Web Systems & Technologies: 
An Introduction

Prof. Ing. Andrea Omicini
Ingegneria Due, Università di Bologna a Cesena
andrea.omicini@unibo.it
2005–2006
Web Systems Architecture

- Basic architecture
  - information is structured as hypertext
  - allocation transparency
  - resources as information
- Use of graphical interfaces
  - ease of use
  - uniform access
    - to heterogeneous resources
    - from heterogeneous envs
Perception of Web Systems

- Clicking on a work/image, you can expand a portion of the document we are interested in
- perceiving the fact that the document may / may not be a local one, it is not needed
- Clicking on link which representing a resource in order to access it
- without worrying about the nature of the resource itself
- whatever it is, a doc, a text, a picture, whatever else
World Wide Web (WWW)

CERN (1989)

scenario: ipertextual integration of Internet resources

Goals

access & allocation transparency

usability

multimedial presentation

effectiveness

different protocols, the same interface

interoperability

accessing and sharing information

accessibility

W3C: http://w3c.org
Basic Components: Client-side

- Browsers
  - doing presentation, handling requests
- Helper Applications
  - particular presentations & formats, such videos, sounds, animations
- Applets
  - local execution of Java applications
- Script
  - local execution of small applications written in JavaScript or other similar languages
Basic Components: Server-side

- Web Server
  - managing access control, accepting requests, administering information
- Server-side Applications
  - remote execution
    - CGI, servlet, JSP, PHP, ASP...
Fundamental Standard Specifications & Languages

- Universal Addressing System
  - URI & URL
    - Uniform Resource Identifier/Location
- HTTP Protocol
  - HyperText Transfer Protocol
- HTML / XHTML + CSS
  - (eXtended) HyperText Markup Language
  - Cascading Style Sheets
- CGI
  - Common Gateway Interface
- Java language for Applet, Servlet & JSP
WWW: Base Architecture
Client / Server Connection

- **HTTP Client**
  - client/server pattern toward one HTTP server at a time
  - by specifying an URL (either writing or clicking)
  - HTML page requests via HTTP
  - HTTP response as HTML pages + other contents (images, scripts...)

- **One-shot connection**
  - one different connection per each object
  - e.g.: an HTML page with a JPEG image = 2 HTTP connections
HTTP Connection

![Diagram showing the HTTP connection process between local and remote systems using TCP/IP and HTTP client communication.]

- **Local System**
  - HTTP Client
  - TCP/IP

- **Remote System**
  - HTTP Client
  - TCP/IP

- **Network**
  - Arrows indicating request and response communications.
Uniform Resource Locators

- Unique names for system resources, specified by clients to determine the server
- Uniform Resource Locators (URL)
  - node providing the resource
  - resource access protocol (e.g. http, gopher)
  - TCP port number (service default port)
  - local path of the resource within the server
    - `<protocol>://<host>[:<port>][<path>]`
    - e.g.: http://www.address.edu:1234/path/subdir/file.ext
- Internet services and their protocols are recognised
  - http, gopher, ftp, wais, telnet, news, nntp, e mail (mailto)
- http://www.w3.org/Addressing/
HTTP for Dummies (I)

- HyperText Transfer Protocol
  - client / server interface protocol
  - based on TCP connections
    - default port 80
- HTTP version 1.0
  - Request/response: only data are requested / sent
  - One-shot connection: TCP connection maintained only as long as necessary to send data
  - Stateless: no information is kept by server between two subsequent requests
    - then, information should be kept by clients
HTTP for Dummies (II)

- typical HTTP interaction
  - client request containing information for server (i.e., page local path)
  - server response containing information (i.e., requested page, or error message)
  - some negotiation possible on information and services
    - e.g., give me a page only if changed since my last request
- HTTP version 1.1: some improvements
  - http://www.w3.org/Protocols/
- It will be the subject of future courses, like “Computer Networks” (Reti di calcolatori)
HTML for Dummies (I)

- http://www.w3.org/MarkUp/
- HyperText Markup Language
  - specification language to encode information
  - derived from SGML (Standard Generalized Markup Language)
    - it is a markup language (TeX, RTF)
  - markup languages use tags to add features to enclosed text
- very simple so as not to make clients computationally complex
HTML for Dummies (II)

- **tag HTML: examples**
  - header level 1
    - `<h1>text</h1>`
  - **bold text**
    - `<strong>text</strong>` or `<B>text</B>`
  - browser-dependent visualisation
  - **link**
    - `<a href = "destination"> description </a>`
  - **image**
    - `<img src = "myimage.gif">`
  - **Java applet**
    - `<applet code="Hello.class" width="100" height="80">`
XHTML for Dummies

eXtended HyperText Markup Language

goals
- solve HTML problems
- toward XML
- some backward compatibility toward HTML
  - to avoid migration problems to programmers and tools
- in this course, we mainly deal with XHTML
Web Style Sheets for Dummies

http://www.w3.org/Style/
Style sheets describe how elements of a web page should be represented on a specific medium, screen, audio, paper, etc.
CSS-1 e CSS-2
Cascading Style Sheets for HTML pages
XSL (Extensible Stylesheet Language Family)
for XML sheets
XSL Transformations (XSLT)
XML Path Language (XPath)
XSL Formatting Objects (XSL-FO)
Other Topics

- **JavaScript**
  - [the main block of the course, only for LTI-LA]
  - associating computations to Web pages (and browser events)
  - to be execute by clients (browsers)
- **PHP (maybe)**
  - a simple but powerful interpreted language for server-side computations
# Browsers: the Ancient Times

<table>
<thead>
<tr>
<th>version</th>
<th>browser</th>
<th>properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>historic</td>
<td>header, lists, emph</td>
</tr>
<tr>
<td>2.0</td>
<td>Mosaic</td>
<td>inline images, forms</td>
</tr>
<tr>
<td>2.1</td>
<td>Netscape/Microsoft</td>
<td>tables, alignment</td>
</tr>
<tr>
<td>3.2</td>
<td>Netscape/Microsoft</td>
<td>frames, ...</td>
</tr>
<tr>
<td>4.0</td>
<td>Netscape/Microsoft</td>
<td>styles, JavaScript</td>
</tr>
</tbody>
</table>
Browsers Today...

- Mozilla / Firefox & Company
  - a world-wide project
  - the reference browser engine for this course
  - also for web page construction / verification
    - Composer is fine, Front Page NOT allowed
- Different versions of Internet Explorer
  - bad seeds we should coexist with
- Safari, Opera, Konqueror, ...
  - good
  - however, remember to verify compliance to standards
    - both in theory [they claim to]
    - and in practice [they actually do]