

Il World Wide Web



Moreno Marzolla
INFN Sezione di Padova
moreno.marzolla@pd.infn.it
<http://www.dsi.unive.it/~marzolla>

Ringraziamenti

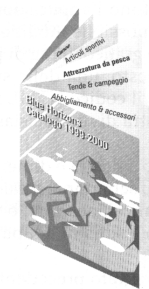
- Parte del materiale presentato è tratto dal Corso di Tecnologie Web, prof. Fabio Vitali, Università di Bologna

Gli ipertesti

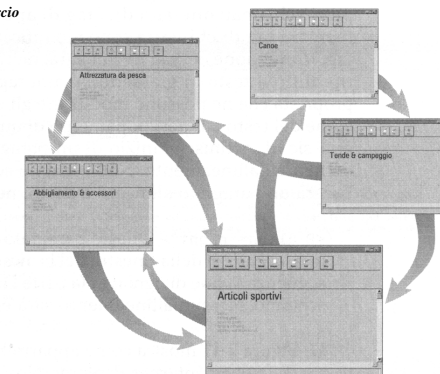
- L'esigenza di evidenziare visivamente riferimenti e connessioni implicite od esplicite tra testi esiste da sempre.
- Era solo inevitabile che la meccanizzazione prima, e l'informatizzazione dopo, cercasse meccanismi per migliorare quest'esigenza, ed automatizzarne la fruizione.

Ipertesti

Fonte: Schneider e Perry, *Commercio elettronico*, Apogeo



Leggere un documento lineare



Leggere un documento ipertestuale

Vannevar Bush e il Memex

- Negli anni quaranta, Vannevar Bush (consulente scientifico alla presidenza degli Stati Uniti) ipotizzò e iniziò a progettare un sistema elettromeccanico basato su microfilm per la memorizzazione e interconnessione di tutte le carte, libri ed informazioni utili
 - Vannevar Bush, "As We May Think", *The Atlantic Monthly*, July 1945
- Il sistema (chiamato *Memex*) era basato su un meccanismo di fotografia e microfilmazione automatica di fogli, sulla possibilità di punzonare secondo codici prestabiliti i microfilm, e su un meccanismo di ricerca rapida di microfilm sulla base di queste punzonature.
- Esso era ovviamente troppo lento per un'utilizzo pratico, e l'elettronica rese obsoleta la tecnologia meccanica. Tuttavia, erano presenti tutte le idee degli ipertesti.



Theodore Nelson e Xanadu

- Negli anni sessanta Ted Nelson, visionario e futuro autore di uno dei libri più importanti per il successo dei personal computer, *Computer Lib*, iniziò a progettare e cercare fondi per realizzare un sistema integrato di gestione della letteratura, ovvero dei contenuti, dei riferimenti espliciti ed impliciti, e del processo che sostiene la produzione letteraria (creazione, pubblicazione, modello economico).
- Il sistema (chiamato *Xanadu*), come evoluto alla metà degli anni Ottanta, era basato su server che costruivano documenti virtuali basati su pezzi di testo di lunghezza arbitraria, e client che permettevano la creazione di link, la modifica dei documenti e possedevano un sistema efficace di gestione monetaria per i documenti a pagamento.
- Il sistema Xanadu non è mai esistito, anche se ci andò vicino alla fine degli anni 80. Su di esso si ispirò Tim Berners-Lee per il World Wide Web.

Douglas Engelbart e Augment

- Nella fine degli anni sessanta Doug Engelbart iniziò a lavorare sul concetto di personal computing, utilizzando costose workstation IBM per realizzare un sistema di video-conferenza, editing di testi gerarchici (outline processor) ed ipertestuali e di supporto per il lavoro cooperativo dotato di interfaccia a finestre, mouse e altri meccanismi rivoluzionari di input ed output.
- Esiste un video del 1967 dove si vedono Engelbart ed il suo team usare queste tecnologie che sarebbero diventate d'uso comune 15 anni dopo almeno!
- L'idea di Engelbart era che il supporto di caratteristiche innovative potesse aumentare il potenziale intellettuale degli uomini, e che l'evidente difficoltà tecnica fosse un ostacolo superabile con un training adeguato.



Storia del WWW / 1

- Nel 1989, un gruppo di ricercatori informatici del CERN (il centro di ricerca in fisica nucleare di Ginevra) ricevette l'incarico da parte della direzione di realizzare un meccanismo per la diffusione rapida di articoli, appunti e opinioni tra i fisici che ruotavano intorno al centro.
- Tim Berners-Lee, Robert Cailliau ed altri identificarono Internet, ipertesti e SGML come elementi chiave per questo meccanismo.
- Nel 1991, alla conferenza sugli ipertesti, Berners-Lee e Cailliau mostrarono (con poco successo) il primo prototipo della loro applicazione, realizzata in client-server su architettura NeXT: World-Wide Web.
- Il prototipo NeXT era composto di:
 - Un server che spediva documenti memorizzati localmente a chiunque lo richiedesse secondo il protocollo stabilito, e che memorizzava documenti spediti da remoto, il tutto senza autorizzazione o verifica
 - Un editor di testi parzialmente WYSIWYG che permetteva di visualizzare documenti ipertestuali e di modificarli, creando link e blocchi di testo.

Storia del WWW / 2

- I principali concorrenti su Internet erano:
 - **FTP**, che forniva un meccanismo di scambio di file
 - **WAIS**, che era un server con notevoli potenzialità di ricerca su documenti di solo testo
 - **Gopher**, che forniva un meccanismo di organizzazione di documenti di testo in gerarchie distribuite su più server
- I principali concorrenti nel campo degli ipertesti erano:
 - **Hypercard** (e Toolbook): strumenti di produttività individuale, a metà strada tra il database, il programma di disegno, l'ambiente di fast prototyping e l'ipertesto. Non distribuito, era un'applicazione monolitica.
 - **Microcosm**: un semplice server su rete locale, con identificazione degli utenti, e una serie di moduli e modifiche (hack) su applicazioni comuni (AutoCAD, MS Word ...) per permettere la ricerca ed visione dei link.
 - **Hyper-G**: un sistema client-server complesso e completo, con un protocollo di comunicazione sofisticato, un modello di dati completo, ma senza il concetto di indirizzamento completo.

Storia del WWW / 3

- Nonostante l'accoglienza fredda dei ricercatori sugli ipertesti, i fisici furono entusiasti del WWW: comodo, facile da usare e da imparare, gratuito, privo di alternative realistiche (Hyper-G è di un anno più tardi).
- Nell'ottobre del 1992 il National Centre for Supercomputing Applications (NCSA) esaminò il prototipo di WWW e decise di realizzarne una versione propria.
- Con la realizzazione del server NCSA e del primo browser WWW, chiamato Mosaic, l'NCSA decretò l'inizio del successo esplosivo del sistema.

Storia del WWW / 4

- Mosaic non aveva possibilità di editing come nel prototipo NeXT, ma poteva visualizzare semplici immagini GIF
- Milioni di utenti iniziarono ad usare Mosaic o altri browser WWW, e altri installarono server HTTP. Nel frattempo, però, l'NCSA aveva deciso di non spendere più energie sul WWW.
- Marc Andreessen, realizzatore del prototipo di Mosaic su X-Windows, e capo del gruppo alla NCSA che aveva realizzato le altre versioni ed il server, si trovò la strada bloccata per l'evoluzione del progetto.
- Jim Clark, ex professore a Berkeley e co-fondatore della Silicon Graphics, standard nella grafica professionale (animazioni, effetti speciali, progettazione, ecc.) cercava una nuova idea da finanziare.

Storia del WWW / 5

- Prendendo Andreessen come condirettore, Clark fonda nel 1993 la Mosaic Corporation, poi rinominata Netscape Corporation
- Il successo di Netscape Navigator (1994) è immediato e totale. La ditta Netscape ha il passaggio più rapido tra la fondazione e la quotazione in borsa della storia, ed una delle quotazioni iniziali di maggior successo.
- Ma fin dall'inizio Clark e soci sanno di essere o di star per entrare nel mirino di Microsoft, e si preoccupano di mantenere competitività e controllo del mercato.

Storia del WWW / 6

- Nel frattempo, Berners-Lee e Cailliau cercano di mantenere il controllo sull'evoluzione del World Wide Web e fondano il W3C, con fondi della ricerca e dell'università.
- Microsoft, dopo una falsa partenza con Microsoft Network, abbraccia definitivamente la tecnologia Internet, ed inizia a realizzare un browser (Internet Explorer) ed un server HTTP (Microsoft Information Server)
- La lotta è senza esclusione di colpi, incluso il bundle di IE in Windows 95, l'integrazione del browser come interfaccia di Windows in Second Edition, la causa di concorrenza sleale all'antitrust, ecc.

Storia del WWW / 7

- Nel marzo 98 Netscape chiede l'aiuto della comunità dei programmatori rilasciando il codice sorgente della versione 5 di Navigator. Nasce il progetto mozilla.org, per sfruttare gli stessi principi di organizzazione della comunità Linux per un grande browser freeware. Tanti progetti nascono, ma le nuove versioni stabili ed utilizzabili non arrivano.
- A novembre del 98 Netscape viene comprata da America On Line, Clark e Andreessen lasciano (multimiliardari).
- A marzo 2000 viene annunciata la versione 6 di Communicator. Solo nel giugno 2001 vedrà la luce la versione definitiva. La velocità e frenesia degli anni d'oro è conclusa. Altri giocatori ci provano (es. Opera), ma occupano un ruolo tutto sommato minore nello sviluppo delle funzionalità dei browser.

Storia del WWW / 8

- All'inizio del 2003 Apple lancia Safari, un browser velocissimo ma limitato nelle tecnologie.
- Poco dopo, Microsoft annuncia che smetterà di supportare IE per Macintosh (fermo alla versione 5.2), in quanto esistono già altri browser per questo computer.
- Anche Opera ha annunciato di voler ritirarsi dal mercato Macintosh.

Storia del WWW / 9

- Avevamo lasciato nel 2000 Netscape e Mozilla a dibattersi verso release che non arrivano e se arrivano non funzionano.
- Netscape 6 (2001) è uno dei browser più brutti mai visti in circolazione. Tre quarti del team di produzione viene licenziato dopo l'uscita del browser, e AOL annuncia di volersi esplicitamente ricondurre alle versioni di Mozilla pubblicate da mozilla.org. Per questo motivo finanzia copiosamente la fondazione senza fini di lucro che se ne occupa.
- Mozilla 0.9.7 (praticamente la versione finale della 1.0) esce a gennaio 2002, e a marzo esce Netscape 7.0. Da qui seguono a distanza di 5/6 mesi uscite di nuove sotto-release (anche radicalmente migliorate). E' annunciata recentemente la versione 1.7 di Mozilla.
- Ma a maggio del 2003, AOL e Microsoft annunciano un accordo che prevede AOL distribuire gratuitamente Explorer ai suoi abbonati, rendendo di fatto inutile continuare a supportare anche Netscape.
- Infatti a ottobre AOL licenzia metà dei 40 dipendenti rimasti che si occupavano dei browser, e ha annunciato di non voler più ulteriormente finanziare Mozilla (che ha 10 dipendenti)

Storia del WWW / 10

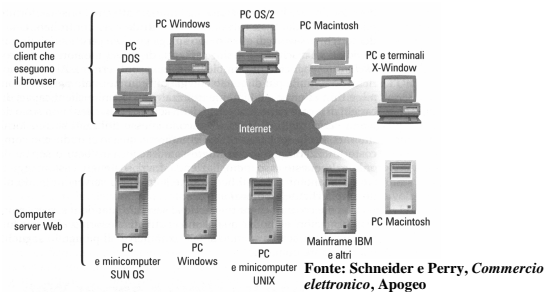
- Nel 1997 nasce XML. Caso unico nella storia dell'informatica, sembra accontentare tutti: fornitori di contenuti, grafici e designer, realizzatori di applicazioni su Web, perfino progettisti di DBMS.
- Tra i supporter principali, Microsoft, Sun, IBM, Oracle, Adobe e tutti i grossi provider di contenuti (AOL-Time Warner e Walt Disney in testa)
- Il 2000 vede anche la saturazione del mercato per le piccole ditte di software che fanno siti web e piccole applicazioni server-side. Il crollo di borsa di tante aziende della new-economy blocca molti sviluppi interessanti ed innovativi.
- Sopravvivono quelli molto grossi, o quelli che fanno applicazioni più tradizionali, anche su Web: gestionali, integrazione di sistemi informativi, applicazioni di database.
- Da due o tre anni gli aspetti creativi e di presentazione di contenuti nel web sembrano ridotti rispetto alle esigenze di fare del web un'infrastruttura per lo scambio di dati tra applicazioni eterogenee
- Si veda ad esempio il supporto per le nuove release di CSS, una volta motivo di vanto di Microsoft, che invece non verranno inserite né a breve né forse mai.

Cos'è il WWW

- Il World Wide Web è un sistema per la presentazione a schermo di documenti multimediali, e per l'utilizzo di link ipertestuali per la navigazione.
- Il sistema è distribuito e scalato su tutta Internet, ed è basato su alcuni semplici concetti:
 - Il client o browser è un visualizzatore di documenti ipertestuali e multimediali. Può visualizzare testi, immagini e semplici interfacce grafiche, ma non permette di editare documenti.
 - Il server è un semplice meccanismo di accesso a risorse locali (file o record di database, ecc.) in grado di trasmettere via socket TCP documenti individuati da un identificatore univoco
 - Il server può collegarsi ad applicazioni server-side (tramite protocollo CGI) ed agire da tramite tra il browser e l'applicazione facendo del browser l'interfaccia dell'applicazione.

Client e server WEB

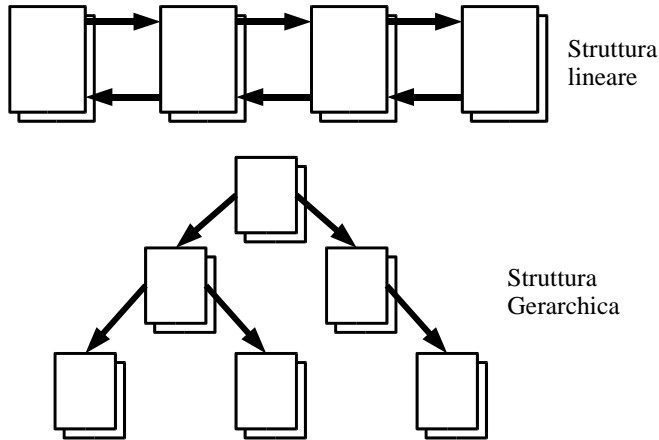
- L'insieme di documenti HTML, residenti su server diversi e collegati da link, costituisce il *World Wide Web*



I protocolli del WWW

- Alla base di WWW ci sono i seguenti protocolli:
 - Uno standard per identificare in maniera generale risorse di rete e per poterle specificare all'interno di documenti ipertestuali (chiamato URI).
 - Un protocollo di comunicazione state-less e client-server per l'accesso a documenti ipertestuali via rete (chiamato HTTP).
 - Un linguaggio per la realizzazione di documenti ipertestuali (chiamato HTML) basato su SGML e incentrato sulla possibilità di realizzare connessioni ipertestuali in linea nella descrizione strutturale del documento.

Strutture di link



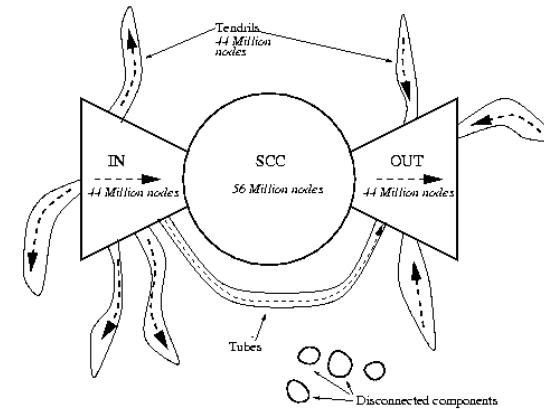
Moreno Marzolla

Tecnologie Web

21

La struttura a link del WEB

<http://www9.org/w9cdrom/160/160.html>



Moreno Marzolla

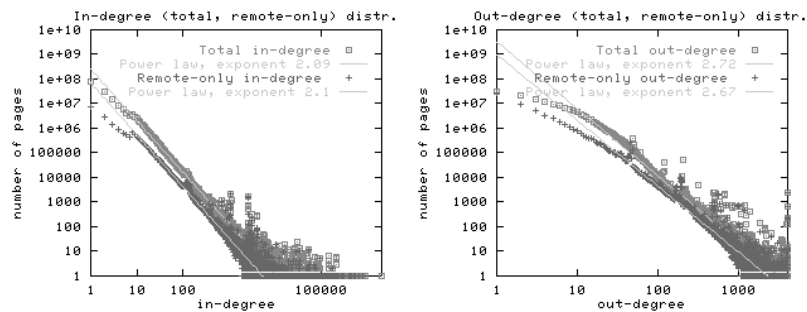
Tecnologie Web

22

Distribuzione dei link in/out

La distribuzione dei link in ingresso segue la "power law"

$$P[X=x] \approx x^{-a}$$



Moreno Marzolla

Tecnologie Web

23

Il concetto di risorsa

- Una risorsa è qualunque struttura che sia oggetto di scambio tra applicazioni all'interno del World Wide Web.
- Anche se molti identificatori fanno riferimento a file memorizzati in un file system gerarchico, questo tipo di risorsa non è né necessario né frequente:
 - Potrebbe essere in un database, e l'URI essere la chiave di ricerca
 - Potrebbe essere il risultato dell'elaborazione di un'applicazione, e l'URI essere i parametri di elaborazione.
 - Potrebbe essere una risorsa non elettronica (un libro, una persona, un pezzo di produzione industriale), e l'URI essere il suo nome Uniforme
 - Potrebbe essere un concetto astratto (la grammatica di un linguaggio)
- Per questo si usa il termine Risorsa, invece che File, e si fornisce una sintassi indipendente dal sistema effettivo di memorizzazione. Mai assumere che si stia lavorando con un file system!

Moreno Marzolla

Tecnologie Web

24

URI

- Gli URI (Uniform Resource Identifier) sono una sintassi usata in WWW per definire i nomi e gli indirizzi di oggetti (risorse) su Internet.
- Questi oggetti sono considerati accessibili tramite l'utilizzo di protocolli esistenti, inventati appositamente, o ancora da inventare.
- Gli URI risolvono il problema di creare un meccanismo ed una sintassi di accesso unificata alle risorse di dati disponibili via rete.

URI

- Gli Uniform Resource Identifier (URI) sono, per definizione:
 - **Uniform Resource Locator (URL)**: una sintassi che contiene informazioni immediatamente utilizzabili per accedere alla risorsa (ad esempio, il suo indirizzo di rete)
 - **Uniform Resource Names (URN)**: una sintassi che permetta una etichettatura permanente e non ripudiabile della risorsa, indipendentemente dal riportare informazioni sull'accesso.
- Nota: il termine URI ha rimpiazzato i termini URL/URN

URI

- Uniform
 - Diversi tipi di identificatori di risorse possono essere usati nello stesso contesto, anche quando sono diversi i meccanismi usati per accedere a tali risorse
- Resource
 - La specifica degli URI non pone vincoli a cosa si intende per "risorsa"
- Identifier
 - Un identificatore serve per distinguere una risorsa da un'altra

Criteri di design degli URI

- Trascrivibili
 - gli URI sono sequenze di caratteri tratti da un set molto limitato (lettere, numeri, alcuni - pochi - caratteri speciali), per permettere la trascrizione degli URI su canali anche non elettronici (un pezzo di carta, un cartellone, ecc.)
- Fornire identificazione, non interazione
 - le operazioni eseguibili sulle risorse ed i protocolli per eseguirli non sono indicati nell'URI, ma nell'interpretazione che degli URI si fa all'interno dello specifico schema.
- Fornire spazi di nomi organizzati gerarchicamente
 - i caratteri ":", "/", "?", e "#" separano ambiti diversi all'interno degli URI. Il primo, obbligatorio, è lo schema di interpretazione del resto. Schemi gerarchici usano il carattere "/" per separare livelli di specificità diversi del nome locale. "?" separa la parte di identificazione della risorsa da qualunque parametro necessario per accedervi, e "#" separa l'identificativo della risorsa da quello del frammento interno a cui si fa riferimento.

Organizzazione degli URI

```
schema : [// authority] path [ ? query] [# fragment]
```

Esempi:

```
ftp://ftp.is.co.za/rfc/rfc1808.txt
http://www.ietf.org/rfc/rfc2396.txt
ldap://[2001:db8::7]/c=GB?objectClass=one
mailto:John.Doe@example.com
news:comp.infosystems.www.servers.unix
tel:+1-816-555-1212
telnet://192.0.2.16:80/
urn:oasis:names:specification:docbook:dtd:xml:4.1.2
```

Componenti degli URI

- Lo schema (negli URL è il protocollo) è identificato da una stringa arbitraria (ma registrata presso IANA) usata come prefisso.
- La presenza esplicita di un'authority individua un'organizzazione gerarchica dello spazio dei nomi a cui sono delegati i nomi (che possono essere ulteriormente delegati). L'autorità è a sua volta divisa in:

```
authority = [userinfo] @] host [ : port ]
```

- La parte userinfo non deve essere presente se lo schema non prevede identificazione personale. La parte host è o un nome di dominio o un indirizzo IP. La port può essere omessa se ci si riferisce ad una well-known port (per http è la porta 80).

Organizzazione degli URI

- path identifica la risorsa all'interno dello spazio di nomi identificato dallo schema e (se esistente) dall'authority. Poiché la presenza di una parte authority evidenzia uno spazio di nomi gerarchico sotto la gestione dell'authority, in questi casi la parte path è divisa in blocchi separati da slash "/", ciascuno dei quali è un componente del path organizzato in gerarchia. In questo caso diventano significativi gli pseudo componenti "." e "..".
- query individua un'ulteriore specificazione della risorsa all'interno dello spazio di nomi identificato dallo schema e dall'URI precedente. Di solito questi sono parametri passati all'URI (un processo) per specificare un risultato dinamico (es. l'output di una query su un motore di ricerca). La parte query è tutto quello che sta dopo il punto interrogativo e prima del carattere hash. Tipicamente ha la forma nome1=valore1&nome2=valore2+in+molte+parole
- fragment individua una risorsa secondaria (una risorsa associata, dipendente o in molti casi un frammento) della risorsa primaria. E' tutta la parte che sta dopo al carattere di hash "#".

HTTP e HTTPS

- I protocolli più usati nel WWW. La sintassi di questo schema è:

```
http://host[:port]/path[?query][#fragment]
https://host[:port]/path[?query][#fragment]
```

dove:

- **host** è l'indirizzo TCP-IP o DNS, dell'host su cui si trova la risorsa
- **port** è la porta a cui il server è in ascolto per le connessioni. Per default, la porta è 80 per HTTP e 443 per HTTPS.
- **path** è un pathname gerarchico (per esempio, un filename parziale) per l'identificazione della risorsa
- **query** è una frase che costituisce l'oggetto di una ricerca sulla risorsa specificata.
- **fragment** è un identificativo di una sottoparte dell'oggetto. La definizione e il ritrovamento di queste sottoparti è a carico del client, e quindi la parte di fragment viene ignorata dal server, che restituisce l'intero oggetto.

FTP

- La sintassi di questo schema è:

```
ftp://[user[:password]@]host[:port]/path
```

dove:

- **User e password** sono utente e password per l'accesso ad un server FTP. La mancanza di user fa partire automaticamente una connessione anonima
- Si tende a scoraggiare l'uso della password nell'URL, in quanto evidente situazione di scarsa sicurezza. Tuttavia lo schema lo prevede come parte facoltativa.
- **Host, port e path** sono l'indirizzo del server, la porta di connessione ed il nome del file dell'oggetto ricercato, come per HTTP. La porta di default è 21.

SMTP e Telnet

SMTP

La sintassi della parte specifica è:

```
mailto:user@host
```

dove

- non esiste il prefisso “//” perché lo schema non è gerarchico
- User e host sono i componenti dell'indirizzo di e-mail del destinatario

Telnet

La sintassi della parte specifica è:

```
telnet:host:port
```

Problemi degli URI / 1

- Opacità degli URI
 - Qual è il formato dati di `http://www.sito.it/a/b.html?`
 - Qual è la nazione della pagina:
`http://www.sito.it/a/b.html?`
- Risposta:
 - **Non è l'URI a doverlo dire.** L'URI rappresenta una stringa opaca da cui non è lecito inferire alcuna proprietà della risorsa corrispondente.
 - HTTP fornisce metodi per capire il formato dati. La parte finale dell'URI *non è attendibile*.
 - Le tecniche di content negotiation di HTTP permettono di associare formati dati diversi allo stesso URI, a seconda delle capabilities dello user agent.
 - Mai inferire, sempre controllare.

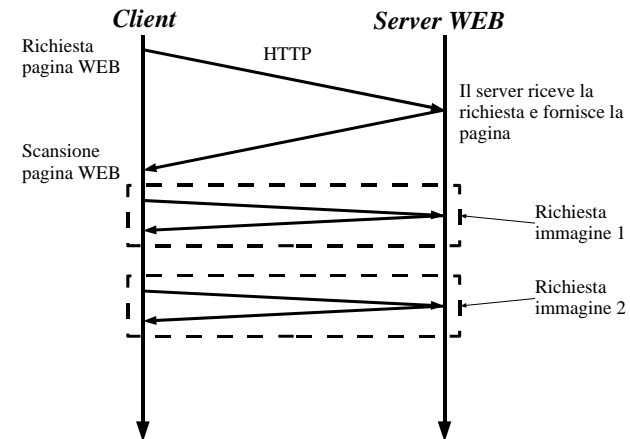
Problemi degli URI / 2

- La lunghezza degli URI
 - `http://it.mappe.yahoo.net/tc/mappa.jsp?com1=pa
dova+%28pd%
29&tol=Via+Francesco+Marzolo&civ1=8&rg=9&pv=pd
&cx=11.8881&cy=45.40915&lx=11.8881&ly=45.40915
&z=0.6&lv=1&ct=&st=&cs=&nm=&al=122481&type=nul
l&d=1`
- Risposta
 - L'URI deve essere il più conciso possibile
 - L'URI deve usare quante più possibile parole comuni, che non vengano sbagliate durante la digitazione
 - Gli URI dovrebbero solo usare lettere minuscole, e mai mescolare i caratteri
 - Gli URI dovrebbero permettere la scomposizione semantica dei contenuti del sito, e non esporre i dettagli tecnici (es., applicazioni CGI o altro).

Comunicazione tra client e server WEB

- La divisione del lavoro tra client e server WEB è molto netta
 - Il vostro PC di casa, con sopra un browser, richiede pagine HTML ad un server e le visualizza
 - Il server WEB riceve richieste di pagine HTML e invia tali pagine a chi le richiede
 - Pagine ricche di immagini fanno sì che il client richieda ogni singolo elemento (immagine) al server, quindi possono risultare lente da caricare

Caricamento delle pagine WEB



Cosa succede quando si carica una pagina WEB / 1

- Il *browser* deve trasformare l'indirizzo del server in forma numerica
 - Viene interpellato un DNS per convertire, ad es., "www.dsi.unive.it" in "157.138.20.20"
- Il *browser* apre una connessione TCP verso il server WEB
- Utilizzando il protocollo HTTP, il *client* richiede la pagina al server
 - Es., /home/it/index.html

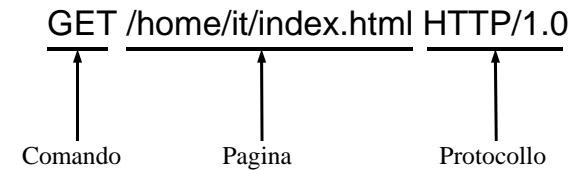
Cosa succede quando si carica una pagina WEB / 2

- Il *server* spedisce al client la pagina richiesta, sempre usando il protocollo HTTP
- La connessione tra client e server viene chiusa
 - Dipende dalla versione di HTTP utilizzata, nel caso di HTTP 1.1 la connessione potrebbe restare aperta
- Il *client* inizia ad interpretare il contenuto della pagina ricevuta

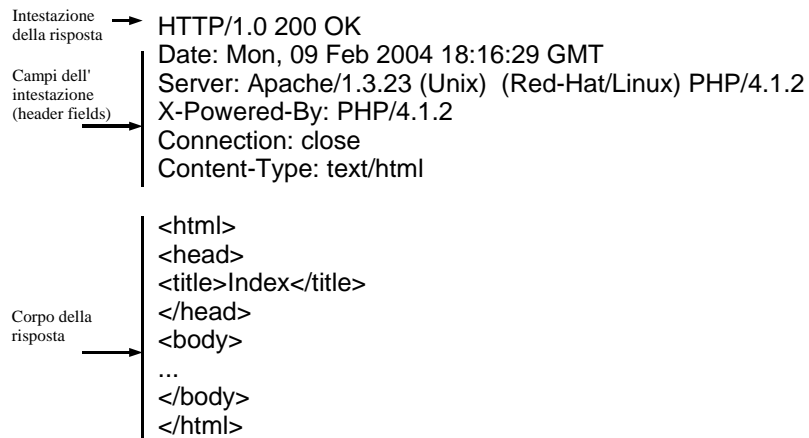
Cosa succede quando si carica una pagina WEB / 3

- Se la pagina contiene immagini, ciascuna immagine deve essere recuperata tramite una richiesta separata al server
- Il browser dispone ora di tutti gli elementi e può visualizzare la pagina

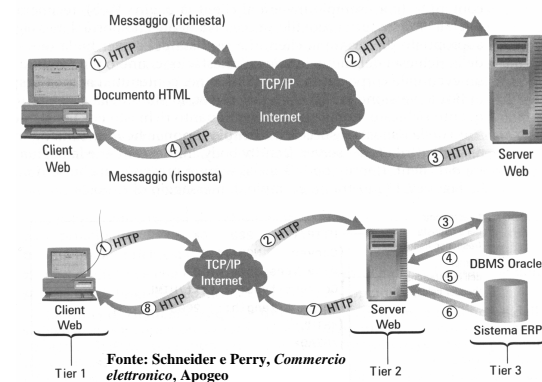
Formato di una richiesta HTTP



Formato di una risposta HTTP



Flusso dei messaggi Architettura a due e tre livelli



Il web oggi

- **Accessibilità ed internazionalizzazione:**
 - possibilità di accedere alle informazioni ed ai servizi da parte di persone handicappate da disabilità fisiche, linguistiche, tecnologiche
- **Mobilità**
 - possibilità di accedere a informazioni e servizi da qualunque parte del mondo, con hardware diverso dal classico PCi: palmari, cellulari, ecc.
- **Semantic Web e Web Services**
 - possibilità di usare le informazioni ed i servizi offerti dal Web non da parte di esseri umani direttamente, ma da parte di applicazioni che fungono da intermediari tra i dati e gli esseri umani, fornendo servizi a valore aggiunto con le informazioni altrui.

Evoluzioni del WWW / 1

- **Inclusione di oggetti:** Mosaic introdusse il supporto per immagini in-line, e Netscape introdusse poi i plug-in per inserire oggetti di tipi diversi nel documento del browser, ed infine le applet Java. IE ha generalizzato le possibilità offerte da COM, realizzando il protocollo proprietario ActiveX.
- **Scripting:** Netscape introdusse LiveScript, poi ribattezzato Javascript, per realizzare semplici applicazioni client-side con linguaggi di scripting appositi. IE rispose con Jscript e Vbscript.
- **Stili:** CSS permette di controllare le caratteristiche dei documenti HTML. XSL si occupa di documenti XML.
- **Server-side includes:** La possibilità di inserire all'interno del documento HTML memorizzato sul server delle istruzioni eseguite dal server stesso prima di spedire il risultato hanno permesso notevoli sofisticazioni nel contenuto (data di oggi, utente registrato, numero di visitatori, piccole computazioni, ecc.)

Evoluzioni del WWW / 2

- **Gestione delle transazioni:** meccanismi per la gestione dello stato sono stati introdotti prima da Netscape, e poi standardizzati (cookies).
- **Siti web dinamici:** I server-side includes e le applicazioni CGI evolvono e si fondono, diventano veri e propri linguaggi o ambienti di programmazione: Perl prima, poi ASP (Javascript e Visual Basic), poi PHP, Python, ecc. Con l'arrivo delle librerie di accesso ai database (ODBC, JDBC) nasce l'architettura a tre livelli (user-interface, application logic, data storage). Il browser diventa l'interfaccia per applicazioni gestionali distribuite.

Future evoluzioni del WWW / 3

- **Modello di link:** il modello di link di HTML è eccessivamente semplice. Xlink ed XPointer permettono di definire link sofisticati, sia per indirizzamento (blocchi di dimensione e locazione arbitraria), sia per funzionamento (inclusione in-line, memorizzazione esterna, multi-direzione, ecc.)
- **Sistemi di interscambio dati:** la diffusione di formati dati estremamente semplici da programmare ed estremamente sofisticati per la generalità, universalità e potenza, sta permettendo di utilizzare modelli e formati Web per l'interscambio di dati. I web service (SOAP, .NET, etc.) sono un meccanismo attraverso il quale realizzare integrazione tra sistemi eterogenei attraverso il Web.
- **Web Semantico:** la possibilità di etichettare i dati con informazioni sulla loro natura, qualità e relazioni con altri dati (meta-informazioni) di trasformerà (?) il Web da machine-readable a machine-understandable, permettendo la nascita di applicazioni molto sofisticate in grado di interpretare e elaborare dati in maniera semi-intelligente.