Andrea Piccione

Applied Information Technology

II. Il World Wide Web

Marzo 2002

Premessa:

L'idea di ipertesto non dipende di per sé dall'utilizzo di mezzi informatici, ma è in questo contesto che gli ipertesti sono diventati particolarmente significativi. La comprensione di questo strumento è essenziale per costruire un sito web.

I browser sono software che permettono l'esplorazione/navigazione di un ipertesto. Netscape Navigator e Internet Explorer non sono gli unici, ma solo i più famosi.

All'interno del WWW esistono diversi protocolli, non solo HTTP, cui corrispondono diverse azioni.

La conoscenza del funzionamento dei motori di ricerca aiuta la progettazione di un sito web.

Obiettivi:

- imparare a strutturare un ipertesto;
- conoscere il funzionamento dei principali browser;
- conoscere significato e funzionamento di alcuni protocolli utilizzati nel web;
- conoscere i fondamenti della navigazione nel WWW.

Contenuti:

1. Ipertesti
   - 1.1 Struttura
   - 1.2 Lettura
   - 1.3 Costruzione
2. Il World Wide Web
   - 2.1 Architettura client-server del Web
3. Standard utilizzati nel Web
   - 3.1 URL

3.2 Il protocollo HTTP

4. Motori di ricerca

Riferimenti:

- G. Bongiovanni, Appunti del corso di Sistemi di Elaborazione: Reti 2, Università La Sapienza - Roma

1 Ipertesti

Il concetto di ipertesto risale alla fine degli anni '40 e si deve a vari scienziati:

- Vannevar Bush (sistema Memex, basato su microfilm);
- Douglas Engelbart (sistema NLS/Augment, basato su elaboratori interconnessi);
- Ted Nelson (sistema Xanadu, con enfasi sulla tutela dei diritti d'autore: un documento poteva contenere un riferimento ad altri documenti, che venivano inclusi "al volo" in quello referente e mantenevano cos la loro unicità e originalità).

Che cos'è un ipertesto?

È un testo non lineare, cioè un insieme di informazioni collegate tra loro in modo reticolare anziché in sequenza. Esistono diversi possibili percorsi di lettura (path), in genere sulla base di opzioni previste dall'autore

1.1 Struttura

L'insieme dei collegamenti tra le informazioni costituisce la struttura ipertestuale - si può schematizzare con un grafo generalizzato. Si parla di struttura ipertestuale:

- pura se ogni informazione può essere collegata a qualsiasi altra;
- a rete se i collegamenti sono a grafo;
- gerarchica se il reticolo dei rimandi è strutturato ad albero;

- semigerarchica se il reticolo è gerarchico con collegamenti trasversali.

L'esplorazione di un ipertesto viene detta navigazione. L'esplorazione automatica di un ipertesto si fa con un browser . Un ipertesto è un modo di rappresentare ed accedere ad informazioni:

- i nodi sono formati da pezzi di informazione coerenti (nel caso di ipermedia un nodo può includere mezzi rappresentativi di natura diversa)
- gli archi rappresentano relazioni semantiche fra i nodi

L'idea è nata dall'analogia con la memoria associativa umana e influenzata dalla natura della conoscenza. L'idea di ipertesto non dipende di per sé dall'utilizzo di mezzi informatici, ma è in questo contesto che gli ipertesti sono diventati particolarmente significativi, grazie alla capacità supportata dalla macchina di seguire riferimenti in modo automatico È anche grazie al supporto informatico che dagli ipertesti si passa facilmente agli ipermedia, grazie alla possibilità di mescolare modi di rappresentazione diversi

Come è fatto un ipertesto?

E' composto da nodi che contengono l'informazione a disposizione e da collegamenti tra di essi.

I collegamenti (link) possono:

- essere mono- o bi-direzionali;
- collegare:
  - parti diverse di uno stesso documento;
  - un testo con una nota;
  - un testo con un'annotazione;
  - documenti diversi;
- essere di tipi diversi:
  - collegamenti referenziali: sono determinati da un punto di partenza e un punto di arrivo; producono un'unico nodo;
  - collegamenti a parola chiave: sono collegamenti impliciti e producono un insieme di nodi.

Nodi e link nel loro insieme formano una mappa concettuale del sapere che si vuole comunicare/rappresentare con l'ipermedia. La mappa concettuale evidenzia diversi percorsi concettuali possibili all'interno dell'ipertesto, ed i link esistenti tra i nodi. Una mappa concettuale non è un semplice indice

I link sono una parte essenziale di un ipertesto

- pochi link rischiano di forzare la navigazione lungo percorsi obbligati;
- molti link non differenziati tra loro rischiano di far perdere il filo durante la lettura;
- necessari link per tornare indietro, o al punto di inizio, o a una mappa o indice (mancanza di organicità);
- attenzione ai percorsi che si autoreferenziano (possibili loop);
- coerenza e completezza degli elementi cliccabili (parole calde), per facilitare lettura e comprensione.

1.2 Lettura

Occorre un browser, cioè uno strumento che consente all'utente di muoversi nel sistema ipermediale. La visita di un ipertesto può essere più o meno facile, a seconda della sua struttura e delle sue dimensioni.

Dal momento che è facile perdersi, esistono strumenti per facilitare la navigazione:

- segnalibri: per memorizzare una locazione;
- indici e tabelle di classificazione dei contenuti;
- parole chiave.

Aiuti alla navigazione possono essere:

- forniti dal browser;
- previsti nel sistema ipermediale;
- messi in atto dall'utente.

1.3 Costruzione

Dal punto di vista concettuale, occorre:

- individuare obiettivi, contenuti, prospettive;
- decidere la qualità e la quantità dell'informazione;
- strutturare l'informazione in forma modulare, coerente, in unità autoconsistenti e facili leggere e apprendere;
- stabilire le relazioni (legami logici e concettuali) tra le unità di informazione.

Dal punto di vista tecnico, occorre:

- scegliere il software di sviluppo/utilizzo;
- progettare l'interfaccia utente;
  - scegliere i media per presentare l'informazione (rappresentazione dei nodi);
  - scegliere i tipi di collegamento tra i nodi e come distinguerli;
  - scegliere quali ausili alla navigazione fornire;
- implementare.

L'interfaccia è una componente importante di un sistema ipermediale, perché rende il sistema più o meno facilmente fruibile. Dobbiamo progettare il nostro ipertesto suddividendo il testo in più parti in base al contenuto logico:

- pezzi troppo lunghi sono lenti da caricare e poco maneggevoli (può essere difficile reperire materiale al loro interno);
- pezzi troppo corti possono far perdere tempo e il filo del discorso con le interruzioni dovute al loro caricamento;
- pezzi contenenti argomenti diversi portano a collegamenti generici e poco chiari;
- se in rete, file troppo corti sono più difficili da trovare con un motore di ricerca.

È importante:

- avere indici e richiami locali;
- fornire ausili per navigare senza perdersi e ripetersi (i lettori possono avere stili di consultazione diversi);
- fare riferimenti a pagine in rete, se opportuno;
- avere componenti di qualità omogenea e struttura coerente;
- se mettiamo l'ipertesto in rete, fare in modo da essere trovati facilmente dai motori di ricerca (servono le parole chiave giuste all'inizio del testo).

Impariamo a strutturare i nostri ipertesti osservando quelli degli altri e valutando se ci sembrano soddisfacenti come organizzazione, presentazione, collegamenti.

Nei file è possibile inserire immagini, suoni, animazioni. Le immagini richiedono più tempo dei testi per essere caricate in memoria:

- abbelliscono un documento rendendolo gradevole;
- aumentano la dimensione dell'ipertesto;
- ne rallentano la visualizzazione, scoraggiando l'utente;
- limitiamo il loro uso a quanto sono necessarie;
- occorre distinguere tra elementi (figure, suoni, animazioni) che abbelliscono l'interfaccia ed elementi che trasmettono un contenuto.

2 Il World Wide Web

Un insieme di documenti collegati fra loro e sparsi su milioni di elaboratori forma un ipertesto (hypertext), cioè un testo che viene percorso in modo non lineare. Il Web ha diverse caratteristiche che hanno contribuito al suo enorme successo:

- architettura di tipo client-server (vedi seguito):
  - ampia scalabilità;

  - adatta ad ambienti di rete;

- architettura distribuita:
  - perfettamente in linea con le esigenze di gestione di un ipertesto;

- architettura basata su standard di pubblico dominio:
  - possibilità per chiunque di proporre una implementazione;
  - uguali possibilità di accesso per tutte le piattaforme di calcolo;

- capacità di gestire informazioni di diverso tipo (testo, immagini, suoni, filmati, realtà virtuale, ecc.):
  - grande interesse da parte di tutti gli utenti.

I documenti che costituiscono l'ipertesto gestito dal Web sono detti pagine Web, e possono contenere, oltre a normale testo formattato, anche:

- rimandi (detti link o hyperlink) ad altre pagine Web;
- immagini fisse o in movimento;
- suoni;
- scenari tridimensionali interattivi;
- codice eseguibile localmente.

L'utilizzo del Web è semplicissimo:

- un utente legge il testo della pagina, vede le immagini, ascolta la musica, ecc.;
- se seleziona col mouse un link (che di solito appare come una parola sottolineata e di diverso colore) la pagina di partenza viene sostituita sullo schermo da quella relativa al link selezionato.

Si noti che la nuova pagina può provenire da qualunque parte del pianeta.

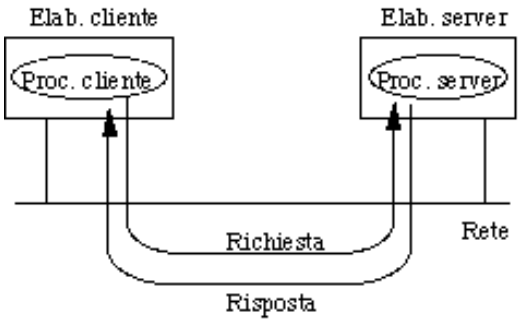

Figura 1. *Architettura client-server*

2.1 Architettura client-server del Web

Il Web è una architettura software di tipo client-server (vedi Fig. 1 , nella quale sono previste due tipologie di componenti software: il client e il server, ciascuno avente compiti ben definiti.

Client

Il client (o user agent) è lo strumento a disposizione dell'utente che gli permette l'accesso e la navigazione nell'ipertesto del Web. Esso ha varie competenze:

- trasmettere all'opportuno server le richieste di reperimento dati che derivano dalle azioni dell'utente;
- ricevere dal server le informazioni richieste;
- visualizzare il contenuto della pagina Web richiesta dall'utente, gestendo in modo appropriato tutte le tipologie di informazioni in esse contenute;
- consentire operazioni locali sulle informazioni ricevute (ad esempio salvarle su disco, stamparle).

I client vengono comunemente chiamati browser (sfogliatori). Gli esempi più noti di browser grafici sono:

- NCSA Mosaic (il primo);
- Netscape Navigator;
- Microsoft Internet Explorer.

In generale è troppo complicato e costoso (sarebbero necessari aggiornamenti troppo frequenti) sviluppare un browser che sappia gestire direttamente tutti i tipi di informazioni presenti sul Web, poiché essi sono in continuo e rapido aumento. Per questa ragione, di norma i browser gestiscono direttamente solo alcune tipologie di informazioni, quali:

- testo formattato;
- immagini fisse;
- codice eseguibile.

Viceversa, di norma gli altri tipi di informazioni vengono gestiti in uno (o entrambi) dei seguenti modi:

- consegnandoli a un programma esterno (helper) che provvederà alla corretta gestione (ad esempio, un file contenente un filmato verrà consegnato a un programma per il playback di filmati);
- se il browser ha un'architettura modulare le sue funzionalità possono essere estese per mezzo di plug-in, ossia librerie di codice eseguibile specializzato che possono essere caricate in memoria secondo le necessità. In questa situazione, se il necessario plug-in è installato, il browser provvede a caricarlo e gli affida la gestione delle informazioni da trattare.

Esistono anche browser di testo come Lynx che ignorano tutti i riferimenti a immagini, suoni, video e colori. Utilizzare un browser di testo può essere utile per caricare molto più rapidamente le pagine web ed accedere ad informazioni.

Esistono anche browser audio e braille per permettere l'utilizzo di Internet anche alle persone non vedenti ed i nuovi standard per la realizzazione di documenti web tengono in considerazione queste necessità.

Figura 2. *Menu e status bar di Netscape*

Figura 3. *Menu e status bar di Explorer*

Server

Il server è tipicamente un processo in esecuzione su un elaboratore. Esso, di norma, è sempre in esecuzione (tranne che in situazioni eccezionali) ed ha delle incombenze molto semplici, almeno in linea di principio. Infatti deve:

- rimanere in ascolto di richieste da parte dei client;
- fare del suo meglio per soddisfare ogni richiesta che arriva:
  - se possibile, consegnare il documento richiesto;
  - altrimenti, spedire un messaggio di notifica di errore (documento non esistente, documento protetto, ecc.).

Nonostante la apparente semplicità di tale compito, la realizzazione di un server non è banale, perché:

- deve fare il suo lavoro nel modo più efficiente possibile, dunque deve essere implementato con un occhio di riguardo alle prestazioni;

- deve essere in grado di gestire molte richieste contemporaneamente, e mentre fa questo deve continuare a rimanere in ascolto di nuove richieste.

Il secondo requisito in particolare implica una qualche forma di concorrenza nel lavoro del server. Essa si può ottenere in vari modi, anche in funzione delle caratteristiche del sistema operativo sottostante. Le due tecniche più diffuse sono descritte nel seguito.

3 Standard utilizzati nel Web

Ci sono tre standard principali che, nel loro insieme, costituiscono l'architettura software del Web:

- sistema di indirizzamento basato su Uniform Resource Locator (URL): è un meccanismo standard per fare riferimento alle entità indirizzabili (risorse) nel Web, che possono essere:
  - documenti (testo, immagini, suoni, ecc.);
  - programmi eseguibili (vedremo poi);
- linguaggio HTML (HyperText Markup Language): è il linguaggio per la definizione delle pagine Web;
- protocollo HTTP (HyperText Transfer Protocol): è il protocollo che i client e i server utilizzano per comunicare.

3.1 URL

Una URL costituisce un riferimento a una qualunque risorsa accessibile nel Web. Tale risorsa ovviamente risiede da qualche parte, ed è in generale possibile accedervi in vari modi. Dunque, una URL deve essere in grado di indicare:

- come si vuole accedere alla risorsa;
- dove è fisicamente localizzata la risorsa;
- come è identificata la risorsa.

http	protocollo nativo del Web
ftp	file transfer protocol
news	protocollo per l'accesso ai gruppi di discussione
gopher	vecchio protocollo per il reperimento di informazioni; concettualmente simile al Web, gestisce solo testo
mailto	usato per spedire posta
telnet	protocollo di terminale virtuale, per effettuare login remoti
file	accesso a documenti locali

Tabella 1. *Metodi di accesso a risorse*

Per queste ragioni, una URL è fatta di 3 parti, che specificano:

- il metodo di accesso;
- l'host che detiene la risorsa;
- l'identità della risorsa.

Un tipico esempio di una URL è:

`http://somewhere.net/products/index.html`

nella quale:

- `http://` è il metodo di accesso
- `somewhere.net` è il nome dell'host
- `/products/index.html` è l'identità della risorsa

Metodo di accesso

Indica il modo di accedere alla risorsa, cioè che tipo di protocollo bisogna usare per colloquiare col server che controlla la risorsa. I metodi di accesso più comuni sono mostrati in Tab. 1.

Il Web nasce con l'idea di inglobare gli altri protocolli di accesso alle informazioni, per costituire un ambiente unificato che soddisfa tutte le esigenze. Quando il client effettua la richiesta di una risorsa, usa nel dialogo col server il protocollo specificato dal metodo d'accesso. Se non è in grado di farlo, affida il compito a una applicazione helper esterna (questo è tipicamente il caso del protocollo telnet: il client lancia un emulatore di terminale passandogli il nome dell'host). Dall'altra parte risponde il server di competenza, che può essere:

- un server Web in grado di gestire anche altri protocolli;
- un server preesistente per lo specifico protocollo (ftp, gopher, ecc.).

Nome dell'host

Può essere l'indirizzo IP numerico o, più comunemente, il nome DNS dell'host a cui si vuole chiedere la risorsa.

Identità della risorsa

Consiste, nella sua forma più completa, della specifica del nome di un file e del cammino che porta al direttorio in cui si trova.

Ad esempio, la URL:

`http://somewhere.net/products/toasters/index.html`

specifica il file index.html contenuto nel direttorio toasters, a sua volta contenuto nel direttorio products il quale si trova nel direttorio radice dell'host somewhere.net. Si noti che:

- la sintassi è quella di Unix;
- il direttorio radice è relativo all'albero dei documenti Web, e non è necessariamente la radice dell'intero file system dell'elaboratore;
- ciò fa s che sia di fatto impossibile accedere per mezzo del Web al di fuori di tale parte del file system: il server, di norma, non consente di accedere a nulla che non sia nell'albero dei documenti Web.

Esistono alcune regole per il completamento di URL non interamente specificate:

- se manca il nome del direttorio, si assume quello della pagina precedente;
- se manca il nome del file (ma c'è quello del direttorio), a seconda del server:
  - si restituisce un file prefissato del direttorio specificato (index.html, default.html oppure welcome.html);
  - se tale file non esiste, talvolta si restituisce un elenco dei file nel direttorio.

Infine, una convenzione usata spesso è la seguente. A fronte di una URL del tipo:

`http://somewhere.net/ username/`

il server restituisce il file welcome.html situato nel direttorio `public_html` situato nel direttorio principale (home directory) dell'utente username. Questo meccanismo consente agli utenti, che di norma hanno libero accesso al proprio home directory, di mantenere facilmente proprie pagine Web.

3.2 Il protocollo HTTP

Il protocollo HTTP sovraintende al dialogo fra un client e un server web, ed è il linguaggio nativo del Web. Il protocollo prevede che ogni singola interazione fra client e server si svolga secondo il seguente schema:

- apertura di una connessione di livello transport fra client e server (TCP è lo standard di fatto, ma qualunque altro può essere usato);
- invio di una singola richiesta da parte del client, che specifica la URL desiderata;
- invio di una risposta da parte del server e dei dati di cui alla URL richiesta;
- chiusura della connessione di livello transport.

Ogni singola interazione è storia a se ed è del tutto indipendente dalle altre.

GET	Richiesta di ricevere un oggetto dal server
HEAD	Richiesta di ricevere la sola parte head di una pagina html
PUT	Richiesta di mandare un oggetto al server
POST	Richiesta di appendere sul server un oggetto a un altro (vedremo che si usa molto)
DELETE	Richiesta di cancellare sul server un oggetto
LINK e UNLINK	Richieste di stabilire o eliminare collegamenti fra oggetti del server

Tabella 2. *Metodi del protocollo HTTP*

La richiesta del client

Quando un client effettua una richiesta invia diverse informazioni:

- il metodo (cioè il comando) che si chiede al server di eseguire;
- il numero di versione del protocollo HTTP in uso;
- l'indicazione dell'oggetto al quale applicare il comando;
- varie altre informazioni, fra cui:
  - il tipo di client;
  - i tipi di dati che il client può accettare.

I metodi definiti in HTTP sono mostrati in Tab. 2. In proposito, si noti che:

- il metodo che si usa quasi sempre è GET;
- POST ha il suo più significativo utilizzo in relazione all'invio di dati tramite form, come vedremo in seguito;
- HEAD si usa quando il client vuole avere delle informazioni per decidere se richiedere o no la pagina;
- PUT, DELETE, LINK, UNLINK non sono di norma disponibili per un client, tranne che in quei casi in cui l'utente sia abilitato alla configurazione remota (via Web) del server Web.

Ad esempio, supponiamo che nel file HTML visualizzato sul client vi sia un'ancora:

```
<A HREF="http://somewhere.net/index.html"> ..... </A>
```

e che l'utente attivi tale link. A tal punto il client:

- chiede al DNS l'indirizzo IP di somewhere.net;
- apre una connessione TCP con somewhere.net, port 80;
- invia la sua richiesta.

Essa è costituita da un insieme di comandi (uno per ogni linea di testo) terminati con una linea vuota:

```
GET /index.html HTTP/1.0          Metodo, URL e versione
                                  protocollo
User-agent: Mozilla/3.0           Tipo del client
Host: 160.10.5.43                 Indirizzo IP del client
Accept: text/html                 Client accetta pagine HTML
Accept: image/gif                 Client accetta immagini

Accept: application/octet-stream  Client accetta file binari
                                  qualunque
If-modified-since: data e ora     Inviare il documento solo se
                                  e‘ piu‘ recente della data
                                  specificata
```

La risposta del server

La risposta del server è articolata in più parti, perché c'è un problema di fondo: come farà il client a sapere in che modo dovrà gestire le informazioni che gli arriveranno? Ovviamente, non si può mostrare sotto forma di testo un'immagine o un file sonoro! Dunque, si deve informare il client sulla natura dei dati che gli arriveranno prima di iniziare a spedirglieli. Per questo motivo la risposta consiste di 3 parti:

- una riga di stato, che indica quale esito ha avuto la richiesta (tutto ok, errore, ecc.);

Esito	**Codice numerico**	**Specifica testuale**
Tutto ok	200	OK
Documento spostato	301	Moved permanently
Richiesta di autenticazione	401	Unauthorized
Richiesta di pagamento	402	Payment required
Accesso vietato	403	Forbidden
Documento non esistente	404	Not found
Errore nel server	500	Server error

Tabella 3. *Codici di risposta del server*

- delle metainformazioni che descrivono la natura delle informazioni che seguono;
- le informazioni vere e proprie (ossia, l'oggetto richiesto).

La riga di stato, a sua volta, consiste di tre parti:

- Versione del protocollo http;
- Codice numerico di stato;
- Specifica testuale dello stato.

Tipici codici di stato sono mostrati in Tab. 3. Dunque, ad esempio, si potrà avere

```
HTTP/1.0 200 OK
```

Le metainformazioni dicono al client ciò che deve sapere per poter gestire correttamente i dati che riceverà. Sono elencate in linee di testo successive alla riga di stato e terminano con una linea vuota. Tipiche metainformazioni sono mostrate in Tab. 4.

Il Content-type si specifica usando lo standard MIME (Multipurpose Internet Mail Exchange), nato originariamente per estendere la funzionalità della posta elettronica. Un tipo MIME è specificato da una coppia

MIME type/MIME subtype

Server:	Identifica il tipo di server
Date:	Data e ora della risposta
Content-type:	Tipo dell'oggetto inviato
Content-length:	Numero di byte dell'oggetto inviato
Content-language:	Linguaggio delle informazioni
Last-modified:	Data e ora di ultima modifica
Content-encoding:	Tipo di decodifica per ottenere il content

Tabella 4. *Metainformazioni del protocollo HTTP*

Type/Subtype	**Estensione**	**Tipologia delle informazioni**
text/plain	.txt,.java	testo
text/html	.html,.htm	pagine html
image/gif	.gif	immagini gif
image/jpeg	.jpeg,.jpg	immagini jpeg
audio/basic	.au	suoni
video/mpeg	.mpeg	filmati
application/octet-stream	.class, .cla, .exe	programmi eseguibili
application/postscript	.ps	documenti Postscript
x-world/x-vrml	.vrml, .wrl	scenari 3D

Tabella 5. *MIME type*

Vari tipi MIME sono definiti, e molti altri continuano ad aggiungersi. I più comuni sono riportati in Tab. 5.

Il server viene configurato associando alle varie estensioni i corrispondenti tipi MIME. Quando gli viene chiesto un file, deduce dall'estensione e dalla propria configurazione il tipo MIME che deve comunicare al client. Se la corrispondenza non è nota, si usa quella di default (tipicamente text/html), il che può causare errori in fase di visualizzazione. Anche la configurazione del client (in merito alle applicazioni helper) si fa sulla base dei tipi MIME. Tornando al nostro esempio, una richiesta del client quale:

```
GET /index.html HTTP/1.0
```

```
User-agent: Mozilla/3.0

ecc.
```

riceverà come risposta dal server (supponendo che non ci siano errori) le metainformazioni, poi una riga vuota e quindi il contenuto del documento (in questo caso una pagina HTML costituita di 6528 byte):

```
HTTP/1.0 200 OK
Server: NCSA/1.4
Date: Tue, july 4, 1996 19:17:05 GMT
Content-type: text/html
Content-length: 6528
Content-language: en
Last-modified: Mon, july 3, 1996 15:05:35 GMT

                    <----- notare la riga vuota

<HTML>
<HEAD>
...
<TITLE>...</TITLE>
...
</HEAD>
<BODY>
...
</BODY>
</HTML>
```

- il protocollo HTTP è molto semplice, essendo basato su interazioni che prevedono esclusivamente l'invio di una singola richiesta e la ricezione della relativa risposta;
- questa semplicità è insieme un punto di forza e di debolezza:
  - di forza perché è facilissimo, attraverso la definizione di nuovi tipi MIME e di corrispondenti funzioni sui client, estendere le tipologie di informazioni gestibili (il server non entra nel merito di ciò che

contengono i file: si limita a consegnare i dati che gli vengono richiesti, senza preoccuparsi della loro semantica);

– di debolezza perché queste estensioni di funzionalità talvolta mal si adattano alla concezione originaria (stateless) del protocollo, come ad esempio è il caso delle transazioni commerciali.

4 Motori di ricerca

Esistono motori di ricerca di due diversi tipi:

- indice;
- elenco (directory).

Nei motori di tipo indice, un programma automatico visita in continuazione la rete, a partire da un dato insieme di indirizzi e, seguendo tutti i possibili collegamenti, compila un indice delle parole trovate nei documenti visitati (esclusi articoli, connettivi, ausiliari, ecc). Alcuni listano tutte le parole, altri solo le più ricorrenti, altri solo quelle nelle prime righe della pagina Quando facciamo una ricerca, la valutazione della pertinenza di un sito viene fatta in base al numero di volte e alla posizione in cui compaiono le parole cercate. Per questo motivo otteniamo spesso più indicazioni di quante possiamo consultarne e tra queste non sempre quello che cerchiamo

Nei motori di tipo elenco, un gruppo di esperti umani visita la rete e classifica i siti in base a un elenco di categorie; quando una categoria diventa troppo numerosa viene divisa in sottocategorie

- gli indici sono più adatti quando abbiamo parole chiave da cercare;
- gli elenchi se cerchiamo su un argomento.

Motori più ampiamente usati (perché molto ricchi)

- indici: Altavista (ricerca in diverse lingue, incluso l'italiano);
- elenchi: Yahoo! , Arianna, Virgilio.

Osservazioni:

- nessun motore ha mai visitato tutta la rete! ma solo dal 30% al 50%;

- non sempre gli indirizzi forniti sono aggiornati di recente;
- motori diversi danno risultati diversi (dipende da quali siti il motore ha visitato);
- i siti vengono scelti dai motori di ricerca mediante link o per segnalazione degli autori.

Se si cerca qualcosa abbastanza comune, è meglio usare un motore più piccolo, perché dà risultati meno dispersivi; argomenti insoliti è più facile trovarli con motori più grandi. Criteri guida per cercare nel www:

- scegliere con cura le parole chiave da cercare (parole generiche come computer producono documenti generici in gran quantità, e per di più sono scartate da alcuni motori nelle ore di punta);
- usare più parole chiave (i guru della ricerca in rete consigliano fino a 5 - 8): questo serve a vincolare meglio la scelta, evitando documenti generici;
- definire una relazione tra le parole chiave mediante connettori logici:
  - chiave1 AND chiave2 significa che il documento deve contenere entrambe le parole chiave; si può anche scrivere +;
  - chiave1 OR chiave2 significa che deve contenere almeno una delle due; è il modo assunto per difetto da molti motori quando l'utente non specifica connettori (ma non da tutti!); serve per specificare sinonimi;
  - "chiave1 chiave2": gli apici indicano che vogliamo esattamente l'espressione tra apici;
  - usare * in sostituzione di parti di stringhe per includere varianti delle parole chiave;
- attenzione agli errori di battitura (i motori non sono intelligenti!);
- attenzione all'uso delle maiuscole (i motori in genere usano caratteri minuscoli, a parte le iniziali dei nomi propri);
- eventualmente usare motori specializzati su qualche argomento o servizi di ricerca multipla.

Tutorial in rete:

- `http://www.thewebtools.com/tutorial/part1.htm`
- `http://www.motoridiricerca.com/`
- `http://www.aib.it/aib/lis/motori.htm`
- `http://www.altavista.wwmind.com/education2/guida`
- `http://www.pcself.com/motori.html`
- `http://www.html.it/dossier/02_motori/index.html`

Esempio: cerchiamo la pagina della European School of Economics

- se cerco *european school of economics* trovo i file contenenti la parola *european* e/o la parola *school* e/o la parola *economics*;
- se cerco *european + school + economics* trovo i file contenenti tutte e tre le parole;
- se cerco "*european school of economics*" trovo i file contenenti l'espressione *european school of economics.*

Il numero di documenti trovati può variare sensibilmente, soprattutto quando le parole chiave sono parole di uso comune Prima di visitare i siti trovati dal motore, leggiamo le due righe esplicative riportate sotto gli indirizzi per guidare la scelta Aiutiamoci anche guardando l'indirizzo web, che spesso dà un'idea sul possibile contenuto di un sito

Una volta trovato il sito che cercavamo, lo visitiamo ed è possibile:

- aggiungere un segnalibro (bookmark o preferiti) per ritrovarlo agevolmente;
- stampare;
- salvare come file;
- aprire il sorgente html;
- copiare una figura (attenzione ai problemi di copyright!);
- aprire i collegamenti proposti (attenzione a non perdersi!);

- mettere un segnalibro per un link proposto senza aprirlo;
- aprire una nuova finestra per visualizzare un'altra pagina;
- ordinare i segnalibri.