codice in funzione di un'azione dell'utente sull'interfaccia del browser. La risposta è semplice: mediante il Document Object Model del browser è infatti anche possibile accedere alle azioni operate sugli oggetti. Ad ogni elemento, cioè, sono associati “eventi” che possono essere associati all'esecuzione di una funzione Javascript. Tali eventi sono, per esempio, il caricamento di una pagina, il click e rilascio del mouse sull'elemento, il click semplice del mouse, la sovrapposizione del mouse all'oggetto, la selezione di parte di testo.

Gli eventi vengono attivati attraverso corrispondenti attributi inseriti nell'elemento HTML. Per esempio, se viene scritto:

```
<div style="font-size:36px; background:red;
  width:200px; text-align:center;"
  onmouseover="this.style.background='grey'"
  onmouseout="this.style.background='red'">
sezione </div>
```

le proprietà dell'elemento identificato da `<div>` vengono variate attraverso il movimento del mouse. Quando il cursore è sopra l'elemento (onmouseover) lo sfondo diventa grigio, quando esce diventa rosso. La variabile **this** identifica nel linguaggio Javascript l'elemento in cui ci troviamo (lo specifico elemento div in questo caso).

Ovviamente un'azione su un pulsante può far modificare anche tutti gli altri elementi del documento, non solo quello ove si verifica l'azione. Si consideri l'esempio seguente:

```
<img id=”personaggio” src=”pluto.png” alt=”pippopluto”>
<form name="mioForm">
<input type="button" value="cambiaimmagine" onclick=
"document.getElementById('personaggio').src='pippo.png'">
</form>
```

In questo caso, l'evento generato da un elemento button, modifica un immagine caricata nella pagina, selezionato dall'id “personaggio”, cambiando l'attributo corrispondente all'url della stessa (provare per credere). Anche se son si conoscono in dettaglio i principi della programmazione, gli esempi mostrati dovrebbero chiarire i concetti base dell'HTML dinamico.

Con un po' più di esperienza nella programmazione, naturalmente, si potrebbe utilizzare il codice Javascript per realizzare vere e proprie applicazioni interattive (giochi, calendari, grafici), come avviene, del resto, in molti siti. Essendo un linguaggio di programmazione di alto livello, Javascript consente infatti di effettuare calcoli matematici e di implementare le strutture tipiche degli algoritmi (sequenze di operazioni, selezione, iterazione), ma rimandiamo chiaramente coloro che volessero approfondire questo aspetto a testi specialistici.

4.16 Multimedia, plugin ed helper

La multimedialità delle pagine web vista finora si limita all'inserimento di immagini digitali all'interno delle pagine con l'elemento `<img />`. Tutti sanno, però che i moderni siti web includono spesso molti altri tipi di informazione: audio, filmati, visualizzazioni interattive di oggetti tridimensionali e via dicendo. L'evoluzione dall'iniziale ipertesto degli albori del web a questo contenuto "ricco" è stata graduale ed ha via via integrato le funzionalità aggiunte dai produttori dei browser (in primis Netscape) nella codifica HTML e poi XHTML ufficiale (inizialmente gli ipertesti non contenevano neppure colori ed immagini).

Un modo per rendere accessibili contenuti digitali di vario tipo da una pagina web lo abbiamo, in realtà, già visto ed è il semplice utilizzo del collegamento ipertestuale o link. Quando a un browser, cliccando sulla relativa àncora, è chiesto di caricare un file differente dall'HTML, questo può essere visualizzato sull'interfaccia del browser stesso, se supportato (come avviene per le immagini .gif, .jpg o .png), oppure attraverso un programma diverso (**applicazione helper** o **player**) che è in grado di visualizzare quel tipo di file e che è registrato sul browser per essere utilizzato per tale scopo. In caso non vi siano programmi adatti, il browser può proporre di salvare il file sul filesystem locale.

Di solito l'associazione del tipo di file al programma helper viene gestito con le preferenze del browser. Ad ogni tipologia di file, identificata dal suo MIME-type viene associato il programma che li legge. L'installazione di nuovo software per leggere contenuti multimediali va in genere anche a modificare le impostazioni del browser stesso per aggiornarlo e quindi può modificare il modo in cui vengono visualizzati i file.

Se invece si vogliono integrare i contenuti multimediali all'interno delle pagine, esistono appositi elementi di (X)HTML per includerli ed apposite componenti software che realizzano l'inserimento del programma all'interno della visualizzazione della pagina. Queste componenti sono dette **plugin**: esse non possono essere eseguiti autonomamente, ma funzionano soltanto all'interno del particolare browser.

Netscape, il browser che ne introdusse l'uso, introdusse anche l'elemento HTML **`<embed>`** per inserire la visualizzazione del file in modo che venga integrata in quella della pagina. Il meccanismo fa sì che l'interfaccia del plugin venga inserita con le caratteristiche desiderate (dimensione, posizione, opzioni, ecc.). L'elemento `<embed>` venne poi supportato da altri browser come Internet Explorer, quindi divenne molto usato ed è per questo che viene ancora supportato dai browser, nonostante il fatto che l'elemento HTML standard poi introdotto dal W3C per lo stesso scopo scopo sia invece invece il similare **`<object>`**. L'elemento **`<object>`**, in teoria, dovrebbe in futuro uniformare l'inserimento di tutti i tipi di contenuto non testuale (anche, per esempio, le immagini e i programmi interattivi Java e Flash che vedremo nel seguito). A causa di questa duplicità di sintassi e del gran numero di implementazioni di browser e di oggetti multimediali, ci sono a volte problemi di compatibilità nell'uso dei plugin. La soluzione che in genere si adotta su gran parte dei siti è quella di usare

```
<object width="425" height="344">
<param name="movie"
value="interazione.AVI"></param>
<embed src="interazione.AVI"
width="425" height="344"></embed>
</object>
```

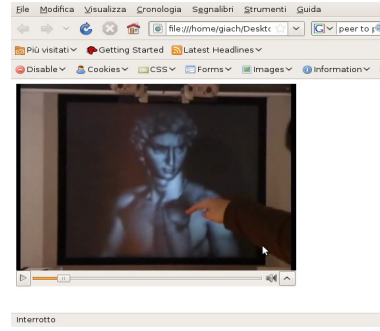


Figura 44: Codice per l'inserimento di un semplice video in formato avi (usando sia object che embed per maggiore compatibilità coi differenti browser) e risultato ottenuto.

entrambi gli elementi (object ed embed) in modo da garantire la massima compatibilità. Quello che accade quando gli elementi object (embed) vengono interpretati dal browser, è che viene verificata la presenza di un plugin adatto; se questo viene trovato, l'interfaccia relativa viene inserita all'interno della finestra della pagina web con le dimensioni indicate, se non viene trovato potrebbe essere visualizzato uno spazio bianco al suo posto. Spesso oggi si fa in modo che in tal caso il browser cerchi automaticamente in rete il plugin mancante da installare.

Object può avere molti attributi, per es. **data** per inserire eventualmente l'url del contenuto multimediale, **type** per determinarne il tipo, **width** e **height** per determinare le dimensioni della componente di interfaccia inserita nella pagina. Ogni tipo di plugin avrà poi i suoi parametri specifici (ad esempio relativi all'URL del contenuto ai controlli, ecc.): questi si inseriscono con elementi `<param>` annidati dentro `<object>`. Non è nostro interesse vedere tutte le possibilità a disposizione né insegnare la sintassi esatta, per cui vedremo soltanto qualche esempio per dare un'idea di come si possano inserire i vari tipi di contenuto. Ad esempio, per inserire un file audio, in formato midi, possiamo usare object come segue:

```
<object type="audio/x-mid" data="nome_midi.mid"
height="40" width="170">
<param name="src" value="nome_midi.mid"></param>
<param name="autostart" value="true"></param>
<param name="loop" value="true"></param>
</object>
```

In questo caso viene creato un elemento di interfaccia di dimensione 40x170 pixel, che avvia automaticamente (`autostart=true`) l'audio, e lo fa ripartire dall'inizio a fine esecuzione (`loop=true`). Lo stesso tipo di sintassi si utilizza per i video, per esempio:

```
<object type="video/quicktime"
data="http://www.miosito.com/video.mov"
```

```
width="320" height="256">
<param name="autoplay" value="false" />
<param name="controller" value="true" />
</object>
```

In questo caso l'interfaccia inclusa è di dimensioni 320x256 pixel, i comandi di controllo dell'esecuzione sono visibili (`controller=true`, in genere appaiono i tasti start, stop e pausa e la barra temporale), e il filmato parte solo quando si schiaccia play (`autoplay=false`). Se si vuole garantire la massima compatibilità rendendo possibile sia l'inserimento con object che alternativamente con embed, si possono utilizzare entrambi in alternativa.

Per convincerci che è probabilmente questo il metodo che garantisce la massima accessibilità, si può osservare che il codice che viene proposto sul celebre sito YouTube per permettere agli utenti di inserire nelle proprie pagine i video archiviati sul sito stesso è strutturato in maniera identica. Questo è un esempio preso dal sito stesso:

```
<object width="425" height="344">
<param name="movie" value="(url del video)"></param>
<param name="allowFullScreen" value="true"></param>
<param name="allowscriptaccess" value="always"></param>
<embed src="(url del video)"
type="application/x-shockwave-flash"
allowscriptaccess="always" allowfullscreen="true"
width="425" height="344"></embed></object>
```

e che usa object e in alternativa embed richiamando il plugin per la visualizzazione dei video in formato shockwave-flash.

Quando si include un oggetto multimediale con object o embed è infine anche possibile fare in modo di indicare dove si dovrebbe scaricare il plugin per gestire il contenuto multimediale, se assente. L'opzione si abilita attraverso l'attributo “pluginspage” in embed e quello “codebase” in object.

Data l'evoluzione recente dei vari tipi di contenuto multimediale e plugin, ci possono a volte essere problemi di compatibilità ed è sempre meglio cercare di inserire tutte le possibili alternative per rendere i contenuti fruibili anche con i vecchi browser. Molto spesso si trovano esempi come il seguente:

```
<object width="160" height="144"
classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"
codebase= “http://www.apple.com/qtactivex/qtplugin.cab">
<param name="src" value="sample.mov">…
```

Lo strano attributo **classid** è un codice relativo al controllo ActiveX (un elemento software

Microsoft) da utilizzare per visualizzare quel tipo di contenuto e che permette eventualmente di scaricare automaticamente il plugin se assente. Ovviamente tale codice è utilizzato solo dal browser Internet Explorer.

Consigliamo, comunque, nel caso si inseriscano contenuti di tipo particolare, di riferirsi alle istruzioni fornite da chi gestisce il formato di dati in questione e di fare le opportune prove con diversi browser per verificare le compatibilità.

L'elemento object non è solo il metodo standard per inserire video, animazioni e audio, ma, per il W3C, dovrebbe diventare il metodo per includere qualunque tipo di oggetto, anche, per esempio i documenti in formato .pdf:

```
<object data="documento.pdf" width="450" height="450"></object>
```

e i vari tipi di programmi interattivi (Java, Flash) che vedremo nei prossimi paragrafi.

Dato che per il W3C l'elemento object dovrebbe essere lo standard per l'inclusione di tutti i contenuti speciali all'interno delle pagine, anche le immagini possono essere inserite con esso, evitando l'uso dello specifico elemento `<img>`, ma scrivendo, invece:

```
<object data="foto.png" type="image/png"> </object>
```

Inclusione di oggetti interattivi: gli applet Java

Tra gli oggetti integrabili in una pagina web utilizzando l'elemento `<object>` vanno sicuramente segnalati programmi interattivi realizzabili con linguaggi di programmazione i cui interpreti (cioè i programmi che traducono le istruzioni del linguaggio in attività del calcolatore) sono integrabili nel browser.

Il primo tra questi ad essere stato proposto per l'arricchimento delle pagine web è il linguaggio Java, creato proprio allo scopo di realizzare applicazioni distribuibili in rete da Sun Microsystem. Java, che non ha nulla a che vedere con il già citato Javascript, è un linguaggio di programmazione di alto livello, che ha la peculiarità di essere un linguaggio interpretato, tale che, cioè, per eseguire il codice occorra installare sulla macchina un programma (interprete) che lo decodifichi e trasformi passo passo in linguaggio macchina. La peculiarità di Java rispetto ad altri linguaggi interpretati (come ad esempio, i vari Javascript, php, asp di cui abbiamo parlato), è che esso non viene distribuito in forma di codice sorgente contenente testo comprensibile, ma, prima di essere distribuito ed eseguito viene trasformato in un cosiddetto “bytecode”, cioè un codice più efficiente da convertire in linguaggio macchina rispetto al testo originario, ma comunque platform-independent, cioè indipendente dalla piattaforma utilizzata per l'esecuzione (come sarebbe invece il codice tradotto in linguaggio macchina). Il meccanismo con cui i programmi in Java sono utilizzati nelle pagine web è il seguente: un interprete del linguaggio detto Java Virtual Machine è stato reso integrabile nei comuni browser e permette di inserire delle componenti software grafiche realizzate in Java all'interno della pagina stessa, in un area di dimensioni determinate. Questi particolari programmi vengono detti “**applet**” e permettono quindi di integrare nella pagina parti interattive (es. giochi, interfacce attive, animazioni controllabili dall'utente, ecc.). L'integrazione veniva in origine realizzata con un apposito elemento HTML, `<applet>`, oggi sostituito teoricamente da `<object>`.

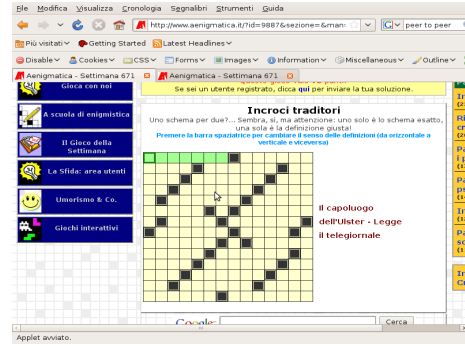


Figura 45: Il cruciverba interattivo di questo sito è un classico esempio di Applet Java.

Quando gli applet Java vennero introdotti, il loro uso era di fatto l'unico modo per avere interattività e programmi in esecuzione sul client, cosa oggi consentita da altre tecnologie spesso più vantaggiose come Javascript o Flash. Un punto di forza dell'uso della tecnologia Java risiede sicuramente nel fatto che esso sia stato progettato appositamente per l'utilizzo in rete garantendo nell'esecuzione sul browser la sicurezza del computer dell'utente (l'esecuzione di codice non controllato dall'utente, potrebbe infatti danneggiare il PC dell'utente se non sono utilizzati sufficienti misure di sicurezza). Altra caratteristica interessante è l'ampia disponibilità di librerie (in pratica software già scritto e disponibile ai programmatori) utili per realizzare applicazioni grafiche e di networking.

Sebbene l'uso degli applet sia abbastanza in ribasso, data la concorrenza di nuovi e più avanzati formati e plugin per l'inserimento di contenuti interattivi, è possibile, per chi avesse interesse, reperire codici per inserire componenti interattive realizzate con tale tecnologia presso archivi in rete, come *http://www.jars.com*. Essi possono essere utilizzati per arricchire le proprie pagine web, anche se si deve ricordare che una buona norma per garantire l'usabilità dei siti consiste nell'evitare di appesantire le pagine stesse con effetti inutili e codice che, quando eseguito, occupa memoria e tempo di calcolo processore con il rischio di rallentare sensibilmente le prestazioni del calcolatore.

Inclusione di oggetti interattivi: Flash e animazioni interattive

Un altro "oggetto" particolare che viene spesso integrato all'interno delle pagine web (con il solito metodo del plugin) è il codice interattivo generato con un altro linguaggio proprietario noto come Flash, generato e interpretato da software proprietario dell'azienda Adobe inc. Si tratta di un linguaggio molto efficiente e potente che permette di creare animazioni complesse, interattive e multimediali. All'interno di esse infatti si possono integrare, forme vettoriali, testo (sia statico sia dinamico) e caselle di input per il testo, immagini raster (BMP, GIF, JPEG, PNG, TIFF e altri formati, vedi capitolo 5), audio (MP3, WAV e altri, vedi capitolo 5), sia in streaming che per effetti sonori, video (AVI, QuickTime, MPEG, Windows Media Video, FLV, vedi capitolo 5).
Diversi siti sviluppati recentemente realizzano tutta l'interazione mediante l'esecuzione di un codice Flash oppure spesso vengono introdotte nei siti HTML animazioni interattive introduttive realizzate in tale linguaggio. Con Flash vengono realizzati anche giochi ed interfacce grafiche.
L'utilizzo di tale tecnologia non è, in generale, sempre consigliabile: ricordiamo infatti che per quanto riguarda Flash, valgono le stesse considerazioni che possiamo fare per Java e per qualunque altra tecnologia che potrebbe prenderne il posto in futuro: i contenuti non standard dei siti vengono sempre visualizzati attraverso i "plugin", questo vuol dire che i produttori dei nuovi formati devono creare visualizzatori integrabili in modo standard nei principali browser. Questo fa sì che possano esserci delle incompatibilità tra i differenti plugin. Inoltre, realizzando

sessioni interattive interamente gestite dal programma “incorporato” (in inglese si dice “embedded”) dentro la pagina, si inibisce la normale navigazione tra le pagine che gli utenti sono abituati a realizzare anche utilizzando i tasti e le opzioni del browser. Il plugin per utilizzare i contenuti Flash si chiama Flash Player ed è disponibile per tutte le principali piattaforme; inoltre esistono flash player standalone (cioè utilizzabili senza browser) e per molti dispositivi differenti come console (es. Nintendo Wii, Playstation) ed una versione per cellulari. Esistono anche varie implementazioni open source di player Flash (es. Gnash).

Rich Internet Applications

Infine, vengono ormai sempre più spesso integrate nelle pagine web mediante plugin anche applicazioni interattive veloci che elaborano dati sul client ma che mantengono gran parte dei dati e dell'applicazione sul server. Le tecnologie più note che vengono utilizzate per questo tipo di applicazioni sono Adobe Flex, che a livello di interfaccia utente richiede l'installazione di Flash Player, Microsoft Silverlight e Java FX.
Troviamo già molti siti che utilizzano tali tecnologie, specialmente nell'ambito televisivo e di intrattenimento.
Si noti che le tecnologie sviluppate per incorporare i contenuti interattivi e multimediali nelle interfacce dei browser web sono le stesse che vengono utilizzate e supportate nei nuovi dispositivi portatili come smartphone, tablet, eccetera. Nella lotta commerciale tra i produttori di tali tecnologie si definiranno gli scenari futuri per la distribuzione dei contenuti e l'uso futuro di Internet, che probabilmente si affrancherà almeno in parte dalla navigazione realizzata con il classico browser.

HTML5

Il lettore potrebbe pensare che, se da una parte tutti questi plugin che gestiscono contenuti multimediali sono molto potenti e consentono numerose applicazioni, sarebbe più opportuno che anche l'accesso a tali tipi di contenuto fosse standardizzato per garantire accessibilità e compatibilità. Il nuovo linguaggio standard del web **HTML5** dovrebbe spingere verso una standardizzazione aperta dell'inclusione di contenuti multimediali, con nuovi elementi come **`<audio>`** e **`<video>`** attraverso i quali inserirli senza utilizzare formati proprietari e plugin di singoli produttori, ma con il semplice adattamento dei vari browser allo standard.

Il problema è che l'implementazione di tali meccanismi di integrazione non è semplice e le aziende che gestiscono i contenuti devono cercare di utilizzare i formati più diffusi e meglio implementati per garantire la qualità e la diffusione dei propri oggetti multimediali. Come vedremo a volte anche i formati di codifica audio e video migliori non sono di pubblico dominio e non possono così essere inseriti liberamente nei browser i relativi decodificatori. Infine gli standard aperti non garantiscono per definizione la protezione dei contenuti, mentre molti siti non vogliono che l'utente salvi e copi liberamente filmati e audio sul proprio computer. Dunque, sebbene HTML5 sia in corso di test e supportato parzialmente dai vari browser, è difficile prevedere se lo standard soppianterà i vari plugin proprietari.

5 Contenuti multimediali: codifica ed elaborazione

In quest'ultima parte del testo ci occuperemo di capire un po' meglio le caratteristiche dei vari tipi di contenuti multimediali che possiamo includere in una pagina web od utilizzare con altri tipi di interfaccia o terminale di uso comune.

Questa sezione è, in un certo senso, più importante delle altre, da un punto di vista pratico, per chi vuole utilizzare il web per diffondere contenuti e comunicare. Se infatti è possibile grazie ai Content Management System e ai siti tipo "Web 2.0" evitare di ricorrere alla codifica diretta delle pagine web utilizzando strumenti guidati per inserire contenuti multimediali, per realizzare immagini, suoni, filmati di cui si possa fruire efficacemente in rete non ci si può esimere da una certa conoscenza di base delle metodologie di codifica di tali dati ed anche, quindi, dei formati di salvataggio e trasmissione, dei plugin usati per la visualizzazione e dei programmi disponibili per la creazione.

Abbiamo già detto che codificare un tipo di dato vuole dire scriverlo sotto forma di sequenza di bit, definendo ovviamente un formato standard tale che chi invia/scrive e chi riceve/legge i dati possano comprendersi. Anche per codificare i numeri usati per fare calcoli sui PC occorre definire delle convenzioni per esprimerli univocamente in cifre binarie. Ad esempio, per i numeri naturali si possono semplicemente convertire i decimali in binari, mentre per gli interi si usa il formato detto "complemento a due" per rappresentare anche il segno e per i numeri reali occorre definire delle approssimazioni opportune (il formato che si usa si chiama "in virgola mobile" e rappresenta con la stessa precisione numeri grandi e numeri decimali).

Anche per il semplice testo la codifica non è ovvia né univoca. Occorre chiaramente stabilire una corrispondenza tra numeri espressi da una sequenza di bit e simboli dell'alfabeto. Si sono evoluti standard diversi (ASCII, Unicode, vedi Appendice 2) ed esistono numerose varianti legate ai diversi linguaggi ed alfabeti. Le problematiche correlate alla codifica del testo sono comunque limitate: una volta determinato il set di caratteri della lingua non ci sono ambiguità ed eventualmente si possono usare le sequenze HTML per i caratteri speciali. Il testo non crea comunque problemi di dimensioni dei file, per cui non richiede particolari attenzioni per ridurre al massimo l'occupazione di memoria della codifica. Come vedremo, per tutti gli altri tipi di dati, la dimensione in bit degli oggetti e la qualità della rappresentazione diventeranno, invece, critici. Per fare un piccolo esempio, con dei numeri precisi, consideriamo la codifica digitale dei caratteri di un libro. La Divina Commedia contiene all'incirca mezzo milione di caratteri: dal momento che ogni carattere occupa con la codifica ASCII estesa 1 byte (cioè 8 bit), l'occupazione di memoria è di circa mezzo Megabyte (1 Megabyte = 1 milione di byte). Se fossero validi i valori nominali, quindi, potrebbe essere trasferito con una connessione ADSL (bit rate p.es. 2 Megabit/s) in un paio di secondi ed in un CD ROM (capienza 640 Mbyte) potrebbe essere contenuto oltre mille volte senza alcuna forma di compressione.

Se consideriamo le immagini, invece, vedremo come il colore di ogni punto sia codificato con 3 byte e che, per esempio, una macchina fotografica digitale produce immagini con milioni di punti (Megapixel). Quindi una sola foto, neppure ad altissima definizione, occuperebbe molta più memoria sul calcolatore dell'intera Divina Commedia, se non si adottassero tecniche di compressione al fine di ridurne le dimensioni della codifica.

E che dire del video? Per avere l'impressione del movimento dobbiamo rappresentare in

sequenza almeno 10 immagini al secondo, che devono ovviamente essere tutte memorizzate nel calcolatore. Quindi ogni secondo di video, anche se limitiamo la dimensione delle immagini a quelle usate per la televisione a bassa risoluzione (ad esempio l'immagine dei comuni DVD ha circa 340000 punti), occuperebbe nella memoria dei calcolatori 10x3x340000 byte, cioè oltre 10 Megabyte. Anche qui le tecniche per comprimere i dati diventano cruciali. Prima di parlare di compressione, tuttavia cominciamo a capire come si memorizzino immagini, suoni e dati sul calcolatore in modo tale da poterli elaborare e riprodurre. Il primo passo è quello di tradurre il segnale fisico che li caratterizza in numeri e viceversa rigenerarlo a partire dai numeri stessi.

5.1 Misure fisiche, campionamento e quantizzazione

Immagini e suoni sono la nostra percezione di quantità fisiche, cioè intensità di radiazione elettromagnetica o variazioni di pressione dell'aria. Per trasformarle in numeri elaborabili dal PC è quindi innanzitutto necessario avere dei sensori in grado di misurarne i valori in determinati punti e degli strumenti in grado di trasformare i valori ottenuti in dati binari. Questi due procedimenti si chiamano, rispettivamente, **campionamento** (non potendo registrare i valori in tutti i punti dello spazio o in tutti gli istanti di tempo si fanno quindi misure campione in un insieme di posizioni/tempi determinato) e **quantizzazione** (trasformare un valore continuo in uno che varia a passi discreti, ed in cui la differenza tra la misura corrispondente ad unità successive determina il limite dell'accuratezza della riproduzione, cioè l'errore di quantizzazione).

La densità con cui il segnale viene campionato determinerà in qualche modo l'accuratezza con cui potrò riprodurne le variazioni nello spazio o nel tempo (se catturo la luce con un sensore che acquisisce molti punti, potrò generare immagini con grande dettaglio, se ho un sensore che cattura il suono con elevata frequenza, cioè campionando l'onda molte volte al secondo, lo potrò riprodurre in modo più fedele). La quantizzazione è altrettanto critica: se si usano pochi bit per codificare la luminosità di un'immagine, non si potranno differenziare e riprodurre variazioni fini di colore, se se ne usano molti l'occupazione di memoria cresce rapidamente. Vedremo a proposito esempi specifici per le diverse tipologie di dato.

5.2 Le immagini digitali

Le immagini digitali o meglio le immagini digitali di tipo **bitmap** (o **raster**) sono il risultato del campionamento di un segnale luminoso su una griglia rettangolare di dimensioni NxM. Le macchine fotografiche digitali ottengono le immagini attraverso griglie o “array” di sensori che misurano la quantità di fotoni che giungono intorno alle posizioni dei punti, ove la luce, come nelle comuni macchine fotografiche, viene convogliata da un sistema ottico. Immagini di questo tipo possono anche essere acquisite anche con altri tipi di strumento (es. scanner, macchinari diagnostici, ecc.) o generate artificialmente.

In ogni caso, quello che è memorizzato nel file digitale che codifica l'immagine è una matrice rettangolare contenente alle posizioni individuate da coppie di numeri interi (indici di righe e colonne), la codifica del colore o del livello di luminosità relativo al punto dell'immagine corrispondente, detto **pixel** (contrazione di picture element).

L'immagine verrà riprodotta poi su carta o schermo colorando in funzione del valore memorizzato tutto il quadratino circostante il punto. E' chiaro che la qualità dell'immagine riprodotta a partire da questa codifica, dipenderà da due fattori cruciali: la densità della griglia su cui vengono acquisiti i campioni (risoluzione spaziale) ed il numero di bit che si usano per

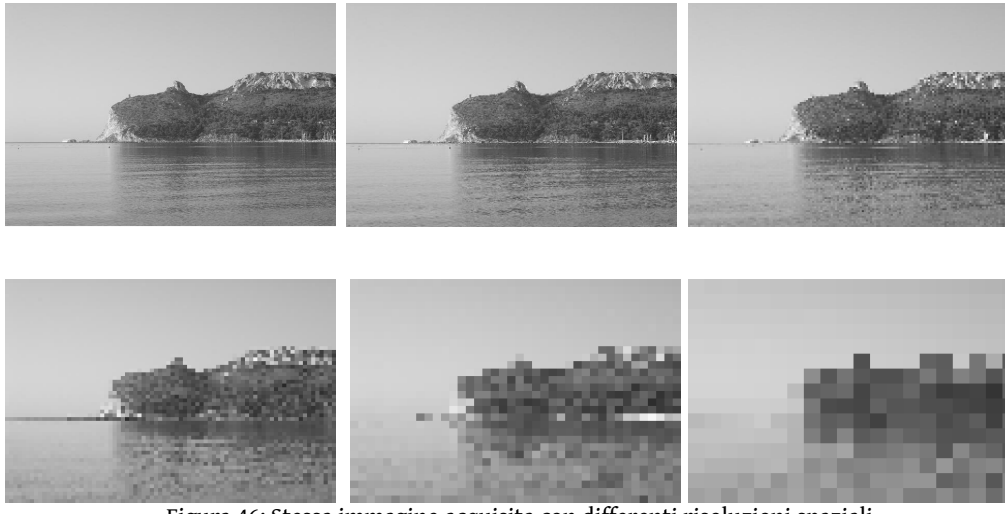


Figura 46: Stessa immagine acquisita con differenti risoluzioni spaziali

memorizzare i colori (range dinamico) che determinerà l'accuratezza con cui si potranno rappresentare le sfumature. Quando si compra una macchina fotografica digitale, si indica in genere la risoluzione in Megapixel (milioni di punti), che è il numero che esprime il prodotto delle dimensioni (NxM) della griglia di acquisizione: tanto più sarà alto tale numero, quanto più l'immagine apparirà dettagliata, mentre se è basso, apparirà evidente la granularità dell'acquisizione (vedi Figura 46). Ovviamente la percezione degli effetti della risoluzione limitata dipenderà dall'ingrandimento con cui viene rappresentata, cioè dalla dimensione fisica dei quadratini (pixel) che compongono la rappresentazione. Per questo motivo le foto digitali, anche se oggi i sensori hanno raggiunto risoluzioni molto elevate, non possono essere ingrandite eccessivamente senza mostrare i limiti del dettaglio acquisito. Il numero di bit con cui si acquisiscono la luminosità o le componenti di colore è invece, in generale, poco variabile: tipicamente le immagini digitali usano 8 bit per pixel per ciascuna componente di colore o per la sola luminosità se a toni di grigio. Solo tipi particolari di immagini, come, ad esempio, quelle diagnostiche, usano un numero di bit più elevato. Gli 8 bit sono sufficienti in quanto l'occhio umano non è in grado di distinguere più di un centinaio di differenti livelli di luminosità (o livelli di grigio). Otto bit codificano 256 livelli differenti e quindi sono sufficienti. Se si scende sotto tale soglia, tuttavia, la perdita di qualità diventa via via rilevante e compaiono effetti fastidiosi come i falsi contorni (“posterizzazione”) evidenziati nella Figura 47.

Se nelle immagini in bianco e nero ad ogni pixel corrisponde quindi un numero da 0 a 255, in quelle a colori vengono codificate 3 componenti corrispondenti ai colori primari (si usano tipicamente rosso verde e blu, quindi si parla di immagini RGB, dai nomi dei colori in inglese: red, green e blue), ciascuna con un byte. Ogni pixel occupa quindi 3 byte, in realtà spesso si usa una codifica da 4 byte con un canale aggiuntivo detto alfa, usato per rappresentare informazioni aggiuntive (ad esempio la trasparenza). Con gli otto bit per colore si possono generare oltre 16 milioni di differenti sfumature e non avrebbe quindi senso, dati i limiti della percezione umana, andare oltre, tant'è che si parla, in questo caso, di immagini “true color” cioè in cui la rappresentazione del colore è realistica (ovviamente nei limiti imposti dalla rappresentazione additiva del colore mediante componenti RGB).

La nota dolente della rappresentazione ottenuta in questo modo per le immagini digitali sul

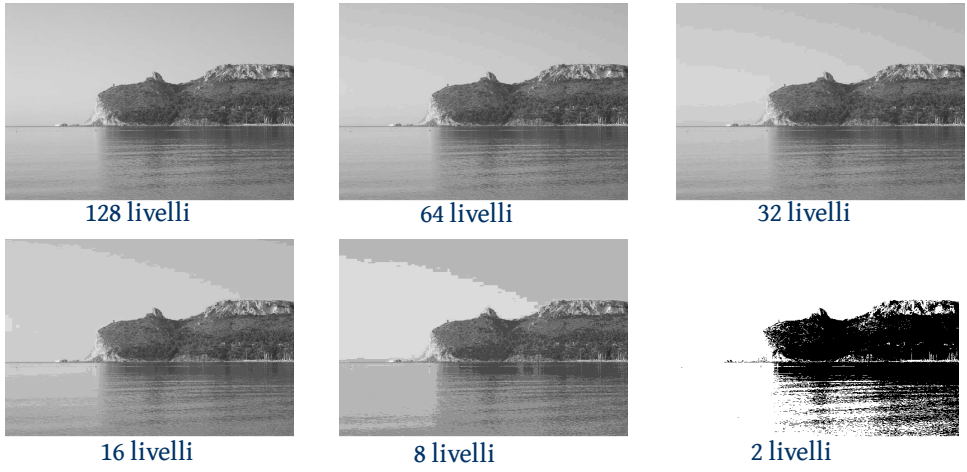


Figura 47: Stessa immagine con range dinamico (numero di livelli di grigio) variato. Si noti la comparsa dei cosiddetti "falsi contorni" (posterizzazione).

calcolatore riguarda naturalmente la quantità di memoria richiesta, che ne renderebbe problematica la memorizzazione sui vari supporti e la trasmissione.

Una foto acquisita con una fotocamera da 8 megapixel, ad esempio, occuperà, se codificata con 4 byte per pixel, ben 32 Megabyte. Questo vuol dire che, per trasferirla con un collegamento veloce a 4 Megabit/s occorrerebbe comunque più di un minuto!

Per questo motivo sono stati via via adottati vari metodi per codificare in maniera più compatta le immagini. Si parla in tal caso di tecniche di **compressione**, in quanto appunto comprimono lo spazio occupato dai file immagine in memoria. Tutti i principali formati di immagini che usiamo comunemente sfruttano adeguatamente queste tecniche.

Le tecniche di compressione possono essere di due tipi: **lossless**, cioè senza perdita, tali che il file contenga comunque la stessa informazione utile (e permetta quindi di ricostruire esattamente lo stesso dato di partenza), o **lossy**, in cui parte dell'informazione utile sia irrimediabilmente perduta nell'operazione.

Per dare un idea di come possano funzionare tecniche di compressione lossless, possiamo accennare a due algoritmi molto usati: il Run Length Encoding (RLE), che sfrutta l'idea di codificare sequenze di dati con lo stesso valore mediante il valore stesso e la lunghezza della sequenza, oppure la codifica non uniforme, che usa un codice con numero di bit ridotto per i valori più frequenti nei dati. Naturalmente questi metodi possono ottenere buoni risultati soltanto per tipi di immagini piuttosto semplici e regolari.

I metodi lossy, invece, cercano di eliminare quella parte dell'informazione che non è rilevante ai fini della percezione umana, in modo tale che l'utente non si renda conto della differenza. Un modo semplice può essere, ad esempio, quello di diminuire lo spazio occupato dalla codifica del colore. Un'idea spesso utilizzata è, anziché memorizzare per ogni pixel la terna RGB, salvare per ogni pixel un indice che corrisponde a un colore specificato su una tavolozza (colormap). Se l'immagine ha un numero limitato di colori, questo consente di ridurre anche notevolmente lo spazio necessario per la memorizzazione dell'immagine. Riducendo il numero di colori della tavolozza si otterranno via via versioni sempre più "compresse" anche se qualitativamente deteriorate delle immagini.

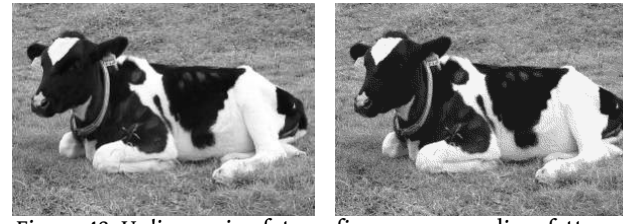


Figura 48: Un'immagine fotografica compressa di un fattore 5 con jpeg (a sinistra) è indistinguibile dall'originale, mentre per ottenere la stessa compressione con gif la riduzione del numero di colori crea deterioramento evidente (a destra)

Il formato di immagini **GIF** (Graphics Interchange Format, estensione dei file .gif), uno di quelli supportati sul web, usa una rappresentazione con tavolozza dei colori (fino a 256). Questo fa si che sia molto adatto a memorizzare e codificare immagini che non hanno sfumature continue, come disegni o scene con grandi regioni uniformi. La conversione di un immagine true color in una codificata mediante tavolozza di colori si deve fare approssimando ciascun colore in quello "più simile" disponibile nella tavolozza stessa. A volte per ovviare al numero limitato di colori si applica il cosiddetto "dithering", operazione attraverso la quale si creano sfumature accostando i colori che le creerebbero per addizione in pixel vicini. Se le sfumature nell'immagine sono molte e continue la perdita di qualità è evidente.

Per comprimere immagini di tipo fotografico si usano, quindi, con maggiore efficacia algoritmi che decompongano il segnale in frequenza ed eliminino dall'immagine le frequenze meno rappresentate. Ad esempio il formato **JPEG** (da Joint Photographic Experts Group, estensione .jpg), anch'esso supportato dal web, cioè visualizzato dai browser, comprime le immagini suddividendole in piccoli quadrati e memorizzando in ciascuno di essi solo le componenti principali in frequenza. In questo modo si ottiene una codifica molto efficace per le foto (fattori di compressione anche dell'ordine del centinaio con immagini ancora perfettamente fruibili, come in Figura 48). Il metodo ha invece problemi nel caso di immagini con contorni netti e colori continui in quanto produce artefatti (si vedono cioè strutture non presenti nel segnale originario, come in Figura 49).

I formati di file per salvare le immagini sono numerosissimi: vari produttori di sistemi operativi o di sistemi di acquisizione di immagini ne hanno introdotti di propri. Ogni formato definisce le modalità di salvataggio dei valori dei pixel, l'uso di algoritmi di compressione, e può aggiungere altre caratteristiche e funzionalità, come ad esempio la possibilità di aggiungere la gestione della trasparenza (attraverso il canale alfa o nel caso di tavolozza, mediante un indice particolare) o la possibilità di essere trasmesse in rete in modalità progressiva: nei formati progressivi i file sono organizzati in modo che i primi byte contengano versioni a più bassa risoluzione dell'immagine ed i successivi contengano correzioni successive: in tal modo, quando trasmesse in rete, i browser possono iniziare a visualizzare le versioni a bassa qualità durante lo scaricamento. Riportiamo di seguito alcune caratteristiche dei formati di file immagine più popolari:

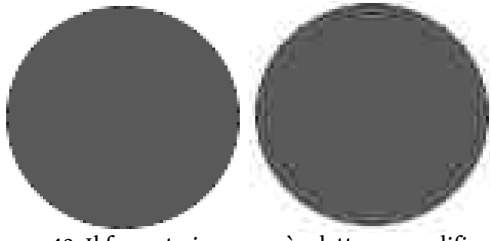


Figura 49: Il formato jpeg non è adatto per codificare disegni con aree di colore costante. A sinistra, immagine 80x80 di circonferenza grigia su sfondo bianco, compressa nel formato gif a 0.7 Kb. A destra stessa compressione ottenuta con l'algoritmo jpeg: sono evidenti gli artefatti ai bordi.

Windows Bitmap (estensione .bmp) era il formato supportato da Microsoft Windows, supporta diverse profondità in bit e tavolozze, ma supporta solo la compressione run length encoding.

Tagged Image File Format (.tiff) è un formato proprietario (oggi detenuto da Adobe) che ha avuto discreto successo, presenta la proprietà di supportare bitmap non compresse o compresse con vari algoritmi e di aggiungere informazione ausiliaria (ad esempio relativa alla calibrazione del colore). Grazie a questa caratteristica, le specifiche TIFF sono state usate anche per sviluppare un noto formato per immagini georeferenziate (geotiff), che fa corrispondere ai pixel un esatto riferimento geografico.

Graphics Interchange Format (.gif) è un formato che ha avuto molto successo sul web, esso codifica come abbiamo detto con tavolozza di colori limitata a 256 valori ed usa una tecnica di compressione lossless detta LZW molto efficiente, ma su cui ci sono stati problemi di diritti di autore, che hanno fatto sì che si sia poi spinto sullo sviluppo del formato png, libero da brevetti, per sostituirlo. Il brevetto sul formato gif è comunque oggi scaduto, per cui lo si può usare liberamente. GIF supporta anche la trasparenza e l'interlacciamento (cioè un formato progressivo in cui prima vengono memorizzate/trasmesse le linee pari e poi le dispari) e consente di salvare anche animazioni.

Joint Photographic Experts Group (.jpg) è lo standard di compressione delle immagini fotografiche più utilizzato. A differenza di GIF è un formato gratuito ed open source. Come detto esso supporta una compressione lossy basata su una trasformata in frequenza (Discrete Cosine Transform), che permette grossi fattori di compressione mantenendo un'apparenza abbastanza buona, seppure creando artefatti evidenti. Recentemente il comitato JPEG ha sviluppato un formato qualitativamente migliore (JPEG 2000) destinato a sostituire quello originale.

Portable Network Graphics (.png) è stato come detto sviluppato per sostituire il formato GIF, non libero. Per questo ha caratteristiche comuni a GIF: gestisce la trasparenza, le immagini indicizzate con colormap, l'interlacciamento, la compressione lossless ed è ottimale per immagini con colori non sfumati. Essendo più recente, ha però molti meno limiti del formato GIF: supporta anche le immagini true color, può supportare anche l'uso di 16 bit per la codifica del grigio o delle componenti di colore e supporta la memorizzazione di informazione ausiliaria. I comuni programmi di elaborazione immagini sono in genere in leggere e scrivere tutti i formati, si ricordi però che invece i **browser web** supportano solamente la visualizzazione dei file **GIF**, **JPEG** e **PNG**. Un'ultima cosa da ricordare è che le caratteristiche dell'immagine memorizzata non corrispondono necessariamente a quelle della rappresentazione poi generata sullo schermo del calcolatore. Il range dinamico della visualizzazione per l'utente dipenderà anche dal sistema operativo e dalle opzioni del programma utilizzato (e, chiaramente, dipenderà anche dalla scheda grafica del calcolatore e dalle caratteristiche del monitor). Agli albori del web, dato che PC e browser dati i limiti delle schede grafiche non erano in effetti in grado di rappresentare immagini true color, si creavano per il web immagini con

Figura 50: La visualizzazione dei colori sul proprio schermo (e anche la risoluzione in pixel) sono gestite dalle opzioni del sistema operativo e della scheda grafica del PC (qui si vede il pannello di controllo per schede nVIDIA su sistema operativo Linux)

tavolozza di colori ottimale per rappresentare al meglio le immagini. E' sempre meglio essere consapevoli dei limiti eventualmente posti dall'hardware grafico alla visualizzazione (basta verificare la configurazione del sistema, come nell'esempio in Figura 50).

Anche altre caratteristiche dell'hardware e del software dei nostri computer limitano la rappresentabilità delle immagini. Dimensioni di finestre e schermo vincolano infatti la risoluzione realmente rappresentata (per cui non sarà affatto utile trasmettere per la visualizzazione su una pagina web un'immagine da 4 megapixel dato che un intero schermo tipicamente rappresenta meno di 1 milione di pixel).

Dimensioni e qualità delle immagini da inserire sulle pagine web dovranno essere sempre sufficienti per generare l'effetto di visualizzazione desiderato e nulla più. Le dimensioni dovranno coincidere possibilmente con quelle della rappresentazione sullo schermo: un dettaglio maggiore comunque non sarebbe percepito dagli utenti.

In questo caso però dovremo sempre ricordare che se andremo ad ingrandire la rappresentazione di un'immagine composta da pochi pixel per visualizzarla su tutto lo schermo non recupereremo un'immagine con elevato dettaglio, ma un'immagine limitata dalla bassa risoluzione del dato. Questa dovrebbe mostrare la tipica quadrettatura dovuta ai pixel ingranditi, effetto a volte ridotto grazie all'uso di interpolazione (si vede in tal caso però l'immagine molto sfumata, vedi Figura 51).

Si noti anche che quando ingrandiamo un immagine sullo schermo di un PC con le normali modalità consentite dai sistemi operativi (ad esempio zoomando la vista del word processor o dell'editor di immagini) i software dovranno mappare i pixel originali in quelli del display e le dimensioni di visualizzazione non sono necessariamente multiple delle dimensioni delle immagini. In tal caso la visualizzazione ingrandita presenta fastidiosi artefatti causati dal cosiddetto **aliasing**, cioè si vedono dettagli non naturali dovuti ad errata ricostruzione del segnale ad alta frequenza. Questo effetto può essere ridotto con l'uso dell'**interpolazione** (cioè si ricava il valore del colore sui punti dello schermo in funzione di un certo numero di campioni del dato originale anziché di uno solo) o applicando filtri per eliminare le alte frequenze spaziali del segnale (**anti aliasing**).

Figura 51: Effetto della visualizzazione ingrandita di un'immagine bitmap codificata con una risoluzione molto bassa (a sinistra): appare evidente la forma quadrata del pixel stesso (al centro). L'effetto è in genere ridotto nei programmi di visualizzazione da tecniche di interpolazione, che calcolano i pixel dell'ingrandimento come funzione di più pixel vicini dell'immagine originale e che rendono però l'immagine sfumata (a destra).

5.3 Immagini e grafica vettoriali

E' possibile memorizzare e visualizzare immagini sul calcolatore anche in maniera completamente differente, utilizzando la cosiddetta grafica "**vettoriale**". I formati di immagini vettoriali rappresentano l'immagine sostanzialmente mediante istruzioni di disegno: memorizzano infatti una serie di forme base rappresentabili mediante parametri (coordinate di punti, dimensioni, colori) che vengono composte per creare le immagini. Il principale vantaggio

di questa rappresentazione sta nel fatto che se si decide di ingrandire o ridurre le dimensioni della rappresentazione, il dettaglio dei contorni sarà sempre il massimo possibile data la risoluzione della finestra di visualizzazione e non dipenderà invece dal file come nel caso delle immagini bitmap. Le immagini vettoriali date le loro caratteristiche, sono molto indicate per codificare disegni, schemi, testo, mentre non sono molto indicate per codificare fotografie in cui vi sono molte variazioni locali di sfumatura senza contorni ben definiti.

Figura 52: Differenza tra grafica bitmap e grafica vettoriale: se quando l'immagine è piccola la rappresentazione è uguale, dopo l'ingrandimento la prima (a sinistra) presenta la caratteristica quadrettatura, la seconda ha un contorno definito finemente.

I formati di descrizione dei documenti utilizzati per la stampa (es. pdf, postscript) codificano generalmente la grafica ed il testo proprio in questo modo in modo da poter sfruttare la maggiore risoluzione che si ha su carta rispetto allo schermo.

Anche i formati proprietari che permettono di codificare disegni animati, ad esempio Adobe Flash, gestiscono grafica vettoriale. Il W3C ha definito per il web un formato standard aperto (cioè accessibile ed utilizzabile da tutti) per le immagini vettoriali detto Scalable Vector Graphics (**SVG**). Derivato dall'XML, è un linguaggio che descrive le figure bidimensionali statiche e animate attraverso un codice testuale che permette di definire tre tipi di oggetti grafici: forme geometriche, cioè linee costituite da segmenti di retta e curve e aree delimitate da linee chiuse; immagini raster (cioè si può includere un'immagine di tipo bitmap all'interno del disegno) e testi esplicativi, eventualmente cliccabili.

5.4 Elaborazione delle immagini

Una delle operazioni che anche i non addetti ai lavoro effettuano spesso sulle immagini digitali è la loro elaborazione, cioè la modifica delle stesse ai fini di creare nuove immagini più adatte alla stampa o all'utilizzo nei vari tipi di documento multimediale.

Le immagini digitali che utilizziamo, sono in genere create da dispositivi come fotocamere, webcam, telefonini oppure scaricate dai numerosi siti web che contengono archivi di vario tipo. Esse, però, non sono in genere adatte ad essere inserite direttamente nelle nostre pagine web, o nei documenti creati coi word processor, in quanto devono essere in genere adattate, ritagliate, migliorate nel contrasto e, per quanto riguarda il web, dovrebbe essere il più possibile ridotta la loro occupazione di memoria per facilitarne il trasferimento.

Esistono come vedremo molti programmi anche gratuiti per la creazione ed elaborazione di immagini ed il cosiddetto “fotoritocco”, ma, per poterne usufruire in maniera ottimale, è utile avere un po' di informazioni di base su cosa si intenda per “**elaborazione delle immagini digitali**” e quali siano le problematiche di questa disciplina.

L'elaborazione di immagini bitmap trasforma la matrice originale dei pixel in un'altra, più adatta alla visualizzazione dell'informazione utile o semplicemente più gradevole.

Se escludiamo il ritocco manuale mediante disegno, le tipiche operazioni di modifica delle immagini che ci interessano sono sostanzialmente di tre tipi:

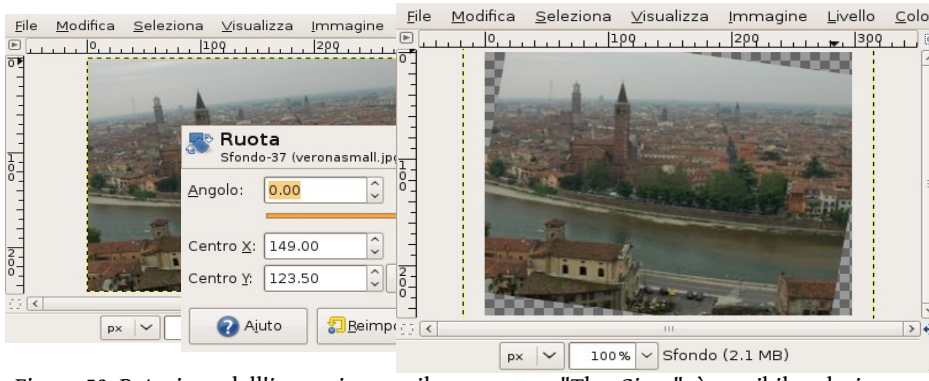


Figura 53: Rotazione dell'immagine con il programma "The Gimp": è possibile selezionare centro e un angolo arbitrario, i pixel sulla nuova griglia vengono calcolati mediante interpolazione con algoritmo a scelta dell'utente.

- trasformazioni **geometriche:** ad esempio per ingrandire, ridurre, ruotare le immagini o correggere le deformazioni prospettiche;
- trasformazioni della **colormap**: per cambiare il colore dei pixel mediante una funzione che faccia corrispondere ad ogni colore dell'immagine di origine, un colore differente nell'immagine di destinazione;
- trasformazioni **locali**: che ricavano nuovi valori dei pixel dai valori di una regione circostante mediante un opportuna regola utile, per esempio, a ridurre il rumore o esaltare i contorni.

Faremo ora cenno a come e perché possano essere utili queste trasformazioni, mostrando anche esempi di applicazione. L'uso di queste trasformazioni è possibile in modo semplice attraverso numerosi programmi per l'elaborazione di immagini disponibili sul mercato, tra cui ottime soluzioni costituite da software libero (vedi Appendice 1), scaricabili gratuitamente dalla rete ed in grado di sostituire egregiamente costose soluzioni proprietarie. Nell'ambito dell'elaborazione fotografica degno di menzione è sicuramente il programma "**The Gimp**", software open source disponibile per tutte le piattaforme ed in grado di applicare le più comuni es utili trasformazioni alle immagini. Esso è stato utilizzato per realizzare tutti gli esempi a seguire. Per scaricarlo o accedere alla documentazione per avere maggiori dettagli basta visitare il sito *http://www.gimp.org*.

Le **trasformazioni geometriche** sono sicuramente utili allo scopo di creare le immagini da inserire nelle pagine web a partire dal dato fotografico. Nella maggioranza dei casi, infatti, non è consigliabile inserire le immagini così come vengono create dalle fotocamere in quanto presumibilmente conviene ridurne le dimensioni (scaling), ritagliarne la

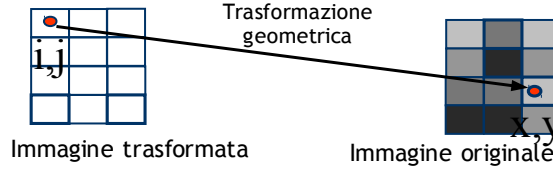


Figura 54: Trasformazioni geometriche ed interpolazione: dato che i centri dei pixel dell'immagine trasformata (es. i,j a sinistra) non corrispondono esattamente ai centri di quelli dell'immagine originale (es. x,y, a destra) trasformati, i valori ad essi assegnati vengono calcolati mediando con pesi opportuni (interpolando) i valori dei centri più vicini.

regione di interesse (cropping), ruotarla per ottenere il giusto orientamento e così via.

In Figura 53 si vede l'esempio di una procedura di rotazione (ma considerazioni analoghe varrebbero per traslazione e ingrandimento): il programma, data la griglia originale dei pixel e i parametri della trasformazione, calcola i valori della nuova immagine che viene in questo caso codificata inserendo i nuovi valori nella griglia di pixel originale. Quest'ultimo passo è in realtà l'unico vero punto critico nelle procedure di trasformazione geometrica delle immagini. Le trasformazioni, infatti non potranno in genere spostare i punti della griglia di campionamento dell'immagine in punti della stessa griglia. Quello che, quindi, fanno i programmi è calcolare i valori del colore sui punti della griglia rettangolare dell'immagine di destinazione sulla base di quelli più vicini al punto corrispondente nell'immagine originale. Questa procedura viene detta **interpolazione**. Se il punto della nuova griglia non corrisponde a una regione interna all'immagine originale, i colori dei pixel restano non definiti, come si fede nelle regioni di bordo dell'immagine di destra nell'esempio di Figura 53. Per comprendere meglio questo passo dell'interpolazione, osserviamo la Figura 54: se si va a cercare quale punto dell'immagine originale corrisponda ad una generica trasformazione che porti alle coordinate intere di un pixel della nuova immagine (i,j), si trova in genere un valore non intero, per cui calcolerò il colore sulla base di una media pesata dei valori dei pixel più vicini a tale valore (interpolazione).

Se si vanno a vedere le opzioni dei comandi di trasformazione geometrica dei programmi di fotoritocco come The Gimp, si trova spesso la possibilità di scegliere l'algoritmo di interpolazione utilizzato: il peggiore usualmente si limita a copiare il valore più vicino (nearest neighbor), mentre altri schemi ottengono risultati qualitativamente migliori (es. interpolazione bicubica).

Una trasformazione geometrica speciale e molto utile per l'elaborazione fotografica è la correzione della distorsione dell'obiettivo, in genere realizzata mediante una trasformazione parametrica piuttosto complessa, ma che, per esempio, in The Gimp può essere realizzata in modo relativamente semplice selezionando l'apposita voce di menù, modificando i parametri e vedendo istantaneamente un'anteprima dell'effetto ottenuto.

Le **trasformazioni globali** o della colormap sono particolarmente importanti per l'elaborazione fotografica. Esse sono generalmente applicate alle immagini true color per migliorarne la qualità visiva. Molte immagini, infatti, non sono efficaci a causa dello scarso contrasto o della scarsa luminosità. Esse, quindi, non sfruttano adeguatamente il range dinamico: nei pixel dell'immagine è rappresentato solo un sottoinsieme dei valori disponibili per la codifica.

Per chiarire il concetto, consideriamo un'immagine in bianco e nero o meglio in scala di grigi (grayscale): tipicamente in essa i colori vengono rappresentati con 8 bit, cioè con valori che

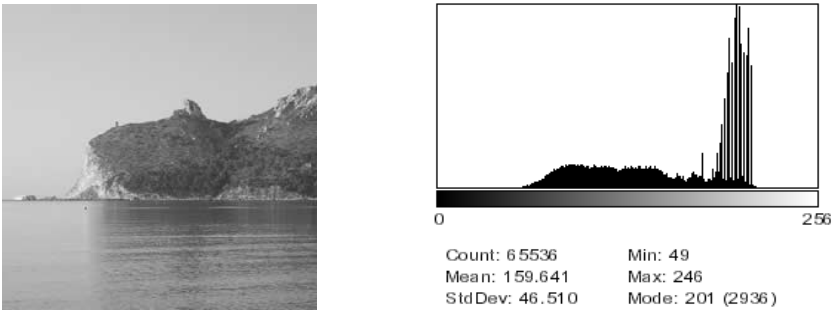


Figura 55: Esempio di istogramma di un'immagine in scala di grigi: non tutti i livelli sono utilizzati.

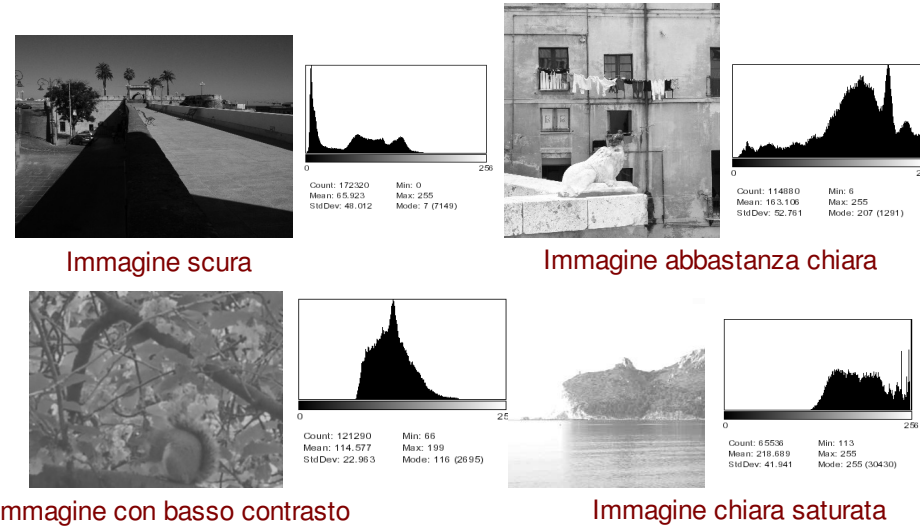


Figura 56: L'istogramma appare molto diverso in caso di immagini chiare, scure, poco o molto contrastate.

vanno da 0 (nero) a 255 (bianco). Per capire come l'immagine sfrutti il range dinamico possiamo osservarne l'**istogramma**: cioè la funzione che rappresenta l'occorrenza dei valori di grigio (per le immagini a colori si possono osservare gli stessi grafici per le varie componenti di colore o per la luminosità). In Figura 55 vediamo un esempio: in quasi tutti i programmi di elaborazione immagini gli istogrammi sono rappresentati con un grafico a barre come quello ivi illustrato. Dall'istogramma possiamo capire se il range dinamico è ben sfruttato, se l'immagine ha un buon contrasto, se è molto chiara o molto scura ed eventualmente applicare le opportune trasformazioni del colore utili a migliorarne la qualità percepita. In Figura 56 sono visibili alcune immagini di differente qualità e gli istogrammi associati. Per rendere visivamente più efficaci le immagini si può quindi operare una trasformazione del colore. In genere la trasformazione sarà una funzione che mapperà il livello originale in un nuovo valore. Ad esempio se la funzione mappa linearmente tra il minimo e il massimo valore rappresentato nell'istogramma ottengo una funzione che fa il cosiddetto “stretching” del contrasto, ottenendo un'immagine molto più contrastata e gradevole (Figura 57). I programmi di fotoritocco implementano anche un metodo di esaltazione del contrasto detto histogram equalization che calcola la funzione di trasformazione del colore ottimale per cercare di sfruttare al massimo il range dinamico. Le trasformazioni della colormap possono poi essere del tutto arbitrarie. Ad esempio è usuale in medicina, ma anche nella visualizzazione scientifica, rappresentare le immagini con i cosiddetti “falsi colori”, cioè facendo corrispondere a determinati intervalli di grigio delle tonalità di colore differenti per evidenziare determinate strutture. I programmi di grafica offrono ampia libertà di applicare colormap predefinite, molte delle quali sono tipiche per alcune applicazioni (ad esempio un campo di temperature sarà ben rappresentato con i valori alti in tinte tendenti al rosso e quelli bassi in tinte tendenti all'azzurro). In generale le trasformazioni globali si applicheranno anche alle immagini già in origine a colori. Qui ovviamente la cosa è un po' più complicata, visto che il dato non ha un'unica componente, ma tre differenti (rosso, verde e blu). Possiamo analizzare l'istogramma e modificare singolarmente ciascuna componente, ma questo significherebbe alterare la tinta dei pixel.

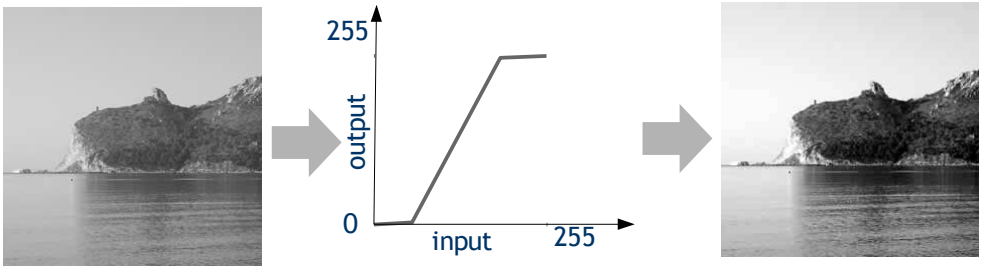


Figura 57: Applicando una funzione di contrast stretching lineare si può rendere un'immagine poco contrastata più gradevole ed efficace

Per questo spesso si sostituisce la rappresentazione del colore RGB con un'altra in cui una delle tre componenti indipendenti rappresenti il valore di luminosità del pixel, che può così essere elaborato in modo indipendente, rendendo più chiara o più scura l'immagine senza modificare la tinta. Una rappresentazione di questo tipo è quella nota come HSV. In essa le componenti che codificano il colore sono la tinta (Hue), la saturazione del colore (Saturation, cioè quanto grigio è mischiato alla tinta pura) e la luminosità (Value).

Si parla di rappresentazioni di colore o spazi di colore, in quanto possiamo considerare le componenti RGB come le componenti in tre dimensioni perpendicolari di uno spazio e le rappresentazioni alternative come cambi di sistema di riferimento in questo spazio. I programmi di fotoritocco forniscono interfacce facilitate per generare trasformazioni utili sulle componenti di colore. Ad esempio, il programma The Gimp permette di agire in modo semplificato (ad es. mediante controlli di luminosità o contrasto) o di definire a mano le funzioni di trasformazioni per le componenti di luminosità o colore (Figura 58).

Le trasformazioni "locali" sono infine quelle che si ottengono applicando quelli che spesso vengono chiamati **filtri digitali** alle immagini. In esse il nuovo valore del colore di un pixel è funzione dei valori di un intorno di punti vicini. Si parla di filtri lineari o di convoluzione

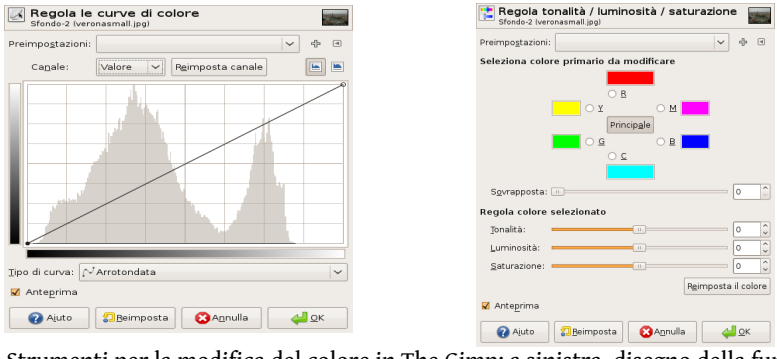


Figura 58: Strumenti per la modifica del colore in The Gimp: a sinistra, disegno della funzione di trasformazione per la componente selezionata (V, R, G o B) a destra funzioni preimpostate di miglioramento contrasto, luminosità.

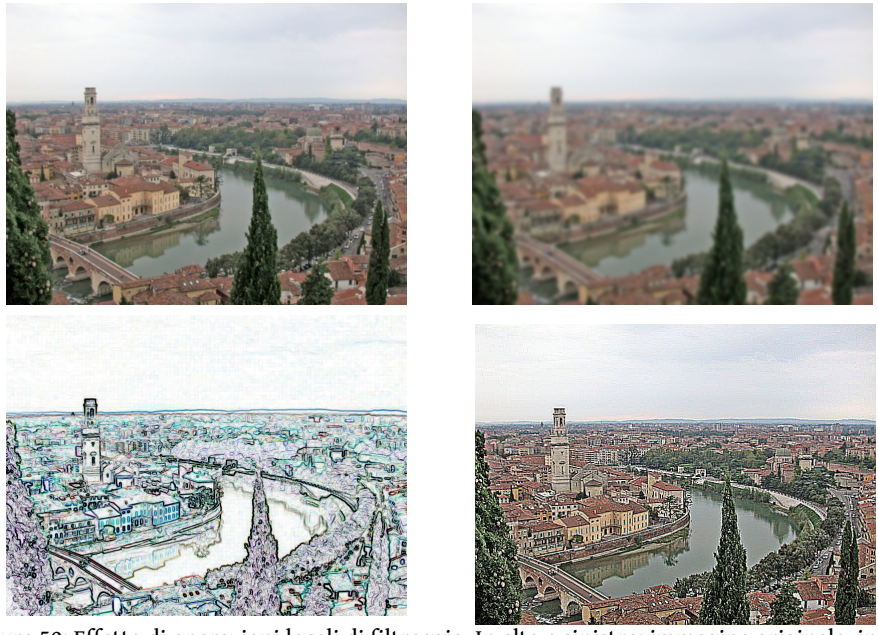


Figura 59: Effetto di operazioni locali di filtraggio. In alto a sinistra: immagine originale, in alto a destra: sfocatura Gaussiana, in basso a sinistra: rilevamento contorni, in basso a destra: affilatura.

quando la funzione è semplicemente una somma con coefficienti dei valori del vicinato. Una tipica applicazione di questi filtri è per la riduzione del rumore. Le immagini acquisite da fotocamere e telecamere possono essere affette da rumore, cioè i valori acquisiti sui pixel non corrispondono alla luminosità esatta proveniente dalla scena in quanto una componente di disturbo (rumore) si somma ad essa. Spesso il rumore può essere eliminato con un'operazione di media locale, quella che nei programmi appare come opzione di "blurring" o sfocatura. Essa viene effettuata mediando appunto i valori dei pixel all'interno di una "maschera" più o meno grande. L'algoritmo più usato (sfocatura Gaussiana) utilizza nella media dei pesi proporzionali ad una funzione che ha coefficienti più alti al centro e più bassi lontano dal pixel centrale,

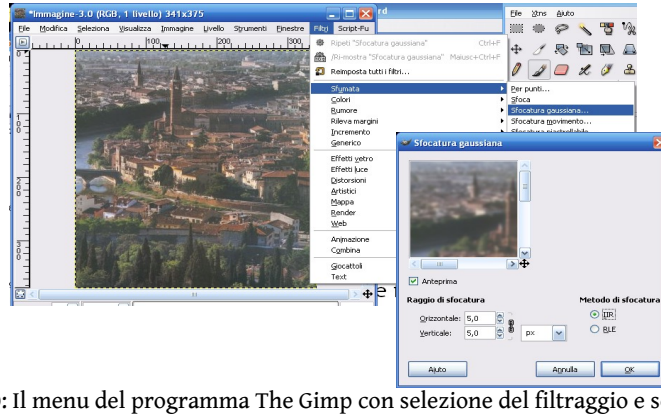


Figura 60: Il menu del programma The Gimp con selezione del filtraggio e scelta sfocatura Gaussiana, in cui si può scegliere la dimensione della maschera e vedere l'anteprima del risultato

descritta da un unico parametro (sigma) che ne determina l'ampiezza del picco (più è elevato, più l'effetto di sfocatura è rilevante). Chiaramente la sfocatura è efficace nel ridurre il rumore, ma peggiora la qualità dei contorni. Esistono comunque altri filtri che cercano di esaltare i contorni stessi (image sharpening o affilatura). Essi in genere sono basati su altri algoritmi che invece esaltano nell'immagine le sole regioni ove vi sono contorni (edge detection). Esistono anche filtri in grado di ridurre il rumore senza sfocare i contorni degli oggetto (filtri non lineari, mediana, ecc.). In Figura 59 si vede l'effetto di alcuni filtri su un'immagine. In Figura 60 si vedono le opzioni del programma per la sfocatura Gaussiana: come per altri filtri, si possono regolare dei parametri, in questo caso sostanzialmente la dimensione dell'area che viene sfumata intorno a ciascun pixel: più è grande più saranno rilevanti sia la sfocatura, sia la soppressione del rumore.

Tra le modifiche dell'immagine che possono essere utili per creare immagini ottimali per le nostre applicazioni ci sono poi ovviamente gli strumenti di disegno, scrittura, ecc. Anche “The Gimp” offre numerosi strumenti di questo tipo (uso di penne, pennelli, spray con diverso spessore o trasparenza, inserimento di testo con varie scelte di caratteri, colori, ombre, effetti, ecc.), tuttavia, dato che il loro uso non ha a che fare con nozioni tecniche informatiche, non ce ne occuperemo qui, rimandando gli utenti alla manualistica dei programmi od ai numerosi tutorial (corsi pratici) presenti in rete.

Ultima nota: anche per quanto riguarda la grafica vettoriale, esistono naturalmente programmi per creare immagini e salvarle in formati adatti come .svg. Anche per l'uso di questi strumenti, non occorrono particolari nozioni tecniche, per cui ci limitiamo qui a segnalare la presenza di simili programmi di software libero, come Inkscape.

5.5 Audio digitale

I segnali acustici sono onde meccaniche trasportate dall'aria percepite dalle nostre orecchie entro l'intervallo di frequenza da circa 20 a circa 20000 Hertz.

La qualità del suono udito è determinata dall'intensità o ampiezza dell'onda e da come le frequenze del segnale sono composte tra loro (il che determina il cosiddetto “timbro” del suono).

L'**intensità** corrisponde a quello che chiamiamo il volume del suono e misura l'**energia** associata all'onda. Si misura in una scala logaritmica (in cui cioè ogni incremento di unità equivale ad una moltiplicazione per 10 del segnale) utilizzando un'unità di misura, il **decibel (dB)** che si riferisce alla percezione umana: l'uomo percepisce i suoni che vanno da 0 a 140 dB, sopra un certo livello di intensità i suoni, come noto, determinano rischi di danni all'apparato uditivo. Oltre i 130 dB si percepisce anche dolore.

La **frequenza** delle onde sonore, misurata in **Hertz** cioè cicli al secondo, determina la nota percepita: intuitivamente le frequenze alte corrispondono ai suoni che chiamiamo acuti, quelli bassi a quelli detti gravi. In genere si dividono gli intervalli di frequenze in ottave, ad ogni ottava la frequenza raddoppia. L'analisi in frequenza dei segnali si fa rappresentando il segnale (nel nostro caso l'energia della vibrazione sonora) come composizione di segnali puri sinusoidali, cioè oscillazioni periodiche con determinata frequenza. Un'onda di questo tipo viene da noi percepita come una nota costante (come, ad esempio, il la di uno strumento per l'accordatura dei pianoforti). In realtà ogni suono o rumore che sentiamo non è in genere una pura onda sinusoidale, ma una composizione di onde con determinate frequenze e ampiezze.

Gli strumenti musicali emettono in genere suoni caratterizzati da una frequenza base più una

composizione di multipli di tale frequenza (armoniche), la forma dell'onda risultante è quella che caratterizza il **timbro** del suono dello strumento. Nei rumori invece la composizione delle frequenze appare casuale (vedi Figura 61).

Ogni strumento musicale così come ogni voce umana è caratterizzata da un particolare timbro (si parla infatti di timbro delle voci) che può essere analizzato studiando la forma d'onda.

Per acquisire il suono come dato digitale, i microfoni traducono l'oscillazione in segnale elettrico che viene campionato a intervalli di tempo regolari (passo di campionamento, il cui inverso è la **frequenza di campionamento**) e binarizzato convertendo la misura in un numero espresso da un numero limitato di bit (conversione analogico/digitale).

La codifica ottenuta si dice PCM (Pulse Code Modulation) e la qualità del dato dipende chiaramente dalla frequenza scelta e dal numero di bit usati.

Se il numero di segnali misurati è abbastanza fitto, si può poi ricostruire e riprodurre il suono in maniera fedele. Esiste un teorema dell'analisi dei segnali (detto di Nyquist/Shannon o teorema del campionamento) che dice che se si campiona a frequenza doppia della massima frequenza presente nel segnale, si può poi ricostruire il segnale originale senza errore (trascurando ovviamente l'errore di quantizzazione). Questo significa che, utilizzando una frequenza di campionamento superiore ai 40 Khz, si possono riprodurre fedelmente tutte le frequenze udibili dall'orecchio umano. Per questo motivo il campionamento del suono nei comuni CD è realizzato a 44.1 Khz.

Questo fa sì che la quantità di memoria occupata dalla codifica sonora sia però enorme. Facendo un rapido calcolo, se utilizziamo 16 bit per codificare l'intensità (range dinamico accettabile per la qualità CD) occorrono 88 Kbyte per un solo secondo di registrazione. Se pensiamo che il suono è in genere codificato in stereo (2 canali per rendere la spazialità del suono) il numero va poi raddoppiato. Una canzone di 4 minuti occuperà quindi 88x2x4x60=42

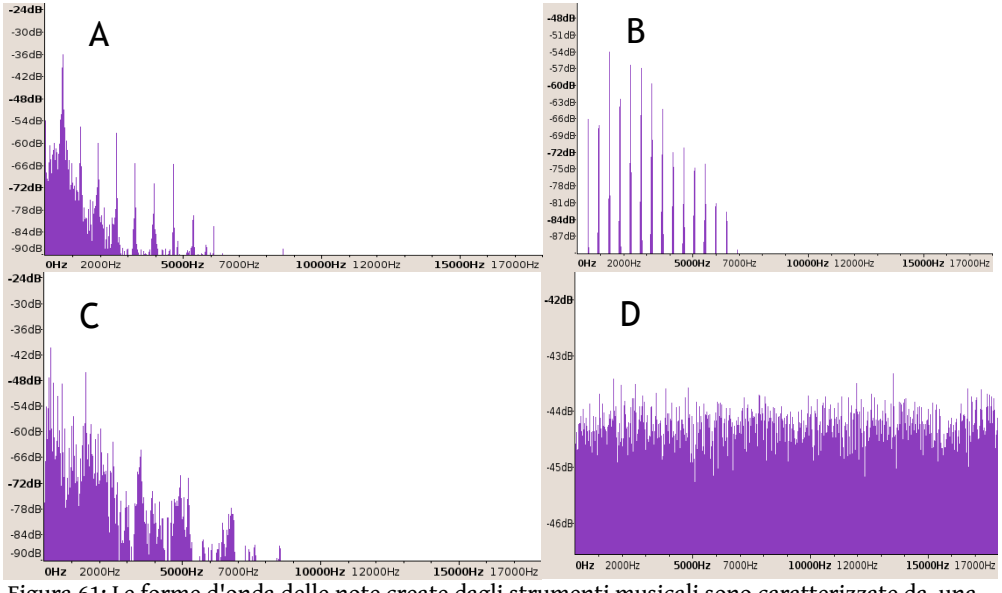


Figura 61: Le forme d'onda delle note create dagli strumenti musicali sono caratterizzate da una particolare composizione di frequenze pure multiple di una frequenza base (A: pianoforte, B: tromba, C:chitarra). Quella del rumore (D) appare una sovrapposizione casuale di frequenze.

Mbyte! Un CD musicale, in effetti, su un supporto che può contenere 640 Mbyte, non può contenere più di una quindicina di canzoni. Per poter trasmettere l'audio e memorizzare più dati nei comuni supporti sono state sviluppate anche in questo caso tecniche di compressione. Come nel caso delle immagini queste tecniche possono essere sia lossless (senza perdita) che lossy (con perdita). Un modo per limitare le dimensioni dei file è certamente quello di limitare la qualità al tipo di dato e applicazione richiesti. Se, infatti, per avere la qualità del CD musicale occorre un campionamento a 44 KHz e 16 bit di profondità, se vogliamo riprodurre audio alla qualità della radio possiamo utilizzare un campionamento a 22 KHz ed eliminare l'effetto stereo, ottenendo una riduzione di un fattore 4. Per riprodurre bene il parlato, in cui il range dinamico (rapporto tra livello più alto e più basso di volume) è basso e le frequenze sono limitate, basta campionare a 11 KHz e usare 8 bit per la binarizzazione, ottenendo quindi un ulteriore fattore 4 di riduzione.

I principali formati usati per memorizzare l'audio PCM supportano diverse frequenze e profondità. Tra questi ricordiamo quello standard su Internet **.au**, supportato dai browser anche se in disuso, il **.wav** di Microsoft Windows e **.aiff** di Apple.

Nel tempo si sono poi sviluppati metodi efficaci per comprimere ulteriormente i file audio. Quelli lossless consentono di comprimere i file del 50-60% e quindi non raggiungono prestazioni adatte per la trasmissione in rete o per l'uso nei lettori portatili. Essi però consentono di ridurre il tempo di scaricamento dei brani senza ridurre la qualità e vengono quindi usati dall'industria musicale per la distribuzione via rete dei CD audio. Un esempio di formato di compressione audio lossless è **flac**.

In genere però, quando parliamo di codifica audio compressa pensiamo sicuramente a **mp3** ed affini, cioè a tecniche lossy, basate su teorie di psicoacustica.

Senza andare nel dettaglio dei vari algoritmi, diciamo quali sono i principi utilizzati per ottenere la compressione:

- la percezione umana delle frequenze non è uniforme: si può quindi ridurre l'accuratezza negli intervalli di frequenza peggio percepiti;
- un suono ad alta intensità ne “maschera”, cioè ne rende impercettibile un altro con frequenza vicina, ma intensità più bassa. Si possono quindi eliminare le frequenze mascherate dal segnale;
- la percezione spaziale (stereo) non è uniforme alle varie frequenze (alle basse non si percepisce, ed è il motivo per cui il subwoofer, cioè l'altoparlante riservato ai bassi degli impianti audio cinematografici è unico, mentre quelli per gli alti sono diversi e distribuiti spazialmente) e si può, quindi, memorizzare segnali distinti per destra e sinistra solo in determinate regioni di frequenza.

Come accennato, il primo metodo di codifica basato su simili principi è stato MPEG 1-Layer 3, meglio noto con il nome dell'estensione dei file, cioè .**mp3.** Questo formato ha rivoluzionato il mondo dell'intrattenimento digitale, consentendo la distribuzione della musica online. Esso consente la riduzione di un fattore 10 della dimensione dei file senza compromettere in modo evidente la qualità audio (comunque degradando il dato originale!). Il formato .mp3 non è un formato libero, anche se esistono software liberi per codifica e decodifica. Esso supporta l'aggiunta al file di metainformazione (ad esempio titoli, genere musicale, autore) mediante i

cosiddetti **tag ID3**.

Il formato .mp3 non è adatto per la distribuzione musicale da parte delle case discografiche in quanto non supporta metodi di controllo sui diritti d'uso e sulla copiatura (Digital Rights Management o **DRM**).

Oggi esistono diversi formati e metodi per la codifica compressa anche più efficienti o completi di mp3 come **OGG Vorbis**, libero e più efficiente del 25% nella compressione, e **Advanced Audio Coding (.aac)**, formato standardizzato da ISO e parte delle specifiche Mpeg2 e Mpeg4, adottato, per esempio, da iPod e iTunes, Playstation, ecc.

Generalmente i formati di files che gestiscono l'audio digitale (es. .wav, realaudio, ecc.) ed anche il video digitale, sono "contenitori" in cui sono definite le solo le informazioni generali sui contenuti e a cui sono collegati poi i dati veri e propri codificati attraverso metodi definiti esternamente al formato stesso. Le componenti software che si occupano della compressione e decompressione vengono definiti **codec** (da Coder-DECoder).

Per questo capita a volte usando la rete di scaricare e leggere un file correttamente con il lettore musicale ma poi ricevere il messaggio che il lettore è sprovvisto del codec per visualizzare il file stesso e ne debba richiedere l'installazione.

Per terminare l'analisi sui formati di file audio digitale, occorre dire che, come per le immagini, anche per l'audio esiste un formato che, anziché memorizzare i dati campionati e binarizzati, dia semplicemente istruzioni standard per la generazione delle sequenze di suoni a programmi in grado di generarli.

Per la musica la descrizione corrispondente al disegno vettoriale consiste nel codificare la partitura musicale (sequenze di note temporizzate per i vari strumenti utilizzati). Il formato che lo codifica fa parte del protocollo detto **MIDI** (Musical Instrument Digital Interface), il protocollo standard per l'interazione degli strumenti musicali elettronici.

I file MIDI, estensione .mid, descrivono le sequenze di suoni su diversi canali codificandone note e temporizzazione (in pratica si rappresenta la partitura musicale). La rappresentazione sonora generata dal computer sarà quindi dipendente dall'hardware e dal software utilizzati. Se la scheda audio genera le note degli strumenti musicali con qualità elevata, il risultato può essere anche molto buono, con occupazione di memoria sul file molto bassa. I file MIDI sono anche supportati dal web e possono quindi essere usati per generare sottofondi musicali ed effetti sonori negli ipertesti.

Distribuzione dei contenuti audio (video): podcast, streaming

I file contenenti informazione audio sono in genere scaricati via HTTP, FTP o mediante programmi di file sharing ed ascoltati su lettori per PC o portatili. Per gestire lo scaricamento automatico di contenuti audio, un metodo di gestione è quello dei cosiddetti **podcast**, termine che deriva dalla contrazione di "PoD" (Portable on Demand, cioè portatile su richiesta) e "broadcast" (trasmissione a utenti multipli).

Il meccanismo si basa sulla tecnologia dei feed XML/RSS (vedi capitolo 3), che permettono ai programmi client (es. iTunes di Apple o Juice) di avere notifica degli aggiornamenti dei contenuti cui ci si è iscritti (mostrando, ad esempio, puntate, locazione dei file, dimensioni, autore, ecc.) e di programmare gli scaricamenti. Questo metodo va bene per ottenere i brani da ascoltare in seguito sul PC o sull'iPod.

Un'altra forma di distribuzione dell'audio attraverso la rete è quella che si crea riproducendo il segnale sonoro man mano che i bit vengono trasferiti, senza dover aspettare lo scaricamento completo. Questo metodo si chiama **streaming**, da stream (flusso) e può essere realizzato in vari modi. Il più comune è il cosiddetto streaming "on demand", in cui i file audio memorizzati su un server e l'utente richiede in modo asincrono l'invio dei contenuti audio/video. Il file non viene quindi scaricato per intero sul PC e poi eseguito, ma i dati ricevuti vengono decompressi e riprodotti pochi secondi dopo l'inizio della ricezione. Per ridurre i problemi derivanti da ritardi e dalla variabilità della velocità di trasmissione, i programmi client utilizzano una tecnica detta buffering, cioè prima di avviare la riproduzione memorizzano localmente alcuni secondi di audio per compensare appunto eventuali "buchi" di trasmissione dovuti a variazioni nel tempo della velocità di scaricamento (il cosiddetto "jitter").

Lo streaming "**on demand**" si può ottenere sia con il solo server web, integrando opportunamente i file nelle pagine web con l'elemento `<object>`, sia attraverso server di streaming "dedicati" (e quindi protocolli particolari di trasmissione, come RTSP, Real Time Streaming Protocol) che garantiscono maggiore qualità. Si noti che lo streaming on demand è particolarmente oneroso per il server che dovrà gestire, in caso di molte richieste, la trasmissione di notevoli quantità di dati in maniera indipendente a ciascun cliente.

L'altra forma di streaming realizzabile è quella "**live**" (dal vivo) in cui c'è sincronizzazione tra trasmissione e ricezione, come, quindi nelle trasmissioni radiofoniche e televisive. In questo caso, ovviamente, la gestione è più complessa, occorre fare in modo che i pacchetti vengano instradati in maniera "multicast", cioè a più indirizzi contemporaneamente, anche se il traffico generato sulla rete è inferiore dato che non si devono reinviare i dati per ciascun cliente.

Software audio

Anche per catturare, generare elaborare, comprimere i files audio esistono molti programmi disponibili, tra cui ottimi prodotti di software libero. Ad esempio, per catturare e codificare l'audio, elaborarlo e salvarlo con codifica mp3 o altri formati con la qualità desiderata, possiamo sicuramente segnalare il pacchetto **Audacity** (*http://audacity.sourceforge.net*), open source disponibile per tutte le principali piattaforme (Windows, Linux, Mac).

Esso permette di catturare l'audio dal microfono o di importarlo da file di vario formato (anche MIDI) e di lavorare su diverse tracce, componendole a piacimento.

Tra le varie elaborazioni che permette di ottenere su ciascuna traccia registrata ci sono ad

Figura 62: Finestra principale del programma Audacity (a sinistra), con il grafico del segnale in funzione del tempo. A destra si vede il tool per visualizzare lo spettro in frequenza.

esempio il taglio e il copia e incolla: come per testo si può selezionare una porzione audio allo scopo di rimuoverla, riprodurla, sostituirla, effettuando eventuali operazioni di missaggio o dissolvenza (crossfading).

E' anche possibile effettuare un resampling, cioè ridefinire il tasso di campionamento di un segnale, ma anche variare il range dinamico, o operare un filtraggio, ad esempio con il filtro passa basso che elimina le alte frequenze, il passa-alto (che elimina le basse), o il passa banda (che preserva solo le frequenze di interesse). Per migliorare la qualità dei propri podcast si possono poi applicare opportune procedure di riduzione del rumore o di eliminazione dei click. Se si vuole rendere più alto o più basso il suono della propria voce, si può anche applicare il cosiddetto cambio del pitch, che sposta la frequenza dominante del segnale lasciando inalterata la durata dei suoni.

Il risultato delle elaborazioni può poi essere esportato in file di vario formato, controllandone i parametri di qualità e compressione.

Una buona norma per generare podcast di lezioni o discorsi ottimali per la distribuzione in rete è quella di controllare i parametri di campionamento (per la voce è sufficiente anche a 11 Khz), e compressione. Nei menu viene proposta la scelta tra vari bitrate, cioè memoria occupata per secondo espresso generalmente in Kbit/s: per la voce bastano tassi molto bassi (es. 16 Kb/s), mentre per la musica occorre almeno superare i 128 per avere una resa appena accettabile.

Oltre ai programmi di elaborazione dell'audio digitale, esistono poi i software musicali, per esempio i cosiddetti sequencer, utili per la composizione assistita di partiture musicali che possono essere poi salvate nel formato MIDI. Anche in questo caso esistono buone soluzioni open source, anche se in genere non confrontabili con gli strumenti professionali più diffusi (Cubase, Pro Tools).

5.6 Video digitale

Come per il video analogico, l'effetto del movimento nei filmati digitali è creato da una sequenza temporale di immagini statiche (frames). Il fenomeno della persistenza delle immagini sulla retina fornisce quindi all'utente l'illusione del movimento.

Il cervello riesce ad “acquisire” una nuova immagine ogni circa 1/10 di secondo (frequenza di fusione), pertanto la sequenza di immagini deve presentarsi agli occhi con sufficiente velocità. La frequenza di aggiornamento dell'immagine nel video digitale, ma anche nel cinema e nella televisione analogica, prende il nome di **frame rate** e si misura in fotogrammi al secondo (frames per second, fps).

Tanto più il frame rate sarà elevato, tanto più il movimento apparirà fluido. I primi film muti avevano un frame rate di 16 fps, poi passati a 24 col sonoro, gli standard televisivi classici presentano una frame rate di 25 o 30 fps, quelli della TV ad alta definizione 50-60 fps.

La televisione analogica trasmette fotogrammi mediante segnali continui (segnali elettrici contenenti le informazioni sui colori, l'audio e i segnali di sincronizzazione). Per il video digitale in rete si utilizzano ovviamente formati che codificano digitalmente i frame (inclusi, ovviamente, i segnali di sincronizzazione e controllo), aggiungendo eventualmente la traccia audio in formato compresso.

Sia per il caso analogico che per il digitale, la grande mole di informazione da trasferire ha sempre creato la necessità di ridurre la dimensione di questi dati, limitando le dimensioni delle

immagini o adottando apposite tecniche di compressione.

Nelle trasmissioni televisive, oltre all'avere una risoluzione limitata (nel nostro formato PAL 720x576) si trasmette in generale in maniera interlacciata: ad ogni intervallo temporale, anziché ritrasmettere tutta l'immagine, si trasmettono solo le righe pari o dispari alternativamente: per questo nei fermo-immagine di videoregistratori e DVD appaiono spesso le caratteristiche frastagliature (Figura 63). Quando invece l'immagine viene trasmessa tutta ad ogni istante si parla di "progressive scan". Inoltre, dato che l'uomo percepisce più in dettaglio le variazioni di luminosità che quelle di tinta, un altro metodo applicato negli standard televisivi e home video consiste nella trasmissione della componente di luminosità del colore a piena risoluzione e delle componenti di tinta (crominanza) a risoluzione ridotta in genere di un fattore 2 in una direzione o in entrambe.

Oggi per la trasmissione ed elaborazione dei video, non solo su Internet, si usano in generale formati digitali. Se convertiamo in digitale l'immagine in standard PAL (lo standard televisivo utilizzato in Italia), avremo, per ogni secondo, 25 immagini con 720x576 pixel in cui il colore è rappresentato da 3 componenti, che non in questo caso sono RGB, ma la luminanza (in pratica il bianco e nero) e 2 componenti dette di crominanza. Facendo due conti, quindi, avremmo che, senza compressione, un secondo di video occuperebbe 720x576x3x25=31Mb. Utilizzando gli schemi di sottocampionamento e trasmissione progressiva possiamo ridurre un po' la dimensione, ma per avere trasmissione in rete e streaming del video digitale, come abbiamo oggi su Internet, ed anche poter comprimere i film in modo tale da memorizzarli, come usiamo fare, su comuni CD e DVD, è chiaro che si devono adottare metodi di compressione più efficienti.

Come nel caso dell'audio e delle immagini, cerchiamo di capire le idee alla base dei metodi di compressione senza andare nel dettaglio tecnico.

E' chiaro che ai singoli frame potremo applicare tutti i metodi di compressione visti per le immagini. Si parla in questo caso di compressione **intra-frame**: con tecniche lossy come JPEG sappiamo di poter ridurre anche di un fattore 10 le dimensioni di una singola immagine pur avendo sempre un buon risultato visivo.

Ma sul video possiamo anche utilizzare altre tecniche di compressione, che in qualche modo sfruttino la ridondanza temporale: se, per esempio, la scena è ferma, in realtà tutti i frame contengono la stessa informazione per cui non serve trasmetterla tutta più volte. Le tecniche che si basano su questi principi si dicono **inter-frame** ed in pratica per una parte dei frame della sequenza non memorizzano i pixel, ma li ricavano dai precedenti o seguenti sulla base di stime locali del moto ottenute dalla sequenza.

Figura 63: Classico effetto di frastagliatura del fermo immagine di sequenze di movimento dovuto alla trasmissione interlacciata.

Gran parte del lavoro di standardizzazione dei metodi di codifica del video digitale è stato svolto dal comitato Moving Pictures Expert Group (MPEG), creato delle organizzazioni internazionali ISO e IEC. Il comitato definisce standard per la rappresentazione in forma digitale di audio, video e altri tipi di contenuti multimediali ed ha rilasciato i seguenti importanti standard relativi al video:

- MPEG2: lo standard più diffuso, destinato al

broadcast televisivo, introdotto nel 1994;

- MPEG4: standard emergente che permette un ulteriore aumento della capacità di compressione, particolarmente adatto allo streaming video.

Da questi standard derivano gran parte delle implementazioni dei cosiddetti “codec” video cioè dei software di codifica/decodifica del segnale che sono supportati dai lettori ed inseriti nei formati di memorizzazione più diffusi.

Ricordiamo che, come nel caso dell'audio, il “codec” (COmpressore-DECompressore) è un software che dice al computer con quali operazioni matematiche deve manipolare le immagini per comprimerle e quali eseguire per visualizzare quelle compresse, il "formato" invece è una sorta di scatola che contiene il codec e lo integra con il sistema. I codec esistenti sono tantissimi al contrario dei formati.

Esistono, pertanto, vari formati di file (DV, .avi, .wmv, flash video o .flv, real video, etc.), che possono includere vari tipi di codec propri o sviluppati da altri, come ad esempio i noti DivX, Xvid, Ffmpeg e 3ivx, tutte implementazioni di MPEG4 part 2.

I codec più efficienti sono quelli che implementano lo standard MPEG-4 Part 10 attualmente lo stato dell'arte per la compressione (adottato su XBOX 360, PlayStation Portable, iPod, Nero, Mac OS X v10.4, HD DVD/Blu-ray Disc) e anche supportato dal formato Flash video, oggi forse quello dominante in rete, adottato da YouTube, Google Video, Reuters.com, Yahoo! Video, MySpace. Come accennato in precedenza, esiste una continua evoluzione di questi codici ed una grossa battaglia commerciale tra le varie aziende per supportare diversi formati e plugin, proprietari o liberi, più o meno efficienti. Tra l'altro la battaglia per far adottare formati e plugin sui browser (o per passare a uno standard aperto come HTML5) si è estesa anche al mondo dei dispositivi portatili (smartphone, tablet, ecc,) sui quali è altrettanto importante garantire efficienza e qualità di riproduzione video.

Anche per il video possiamo in qualche modo parlare di codifica vettoriale, cioè animare sequenze di comandi di disegno invece che di immagini bitmap. In pratica si tratta di creare **animazioni**, cartoni animati realizzati mediante primitive di disegno, che consentono ovviamente una codifica molto più efficiente del video digitale. Il formato più diffuso per la codifica di disegni animati è il già citato Flash (.swf), che, come abbiamo visto, consente anche di includere parti interattive (e di includere tra l'altro i video bitmap nel formato Flash Video). Anche il formato .svg comunque include opzioni per la codifica del disegno animato.

Distribuzione dei video in rete, streaming video

Le modalità di distribuzione del video digitale sono sostanzialmente analoghe a quelle dell'audio. Possiamo anche qui cioè permettere lo scaricamento dei file (magari programmato con il meccanismo del podcast) oppure gestire lo streaming includendo i file nelle pagine web o utilizzando appositi protocolli e programmi client.

Attraverso lo streaming video si realizza anche la cosiddetta “Internet TV”. In questo caso i pacchetti di trasmissione possono essere inviati direttamente a più indirizzi (multicast, contrapposto al comune unicast, in cui i pacchetti hanno un'unica destinazione). Questo riduce il traffico sulla rete, dato che lo streaming, specie video realizzato on demand può creare un traffico notevolissimo sulla rete dato che per ogni client il server deve generare un apposito flusso indipendente di dati e se i clienti contemporanei sono molti, possono verificarsi problemi di efficienza. Anche per quanto riguarda le tecnologie di streaming l'evoluzione è continua così

come la battaglia tra aziende per imporre i diversi tipi di formati, protocolli, e programmi sul mercato dei browser e dei dispositivi mobili.

Software per il video digitale

In genere gli strumenti di acquisizione video (webcam, videocamere portatili, macchine fotografiche digitali, cellulari, ecc.) sono oggi nativamente digitali e permettono di trasferire i file video in qualche formato (es. .avi) direttamente sul PC via cavo, scheda di memoria, DVD. Nel caso si voglia convertire video analogico (es. TV, cassette VHS) in digitale, esistono apposite schede di acquisizione che permettono di catturare e digitalizzare il segnale producendo file in un formato digitale sul PC. Una volta disponibili le acquisizioni effettuate può essere tuttavia utile eseguire delle elaborazioni dei propri video per renderli maggiormente fruibili per il pubblico di destinazione. I programmi che realizzano queste elaborazioni si chiamano video editor o software per il Non Linear video Editing (NLE). Le operazioni che consentono sono quelle di importare da varie fonti (telecamere, videoregistratori, dischi, ecc.) spezzoni audio e video, e di effettuare operazioni di montaggio, cioè tagliare parti, incollare spezzoni, creare titoli, aggiungere o sostituire tracce audio, sottotitoli, ecc. Generalmente si basano su un'interfaccia che presenta la cosiddetta timeline cioè una linea temporale su cui si possono giustapporre i vari spezzoni che comporranno il prodotto finale, la possibilità di vedere in anteprima gli effetti di montaggio e di dare il via al cosiddetto "rendering", ossia la creazione del video finale con le opzioni di qualità video e compressione scelte (cioè utilizzando il codec selezionato). Quest'ultima operazione è particolarmente onerosa dal punto di vista computazionale e richiede in genere molto tempo.

I programmi di montaggio video possono essere professionali, come i noti Adobe Premiere o Final Cut, ma se non si hanno ambizioni di produzione cinematografica ne esistono molti gratuiti od open source, semplici, ma più che sufficienti per realizzare gli effetti necessari a creare un video efficace per il web.

Microsoft distribuisce con i sistemi operativi un programma semplice da usare e funzionale, Movie Maker, altro software gratuito e proprietario per Windows è, ad esempio, Pinnacle Videospin. Ci sono anche però buone soluzioni open source disponibili gratuitamente per i diversi sistemi operativi, come Kino, Avidemux, Cinelerra, OpenShot editor o VirtualDub.

Oltre al montaggio, una più semplice operazione che può essere utile realizzare sui video è la sola conversione di formato, per salvare il video nel formato di schermo e con i codec audio e video ottimali per la propria applicazione. Ricordiamo infatti, che se vogliamo inserire contenuti video nei nostri siti dovremo preoccuparci di ridurre al massimo le dimensioni dei file ed usare formati che i destinatari possano visualizzare. Anche qui esistono programmi semplici per convertire formati, anche

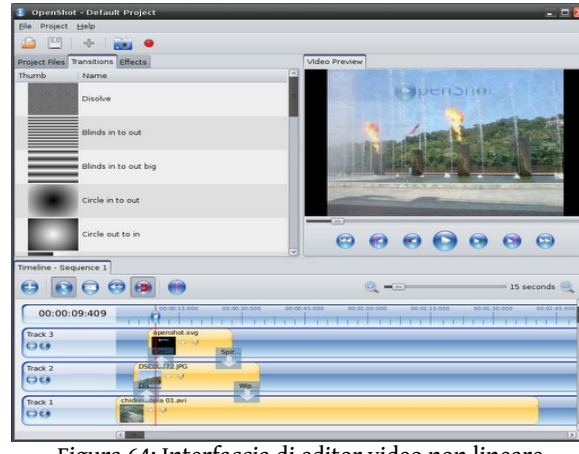


Figura 64: Interfaccia di editor video non lineare (OpenShot), si nota in basso la timeline su cui si possono inserire gli spezzoni video e audio.

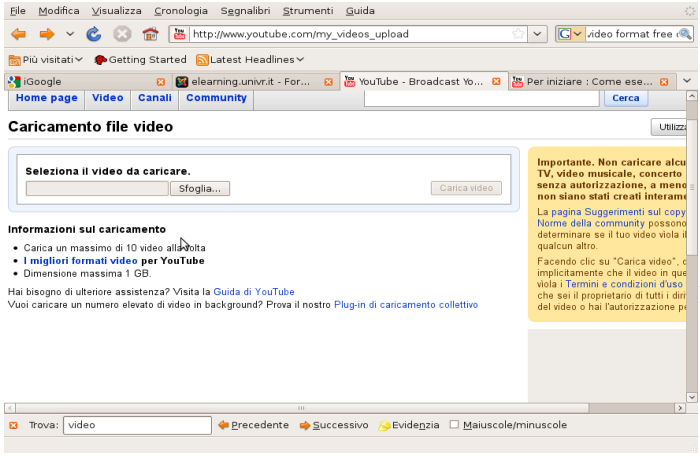


Figura 65: Interfaccia per la pubblicazione di video sul sito YouTube: il video inserito nel formato nativo verrà convertito dal server nel formato Flash Video.

se a volte non è facile orientarsi tra gli innumerevoli formati e codec. Notare che molto spesso chi inserisce video sul web lo fa sul noto portale YouTube e poi collega alle proprie pagine il link che crea la visione integrata (basta copiare la parte di codice dell'object/embed inserito nella corrispondente pagina del sito). Questo sicuramente può semplificare la vita, dato che il sito accetta l'inserimento di video in svariati formati, compresi quelli tipici dei dispositivi portatili. Basta registrarsi al sito, e inserire attraverso l'interfaccia i propri file (Figura 65): il sito stesso provvederà alla conversione del file nel formato ottimale per lo streaming sul web (Flash Video). Ovviamente occorre però ricordare che questo farà sì che si debbano accettare delle condizioni di contratto relative la pubblicazione dei video che, sebbene non debbano necessariamente creare problemi, andrebbero ovviamente considerate con cura. Se non si vogliono problematiche di questo tipo, è preferibile allora gestire direttamente la conversione e la pubblicazione dei contenuti.

5.7 Scene 3D e realtà virtuale

Un'altra tipologia di contenuti multimediali spesso usati sui PC consiste nella creazione di ambienti virtuali **tridimensionali** e nella loro visualizzazione interattiva (si consente all'utente di muoversi nella scena e di cambiare il punto di vista, eccetera). Si parla in questo caso di "**realtà virtuale**" in quanto si simula realisticamente il mondo reale attraverso il calcolatore. Questo avviene ormai da anni nei videogiochi, ma chiaramente ci sono molte applicazioni didattiche (basti pensare ai simulatori di volo o quelli chirurgici) ed anche di grande successo popolare (si pensi ai programmi di navigazione tipo "Google Earth").

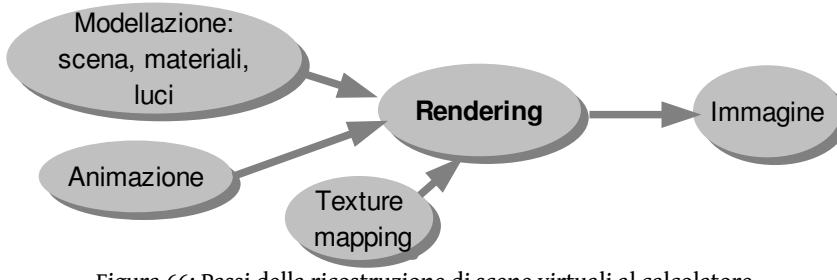


Figura 66: Passi della ricostruzione di scene virtuali al calcolatore

Come viene realizzato tutto questo, ovvero, come sono codificate le scene “virtuali”?

Il primo passo consiste nel creare un modello della scena di interesse. Esso comprenderà sia la geometria (in generale rappresentata da modelli composti da triangoli che sono facilmente trattabili dalle moderne schede grafiche) che le proprietà di riflessione della luce delle varie parti nonché delle sorgenti di illuminazione. A partire dalla descrizione della scena esistono programmi in grado, dato un punto di vista e il modello della telecamera virtuale che riprenderebbe la scena, di creare l'immagine (o le immagini se si vuole addirittura creare una coppia di immagini stereo per simulare la percezione dei due occhi) che l'utente immerso nella scena in tale posizione vedrebbe. Questa operazione di generazione delle immagini viene detta **rendering**, “resa”, e come si può immaginare, nonostante si usino modelli semplificati è estremamente complesso (in pratica si simula matematicamente la fisica dell'illuminazione). Per questo motivo i calcoli relativi al rendering vengono fatti eseguire alle schede grafiche dei calcolatori che presentano hardware parallelo e sono programmabili utilizzando linguaggi appositamente creati per calcolare le immagini a partire dalle primitive grafiche (modelli geometrici composti da triangoli, posizione del punto di vista, modelli di telecamera ed illuminazione).

Tra le opzioni gestibili dai sistemi di rendering ci sono l'uso modelli di illuminazione più o meno semplificati, la possibilità di proiettare immagini sulle superfici rappresentate (texture mapping), quella di gestire differenti effetti di ombreggiatura (shading), trasparenza, attenuazione della luce con la distanza e così via. In Figura 67 sono mostrati alcuni effetti tipici utilizzati per il rendering di modelli.

La realtà virtuale vera e propria necessita anche della realizzazione di sistemi particolari di interazione per rendere all'utente la sensazione di trovarsi nella scena modellata. Questo implica, ad esempio, l'uso di display tridimensionali come caschi binoculari o gli schermi speciali con occhiali che permettono di dare l'illusione della profondità mediante la stereoscopia, oppure le grotte (cave) ove la scena sia proiettata intorno all'utente. Implica anche la necessità di aggiornare la scena in funzione dei movimenti dell'utente, per rappresentare sempre correttamente il suo punto di vista. Implica, magari, anche di abbinare alle sensazioni visive anche sensazioni tattili e sonore. I sistemi di realtà virtuale di questo genere (**immersivi**) sono, quindi, costosi e difficilmente accessibili. La cosiddetta realtà virtuale immersiva è quindi riservata ad applicazioni particolari come simulatori di volo, simulatori chirurgici, di guida, ecc.

In realtà, però, siamo abituati a vedere versioni semplificate di questa realtà virtuale sugli schermi dei nostri PC quotidianamente, non soltanto relativamente a videogiochi tridimensionali, ma anche utilizzando Internet. Esistono, infatti, siti web che integrano contenuti tridimensionali, generati esattamente come visto sopra. Ricordiamo a questo proposito il caso,

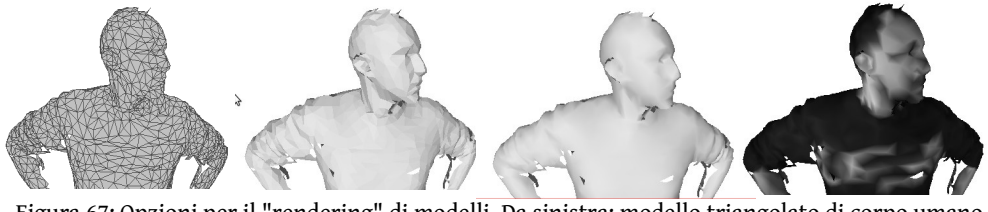


Figura 67: Opzioni per il "rendering" di modelli. Da sinistra: modello triangolato di corpo umano rappresentato dalla proiezione dei poligoni sul piano immagine (wireframe rendering). Stesso modello con effetto di illuminazione e ombreggiatura ottenuto simulando l'effetto di una sorgente luminosa e assegnando proprietà di riflessione al materiale. Eliminazione degli spigoli mediante algoritmo di shading più avanzato. Applicazione di texture ricavata dall'acquisizione fotografica.

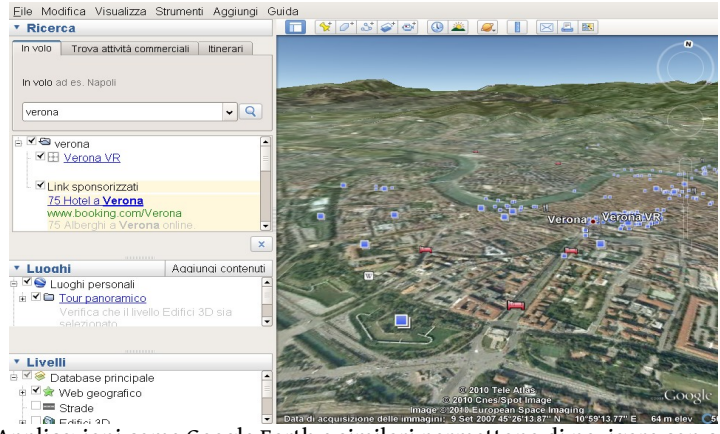


Figura 68: Applicazioni come Google Earth e similari permettono di navigare con comandi tipo aereo su un modello tridimensionale della Terra scaricato progressivamente dalla rete e raffinato in funzione della posizione del punto di vista.

molto sopravvalutato a livello mediatico, del sito Second Life, in cui l'utente può muovere in un ambiente simulato 3D un modello che lo rappresenta o **avatar** (parola dal sanscrito che significa "incarnazione") ed interagire con altri utenti in rete.

Ma esistono anche applicazioni in rete molto utili basate sulla navigazione in ambienti virtuali, basti pensare a Google Earth (*http://earth.google.com*) e simili (Figura 68), che permettono di navigare sulla superficie terrestre vedendone la struttura in dettaglio ed in tre dimensioni. Queste sono applicazioni di quella che possiamo chiamare realtà virtuale "da scrivania" in cui la visualizzazione tridimensionale è forzatamente limitata a un punto di vista e la navigazione dell'utente è vincolata all'uso di particolari metafore che permettano di ricavare i movimenti 3D dall'uso di tastiera e mouse (ad esempio quella della navigazione aerea, che viene controllata tramite l'avanzamento e gli angoli di rollio beccheggio ed imbardata). L'industria dei videogiochi sta comunque creando dispositivi per manipolare le scene virtuali con azioni direttamente in 3D, per cui si possono realizzare effetti di interazione "naturale" anche con il PC di casa (l'esempio classico è la console per videogiochi Nintendo Wii che ha un controller in grado di trasferire all'unità di calcolo il movimento in 3D).

Per la rappresentazione di scene 3D sul web esistono appositi standard (si parla in questo caso di "web 3D". Il primo standard definito si chiamava **VRML** (Virtual Reality Modeling Language) e consentiva di definire completamente geometria della scena ed illuminazione. Diverse case produttrici hanno sviluppato plugin per l'inserimento di tali scene nelle pagine web con il normale meccanismo dell'object/embed e per la visualizzazione a sé stante delle scene. Tuttavia la diffusione dell'uso di tale standard non è mai diventata elevatissima, lo stesso è accaduto con lo standard successivo a VRML, detto **X3D**, derivato dal linguaggio di markup XML (lo stesso di XHTML e dei feed RSS). Esistono comunque vari plugin gratuiti in grado di visualizzare i contenuti VRML e X3D (es. Octaga player, Cortona 3D, OpenVRML, ecc.), sebbene la compatibilità tra i vari sistemi lasci ancora abbastanza a desiderare. Anche per questo, le applicazioni più avanzate sono realizzate in genere con tecniche proprietarie (e sono in ogni caso di realizzazione complessa, anche perché occorre studiare le modalità di interazione). VRML e X3D permettono, comunque, di rendere abbastanza semplice aggiungere una visualizzazione interattiva di modelli 3D in pagine web e questo può essere utile per scopi educativi, in quanto l'esplorazione di scene in tre dimensioni consente di acquisire più

```
<Scene>
<Background skyColor='1 1 1'/>
<Viewpoint description='Side View' position='0 0 3'/>
<Viewpoint description='Book View' orientation='-1 0 0
0.68' position='0 1.11 1.93'/>
<Shape>
<Cone height='1'/>
<Appearance>
<Material/>
</Appearance>
</Shape>
</Scene>
```

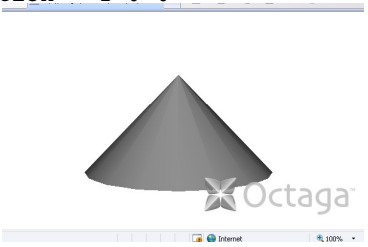


Figura 69: Frammento di codice X3D e scena visualizzata risultante (il cono): è evidente la descrizione con i punti di vista, illuminazione, materiale (qui senza attributi).

facilmente ed in maniera personale le informazioni spaziali sugli oggetti (si pensi a rappresentazioni di pianeti, di molecole in chimica, di aree geografiche, di monumenti, eccetera). La Figura 69 mostra un semplice esempio di modello 3D ed il codice VRML che lo genera, mentre in Figura 70 è mostrato un esempio di applicazione didattica per lo studio dell'anatomia del cervello realizzata con tale tecnologia. In tale esempio l'utente può vedere il modello 3D da diversi punti di vista e capire meglio la disposizione spaziale ed i collegamenti funzionali delle varie parti. Se non ci limitiamo alle applicazioni web, l'uso didattico della grafica 3D e di varie forme di realtà virtuale è decisamente ampio (ad esempio per il training di piloti, chirurghi, militari). Vari studi hanno mostrato come anche l'uso di semplici videogiochi 3D possano allenare con successo determinate abilità spaziali (per esempio quelle utili ai chirurghi). Ovviamente la realizzazione di veri e propri simulatori o anche di videogiochi 3D non è alla portata dei non addetti ai lavori ed è difficilmente realizzabile sul web. Quello che può essere comunque utile, per chi si occupa di formazione, è sapere che anche prodotti generici (videogame, simulatori, mappe 3D) disponibili sul mercato possono essere efficacemente impiegati per l'apprendimento di importanti abilità motorie o di contenuti con relazioni spaziali.

5.8 Contenuti multimediali e informatica pervasiva

In questo testo ci siamo occupati di descrivere le tecnologie legate alla creazione di contenuti ipertestuali e multimediali, descrivendo particolarmente il fenomeno Internet, che ci offre la possibilità di fruire dal nostro personal computer di tali contenuti e di interagire attraverso la rete con le altre persone. Come abbiamo scritto nel capitolo 1, stiamo oggi vivendo un'epoca di profondo mutamento nell'utilizzo degli strumenti informatici. Da una parte le tecnologie del Web sono mature e consentono, come abbiamo visto la realizzazione di servizi molto efficaci per la comunicazione e l'apprendimento, dall'altro, il modello di utilizzo dei calcolatori sta cambiando molto rapidamente a causa della diffusione di nuove tipologie di calcolatore “nascosto”, o meglio, dotato di differenti modalità di interazione.

Molti dei contenuti di cui ci siamo occupati verranno probabilmente fruiti in futuro almeno in parte in modo differente, utilizzando al posto di PC desktop e laptop, dispositivi portatili come e-book, tablet e smartphone, oppure ambientali come lavagne interattive. La rete su cui viaggeranno i contenuti sarà sempre gestita con le stesse tecnologie, è possibile tuttavia che molti dei contenuti multimediali non siano più gestiti tramite il Web, ma utilizzando altri protocolli di Internet, come già avviene con molte applicazioni per cellulari, iPad, eccetera.

I concetti e le informazioni di base che abbiamo fornito nei capitoli 2-5 resteranno comunque fondamentali per farsi un'idea di quali siano le basi tecnologiche delle applicazioni interattive e multimediali in rete, quello che cambierà in modo probabilmente imprevedibile saranno le modalità di interazione degli utenti ed i protocolli e le applicazioni di alto livello.

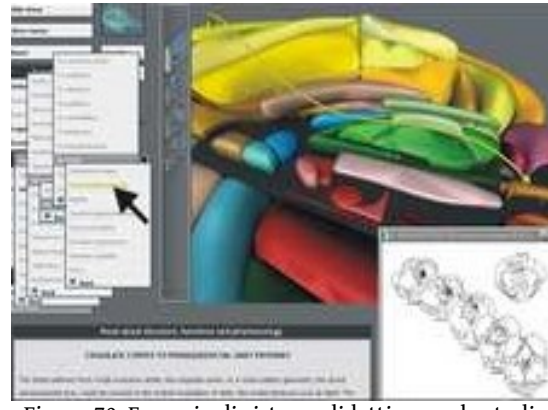


Figura 70: Esempio di sistema didattico per lo studio dell'anatomia cerebrale che usa tecnologie web 3D

Quello che sarà sicuramente importante per capire e prevedere gli utilizzi delle nuove tecnologie sarà anche l'analisi dell'usabilità in termini di efficacia ed efficienza dei nuovi terminali e delle nuove applicazioni. Saranno sempre più importanti, in definitiva, la collaborazione e la comprensione reciproca tra sviluppatori di sistemi e programmi, psicologi, ed esperti di comunicazione. Capire quali siano le opportunità offerte dalle nuove tecnologie sarà sempre più un problema legato alla comprensione dell'uomo e delle le sue esigenze e sempre meno un problema puramente tecnico ed informatico.

6 Web e multimedialità in azione: le piattaforme di e-learning

Dato che questo testo è rivolto soprattutto a studenti di Scienze della Formazione, la nostra carrellata sulle moderne tecnologie di Internet e del web si conclude mostrando come molte delle componenti viste nei capitoli precedenti possano essere in qualche modo integrate per realizzare nella pratica quelle piattaforme per supportare l'educazione "centrata sull'utente" che oggi sono una realtà in crescente diffusione.

Per **e-learning** intendiamo un processo educativo centrato sull'utente che sfrutti le tecnologie informatiche per fornire varie modalità di fruizione di contenuti ed attività, organizzate opportunamente da chi cura il processo stesso.

Si tratta, quindi, di qualcosa di ben diverso della cosiddetta **formazione a distanza** (semplice trasmissione di lezioni per chi non può accedere al luogo fisico della didattica).

L'e-learning, che, se realizzato sul web possiamo anche chiamare **web learning**, comprende, infatti, molte forme di interazione che vanno al di là della lezione in videoconferenza (sincrono: stesso tempo, luogo diverso) ed anche al di là della fornitura di materiale di consultazione (fruizione asincrona) che ormai molti docenti sono abituati a fare tramite i siti istituzionali.

L'uso delle tecnologie informatiche e di rete, unito alle normali procedure didattiche, consente di fornire allo studente la possibilità di accedere a tutte le differenti modalità di interazione rappresentabili nella cosiddetta **matrice spazio-tempo** (Figura 71). Essa mostra vari tipi di strumenti interattivi classificati sulla base del fatto che erogazione e fruizione avvengano nello stesso luogo (co-situati) o in luoghi differenti (remoti) oppure che avvengano allo stesso tempo (sincroni) o in tempi diversi (asincroni). Inoltre l'e-learning consente allo studente di utilizzare tali strumenti non soltanto per l'interazione col docente, ma anche per dialogare con altri studenti o per cercare informazioni all'esterno.

Le varie componenti dei sistemi di e-learning possono essere efficacemente integrate per creare percorsi didattici completi e personalizzati tenendo conto delle caratteristiche specifiche degli utenti. E' chiaro, infatti, che avere a disposizione modalità alternative di erogazione dei contenuti consente di adattarli meglio alle singole problematiche (pensiamo al problema dell'accessibilità di cui abbiamo discusso).

Il rischio, a volte paventato, di minore controllo o di sminuire il ruolo del docente è in realtà tutto relativo all'uso limitato o improprio degli strumenti: infatti è sicuramente molto più impegnativo per chi cura la didattica, cercare di progettare percorsi che integrino più sorgenti di apprendimento adatti a utenti diversi piuttosto che preparare semplici lezioni frontali (basti pensare al necessario controllo sul materiale o sulle attività di comunicazione tra studenti).

Non è argomento di questo testo analizzare le teorie filosofiche e pedagogiche dell'e-learning, ci limiteremo solo ad analizzarne le possibilità tecniche nel caso specifico delle piattaforme web già disponibili che sostanzialmente non fanno altro che usare le tecnologie viste nei capitoli precedenti applicandole ai processi della didattica.

Come abbiamo visto, i siti web moderni consentono di distribuire testi e documenti scritti (appunti, dispense, ecc.). Con le tecniche di codifica che abbiamo visto si possono acquisire e codificare audio e video (quindi, per esempio, le lezioni) e li si possono trasmettere in streaming, sia in diretta che on demand, oppure renderle scaricabili come podcast. Internet consente anche la comunicazione asincrona tra studenti e col docente (per es. con la posta elettronica) ed anche quella sincrona (chat o telefonia su IP).

L'uso di script e pagine attive consente di realizzare contenuti interattivi come test utili per l'autovalutazione, giochi di abilità, mappe utili per l'apprendimento.
I sistemi di gestione delle basi di dati dei siti consentono anche di automatizzare la parte amministrativa dei corsi (iscrizioni, esami, ecc.).
Le tecnologie del web e della multimedialità digitale consentono, quindi, oggi di integrare tutte queste cose in singoli siti che consentono, un facile uso ed un buon controllo delle varie attività da parte dei docenti ed amministratori.

Figura 71: Matrice spazio-tempo e collocazione dei vari elementi di un sistema didattico

Questi siti sono oggi realizzati attraverso dei Content Management System specifici per la didattica che, per questo, vengono detti **Learning Management System** e **Learning Content Management System** (LMS e LCMS) a seconda che gestiscano iscrizioni e statistiche o contenuti didattici.
Esistono diverse soluzioni per la gestione di siti di e-learning, più o meno adeguate agli standard che la comunità del settore ha cercato di introdurre (esiste uno standard detto SCORM, Shareable Content Object Reference Model cioè Modello di Riferimento per gli Oggetti di Contenuto Condivisibili che propone specifiche tecniche per rendere gli elementi di un sistema di e-learning il più possibile riutilizzabili ed indipendenti dal sistema).
Tra le varie piattaforme per l'e-learning sul web, ve ne sono anche di ottime "open source" (Vedi Appendice 1). La più utilizzata è senza dubbio **Moodle** (Module Objects Oriented Dynamic Learning Environment, cioè ambiente di apprendimento dinamico orientato agli oggetti modulari) che è anche quella attualmente utilizzata per il sito di e-learning dell'università di Verona (*http://elearning.univr.it*).

Moodle, come tutti i CMS è strutturato come un servizio web, per cui si basa su programmazione server side e sull'uso di basi di dati (php/MySql). L'utente può collegarsi da qualunque località per accedere al materiale e tutta la sua attività viene registrata sul database.

Si parla di "orientato agli oggetti" perché la filosofia del sistema (in generale di tutti i sistemi di e-learning) è quella di modellare le attività didattiche fondamentali come oggetti astratti (learning objects) inseriti poi in istanza pratica nei corsi, tracciando le interazioni degli utenti con esse. Il sistema è modulare e la comunità di utenti e sviluppatori può aggiungere nuovi moduli per fornire nuovi tipi di attività e servizi.

Moodle consente l'inserimento nella piattaforma di due tipi di "materiali": risorse ed attività. Le risorse non sono di per sé interattive, possono essere file di vario tipo, mostrati in finestre pop up oppure "embedded", o anche collegati dall'esterno.

Esiste poi una grande varietà di attività, cioè di sistemi interattivi inseribili nei corsi, come test, quiz, forum, chat, esercizi, workshop, ecc. La modularità fa sì che attività nuove siano costantemente aggiunte e installabili nei sistemi.

E' chiaro che una così grande quantità di strumenti possa consentire, pur di saper controllare i mezzi, la realizzazione pratica di varie tipologie di didattica, dall'arricchimento dei corsi con esercizi e test di autovalutazione, alla messa in pratica di una visione costruttivista mediante i

forum o i wiki.

Il sistema Moodle consente di accedere ai corsi con diversi ruoli, dall'amministratore, al creatore di contenuto all'utente registrato e all'ospite, definendo per ciascun livello di accesso i permessi che limitano la possibilità di creare materiale od utilizzare strumenti.

Se comunque è relativamente semplice gestire l'arricchimento di corsi con materiale multimediale ed interattivo in appoggio ad un corso tradizionale (approccio “blended”) resta invece abbastanza delicata la progettazione di attività di vero e-learning con cui lo studente possa costruire un percorso personalizzato. Si tratta però di problemi più che altro organizzativi e motivazionali: la tecnologia è, invece, ormai matura e fornisce ottimi strumenti per realizzare tali progetti.

Figura 72: Esempio di pagina di corso in elearning realizzata con il LCMS Moodle.

Appendice 1: software libero/open source

Il cosiddetto software “libero” è software che, pur essendo dotato di una licenza d'uso, si distingue dal normale software detto **proprietario**, su cui esiste cioè un titolare che ne impone dei limiti di utilizzo e copiatura dello stesso, per il fatto di non imporre questo tipo di limite, ma, anzi, incoraggia in qualche modo la sua libera diffusione.

Questo software quindi non è però necessariamente privo di condizioni d'uso imposte dall'autore (cioè di **pubblico dominio**), ma gli autori lo rilasciano con licenze d'uso di tipo “**copyleft**” cioè attraverso cui non si fa in modo che gli utenti non possano copiare o diffondere il software o debbano pagare l'autore, bensì si fa in modo che le libertà di uso del programma originale vengano in genere preservate. Esistono comunque vari tipi di licenza con differenti caratteristiche.

Nella maggior parte dei casi le licenze lasciano piena libertà di modifica e di uso dei programmi. Le licenze del software libero secondo quanto stabilito dal suo pioniere Richard Stallmann e dalla **Free Software Foundation** garantiscono che per i programmi ai quali si applicano siano garantite le cosiddette quattro "libertà fondamentali":

- Libertà di eseguire il programma per qualsiasi scopo ("libertà 0")
- Libertà di studiare il programma e modificarlo ("libertà 1")
- Libertà di copiare il programma in modo da aiutare il prossimo ("libertà 2")
- Libertà di migliorare il programma e di distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio ("libertà 3")

Dato che si considera anche la disponibilità del codice sorgente nella definizione di software libero, spesso si usa anche la definizione di software open source, cioè di cui è disponibile il codice sorgente.

La definizione però, non è del tutto coincidente: infatti con Open Source ci si riferisce a quanto stabilito da un'altra associazione detta **Open Source Initiative** le cui “regole” presentano delle differenze rispetto a quanto stabilito dalla FSF. In ogni caso la maggior parte degli applicativi di software libero possono essere definiti open source e viceversa.

In ogni caso la metodologia di sviluppo del software libero/open source sta mostrando enormi potenzialità: le comunità di sviluppatori sono molto attive, sempre più ampie e la motivazione dei partecipanti e la facile comunicazione in rete tra utenti e programmatori rende più semplice la correzione di errori e l'aggiunta di nuove funzionalità alle applicazioni.

Il successo ha fatto sì che molti progetti di software libero abbiano potuto anche beneficiare di investimenti da parte di organizzazioni pubbliche e di privati. Aziende produttrici di software possono infatti avere interesse a sponsorizzare la comunità di sviluppo del software libero in quanto, a seconda del tipo di licenza con cui il software viene rilasciato possono poi utilizzare liberamente i suoi prodotti per le proprie attività commerciali.

Il software libero non coincide con il software gratuito: il software distribuito gratuitamente (**freeware**) può benissimo essere proprietario, mentre in teoria è possibile “vendere” il software libero: le libertà sopra citate non lo impediscono. Questo di fatto accade, tra l'altro, ad opera di grandi aziende, che forniscono insieme ai programmi liberi, assistenza e manutenzione che per l'uso professionale sono sicuramente utili.

Oltre al ben noto sistema operativo Linux, esistono moltissime applicazioni per l'utente che sono software libero, disponibili anche per i sistemi operativi commerciali. Esempi famosi sono il server web Apache, il database Mysql, il client web Mozilla Firefox, la suite OpenOffice, usata tra l'altro per scrivere questo libro e molti altri. I più diffusi sistemi di content management per il web sono open source (es. Joomla) così come la piattaforma di e-learning Moodle. Per l'elaborazione di immagini e audio esistono, come abbiamo visto, ottime soluzioni che possono essere usate anche per uso professionale come The Gimp e Audacity.

Esempi di software libero utile per la creazione di contenuti multimediali

Testo, documenti, disegni, ufficio:

Openoffice	http://www.openoffice.org

Grafica bitmap:

The GIMP	http://www.gimp.org

Grafica vettoriale:

Inkscape	http://www.inkscape.org

Audio:

Audacity	http://audacity.sourceforge.net

Montaggio video:

Kino	http://www.kinodv.org
Cinelerra	http://cinelerra.org
OpenShot Video Editor	https://launchpad.net/openshot

Content Management System:

Joomla!	http://www.joomla.org
Wordpress	http://wordpress.org
Drupal	http://drupal.org
Plone	http://plone.org
Moodle (per e-learning)	http://moodle.org

Appendice 2: la codifica del testo e il web

Nei capitoli precedenti abbiamo analizzato in dettaglio la codifica digitale di molti tipi di dato, ma abbiamo trascurato di parlare della codifica del testo. I dati nel computer sono, come è noto, codificati con **sequenze di bit** (cifre binarie con valore 0 o 1), in genere multipli di 8 bit (1 byte). Tutti i tipi di dato sono, quindi, codificati in tal modo, cioè tutti i possibili valori che può assumere il dato stesso (l'**alfabeto** che caratterizza il tipo di dato) devono essere fatti corrispondere a sequenze di un certo numero di bit. Per rappresentare il testo, il primo modo standard di codifica è stato il cosiddetto codice ASCII, da American Standard Code for Information Interchange (ovvero Codice Standard Americano per lo Scambio di Informazioni). Esso usa 7 bit e fa corrispondere i numeri binari di 7 cifre, che sono 2^7=128 ai differenti caratteri alfabetici. Quindi, quando parliamo di file di testo in codifica ASCII, intendiamo una sequenza di bit che, nota la corrispondenza, viene poi tradotta in una sequenza di caratteri.

Dato che 128 caratteri non erano sufficienti per rappresentare i vari alfabeti particolari, i caratteri speciali, eccetera, si sono poi sviluppati vari metodi di codifica più avanzati, o adattati alle varie regioni, che mantengono, in genere, la compatibilità con ASCII (in pratica usano più bit e mantengono i caratteri codificati da ASCII nei primi 7).

I più recenti protocolli del web utilizzano la codifica dei caratteri del testo detta **Unicode**, sistema che assegna un numero univoco ad ogni carattere ed è usato per la scrittura di testi in maniera indipendente da lingue, programmi e piattaforme utilizzate. Unicode utilizza fino a 21 bit per la codifica. I codici Unicode possono essere, però, trasmessi in formati differenti. Per l'HTML la codifica utilizzata è detta **UTF-8**. Essa supporta dimensioni variabili del codice di un carattere multiple di 8 bit e la limita, per i caratteri più usati, ad un solo byte.

La codifica del testo di una pagina HTML può essere segnalata nella pagina stessa inserendo, nella sezione head del documento, il tag

```
<meta http-equiv="Content-Type" content="text/html;
charset=CODIFICA" />.
```

Attenzione, però, che la codifica deve corrispondere esattamente con quella con cui è stato salvato il file (gli editor di testo permettono di selezionare la codifica).

L'uso di differenti formati di testo per il salvataggio e la decodifica dei caratteri può far sorgere a volte problemi di visualizzazione dei caratteri speciali (es. lettere accentate, simboli), specie se non si usano sistemi operativi o browser che non li supportano pienamente. Un modo per evitare questi problemi è di scrivere invece nelle pagine web tali caratteri attraverso le cosiddette sequenze di escape, sequenze di caratteri ASCII che il browser interpreta come singolo carattere speciale, non incluso nella codifica. Per fare qualche esempio, "è" creerà la lettera è, "Ã" creerà Ã, eccetera (rimandiamo ovviamente ai manuali per l'elenco completo dei caratteri).

Bibliografia/Sitografia (siti visitati 01/11/09)

Per approfondimenti o chiarimenti sugli argomenti relativi ai diversi capitoli, suggeriamo alcune fonti:

Cap. 1,2

Sciuto, Buonanno, Mari, **Introduzione ai Sistemi Informatici.** McGraw-Hill (2008)

Giachetti, Bettio, **Fondamenti di Informatica,** dispense,
http://profs.sci.univr.it/~giachetti/testi/fondamenti.pdf

Wikipedia, voci: "computer", "central processing unit", "rete di calcolatori", "internet", "sistema client/server"

Cap. 3

Castano, Ferrara, Montanelli, **Informazione, Conoscenza e Web per le Scienze Umanistiche**, Paravia, Bruno Mondadori (2009)

Brajnik, Toppano **Creare Siti Web Multimediali**, Paravia, Bruno Mondadori (2007)

D Norman, **La caffettiera del masochista**, Giunti (2005)

J Nielsen **Web Usability**, Apogeo (1999)

Sito W3C: http://www.w3.org

Wikipedia, voci: "sito web", "web 2.0", "accessibilità", "content management system"

Cap. 4

E. Castro**, HTML per il World Wide Web con XHTML e CSS,** Pearson (2003)

Guide HTML/XHTML su sito **html.it**: http://xhtml.html.it/

Guide CSS su sito html.it http://css.html.it/

Wikipedia, voci: "html", "xhtml", "Fogli di stile a cascata"

Cap. 5

Gonzalez, Woods, **Elaborazione delle immagini digitali**, (3 ed.), Pearson Education Italia - Prentice Hall, 2008

Scateni, Cignoni, Montani, Scopigno, **Fondamenti di Grafica Tridimensionale Interattiva**, McGraw-Hill, 2005

Lombardo, Valle, **Audio e Multimedia**, III ed. Apogeo, 2008

Guida Gimp su sito html.it: http://grafica.html.it/guide/leggi/15/guida-gimp

Wikipedia, voci: "immagine digitale" "audio digitale" "video digitale"

Cap. 6

Wikipedia, voci: "e-learning"