

Lecture 10: Overview of WWW Technologies (Part I)

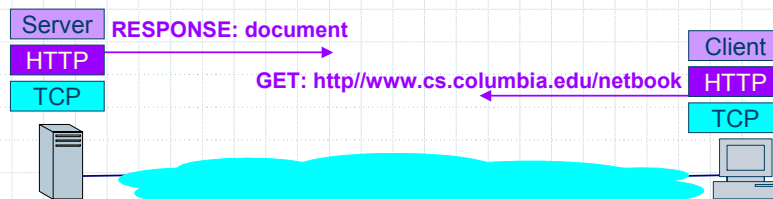
Architecture & operations of WWW
HTML and beyond; client-side and server-side computing

Prof. Yechiam Yemini (YY)
Computer Science Department
Columbia University

(c) Copyrights, Yechiam Yemini, 1993-2003.

Preview

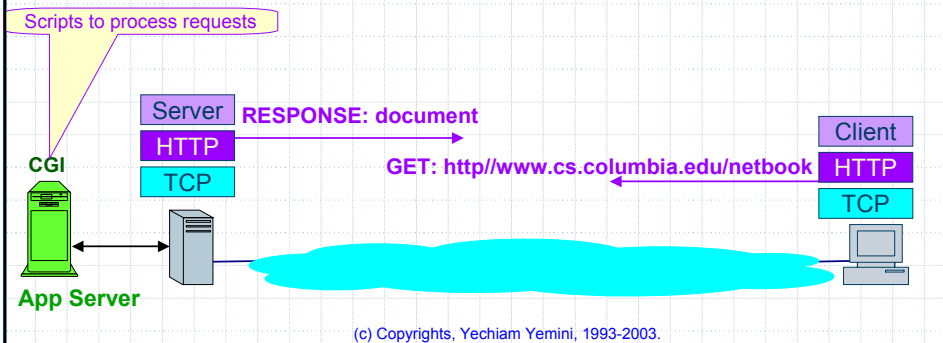
- The Web was created by Tim Breners-Lee in 1989
- It entirely redefined the course of network applications
- Accounts for most Internet traffic; universal deployment
- References: <http://www.w3.org/>



(c) Copyrights, Yechiam Yemini, 1993-2003.

Architecture

- Server provides documents to client browsers
- Browser presents documents to users
- HTTP: request/reply protocol to retrieve/post documents
- URL: naming scheme to identify documents
- HTML: language to program hyperlinked documents



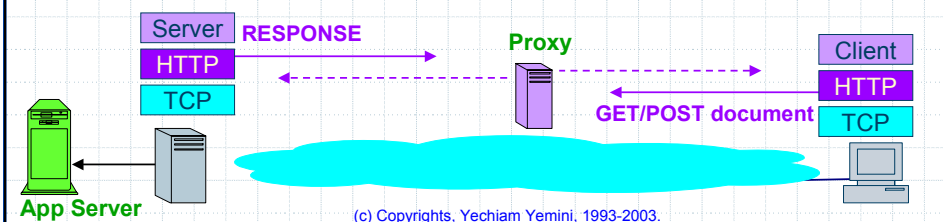
Confluence of Two Themes

I. Recast computing as tagged-data interpretation

- o Tagged-data languages (HTML...XML) provide universal presentation syntax
- o Computing tasks defined as tagged-document processing
 - Client-side: tagged-document presentation
 - Server-side: tagged-document creation/processing

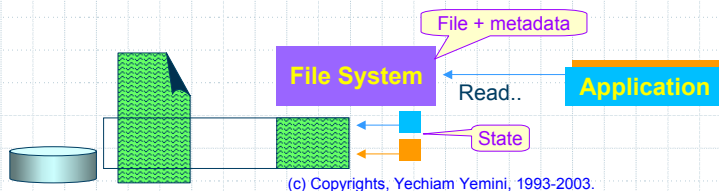
II. Recast client-server interactions as global file access

- o URL= a global naming scheme (directory structure) for a file-repository
 - Repository access: GET/POST...;URL ("file") can hide processing functions
- o A network of servers, proxies, load-balancers provides scalable file access



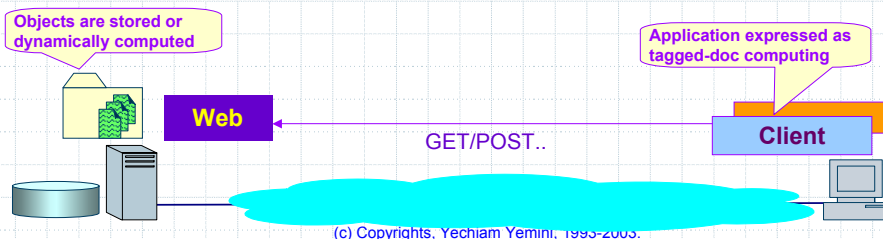
Digression On File Systems

- Traditional file system: shared access to persistent objects
 - Persistent objects exist in space and time outside a process scope
 - Must (a) have a name; (b) be accessible from process space; (c) shared by many processes
 - File system services: naming + file-data + metadata+ state + methods....
 - Methods: read/write data; manipulate metadata; manipulate state...
 - Unify I/O access: cast I/O as file access (e.g., block-devices, pipes...)
 - Note: a database adds data semantics and respective query services
- Applications are programs that manipulate the file system



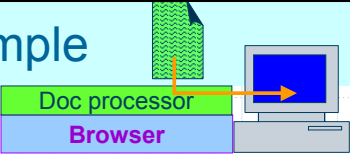
The Web As A Global File Repository

- HTTP provides network access to persistent objects
 - URL: global directory of persistent objects (objects hide services)
- Repository access
 - Reduced-granularity-access: GET/POST retrieval of an entire object (or a local aggregate)
 - Relaxed-response-time: user-machine scale
 - Reduced-services: no metadata, no state...
 - Metadata and state services are incorporated into document
- Network applications are programs to manipulate this global repository




Recasting *computing* as tagged-data interpretation

HTML By Example



```
<html>
<!--Just a simple homepage -->
<head>
  <title>My Home Page</title>
</head>
<body bgcolor=blue text=yellow link=white vlink=orange>
  <H1 ALIGN=center>
    <big>Welcome to my home page</big>
  </H1>
  <hr size=5 color=red>
  <center>
    <IMG SRC="riteofspring.jpg">
  </center>
  <H2 ALIGN=center>
    <u>Celebrating Spring Break</u>
    <br><small>(The Primavera; Sandro Botticelli 1445-1510)</small>
  </H2>
  <p>
    This course aims to provide:
    <ol>
      <li>Knowledge of networking fundamentals
      <li>Basic network building skills
      <li>An understanding of the state of the art
    </ol>
  </p>
  <a href="http://www.cs.columbia.edu/networks03/"> Click here for more details </a>
</body>
</html>
```

Welcome to my home page



Celebrating Spring Break
(The Primavera; Sandro Botticelli 1445-1510)

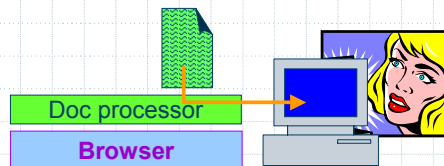
This course aims to provide:

1. Knowledge of networking fundamentals
2. Basic network building skills
3. An understanding of the state of the art

Click here for more details

Goals

- HTML started as a simple hypertext language
 - Enable user interaction with information
 - Simplified version of SGML; Basic construct: is markup text strings
 - Simple programming of attractive & effective GUI
 - To enable scalable navigation in vast information spaces
- Documents can be composed from components
 - Provide linking among documents
- Extensibility to present many types of objects
- Enable dynamic, interactive objects (animation, streaming)
- Evolutionary changes (HTML v.1 → v.4...DHTML, XHTML...)



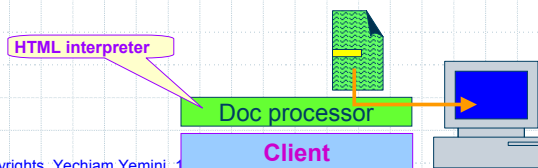
(c) Copyrights, Yechiam Yemini, 1993-2003.

HTML At A Glance

- Text layout constructors operate on text strings:
 - `<M> ...</M>` : e.g., `<p>..</p>`: paragraph brackets `...`: bold text
 - `<H1 ALIGN=CENTER> ... centered heading ... </H1>`: level 1 header brackets
 - `<BODY bgcolor=white text=black link=red vlink=maroon...>...</BODY>` document-wide attributes
- Incorporates a rich set of text formatting capabilities

- Bullet list, ordered list, definition list.....
-

```
<OL>
<LI> ... first list item
<LI> ... second list item
...
</OL>
```

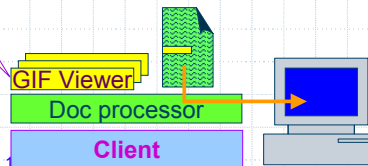


(c) Copyrights, Yechiam Yemini, 1

Linking and Media

- Reference links maintain relationships among documents
 - To local document components: `click here for something`
 - To documents on local computer: ` goodstuff`
 - To remote documents: `beststuff`
- Media files
 - In line image: ``
 - Mpeg movie: ` try this oldie`
 - Sound: ` happy birthday`

Viewers & plug-ins



(c) Copyrights, Yechiam Yemini, 1993-2003.

Advanced Features

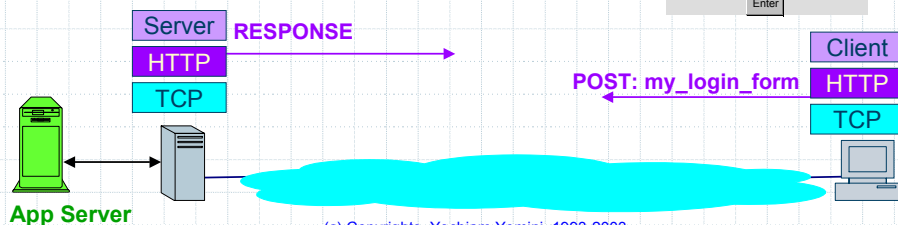
- Controlling timing
 - Animation; streaming media....
- Document structuring
 - Tables, frames...
- Templates
 - Style sheets; inheritance & cascading...
- Forms transmit info from client to server
 - HTTP supports posting forms/email to server
 - Forms are processed by server applications

```
<FORM Action="/my-CGI-dir/login.pl" Method=post>
<H1>Welcome </H1>
<PRE>
Login: <INPUT TYPE=text NAME="login_id">
Password: <INPUT TYPE=password NAME="pswd">
<INPUT Type=submit value="Enter">
</PRE>
</FORM>
```

Welcome

Login:

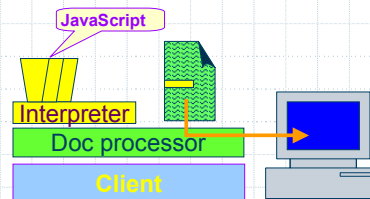
Password:



(c) Copyrights, Yechiam Yemini, 1993-2003.

Active Documents

- Agent-computing
 - Distributed computing: move data to programs
 - Agent computing: move programs to data
- Source-code agents: scripts dispatched to interpreters
 - LISP; Basic; Actors (1976, C. Hewitt)....
 -Tcl Agents → Java (94)... JavaScript; Jscript; VBS..
- Active documents incorporate scripts
 - Scripts must be syntactically identifiable
 - Dispatched to scripting language interpreter
 - Backward compatibility with legacy browsers by hiding scripts in comments



(c) Copyrights, Yechiam Yemini, 1993-2003.

Client-Side Computing With JavaScript

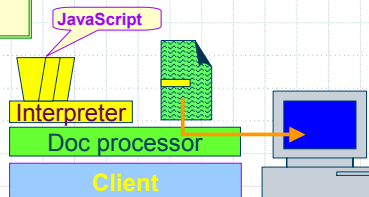
- Scripts provide extensibility of tag-processing
 - Primarily used to enrich interactions with users
- Interactivity: through event handling

```
...
Do you submit homework in time?
Yes: <INPUT type=checkbox onClick=write("Great")>
No: <INPUT type=checkbox onClick=write("Try-harder")>
...
```

- Interactivity: through document manipulations

```
...
<script language="JavaScript">
  nm=prompt("Hi;what is your name?","")
  document.write("welcome"+nm+"to the course site.")
</script>
```

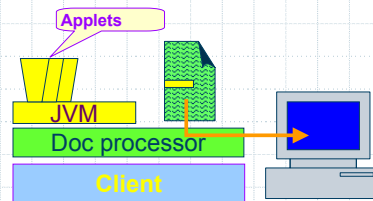
- What about security risks?
 - Richer expressiveness= greater risks
 - Scope = exposed data (e.g., what if scripts can only access local document)



(c) Copyrights, Yechiam Yemini, 1993-2003.

Client-Side Computing With Virtual Machines

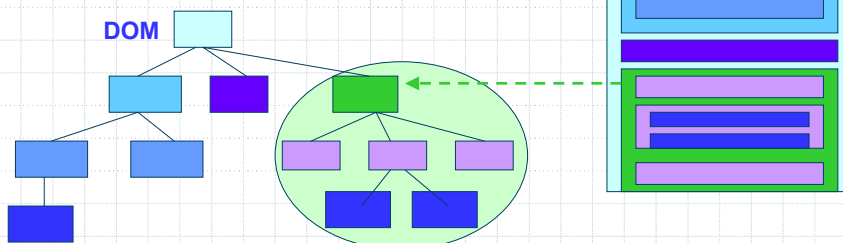
- Object-code agents: dispatched to virtual machine
 - Intermediate VM code is widely used for machine-independent compilation
 - Java Applets, ActiveX....
 - Active documents can incorporate agents' object code
- Client-side agents present security risks
 - Agents are (benign?) viruses
 - Often filtered by firewalls and protection programs
 - Object-code is more difficult to analyze or limit than source-code
- It is possible to improve security protections through
 - Sand-boxing
 - Signatures



(c) Copyrights, Yechiam Yemini, 1993-2003.

Dynamic HTML (DHTML)

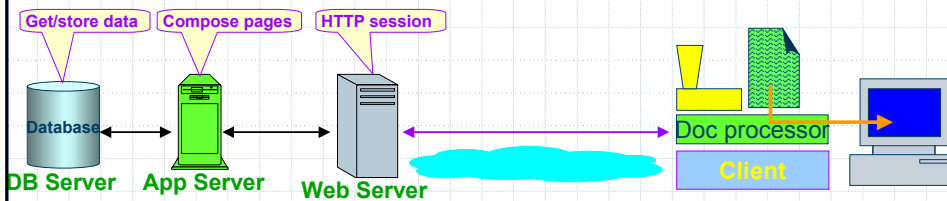
- Transforming HTML into a fuller programming language
- HTML+ Extensibility through style-sheets (~"macro")
- + Scope structure
 - Cascading, inheritance
 - Document Object Model (DOM) (~"parse-tree")
 - Sequential, one-pass interpretation of DOM
- + Extensibility through JavaScript



(c) Copyrights, Yechiam Yemini, 1993-2003.

Server-Side Computing

- Server-side computing
 - Web server: manage HTTP sessions
 - Applications server: assemble pages; process forms
 - Database: provide/store data
- Session management through cookies
 - Cookies help synchronize session state
 - But enable attacks on privacy
- Dynamic generation of documents



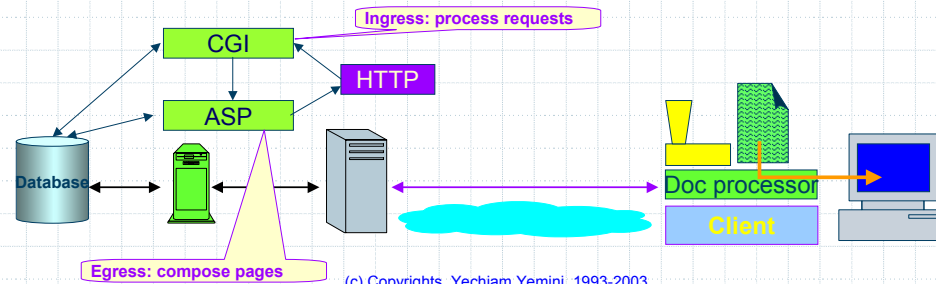
(c) Copyrights, Yechiam Yemini, 1993-2003.

First Generation Server-Side Computing

- CGI Scripts: process HTTP requests
 - Scripts (Perl) to process URL and forms
- Active Server Pages (ASP)
 - Use server-side scripts to compose page
 - VBS or JavaScript
 - Separate style (HTML) from content
- From CGI to application frameworks
 - Scaling server-side computing
 - The battle for the market: J2EE vs. .Net

```

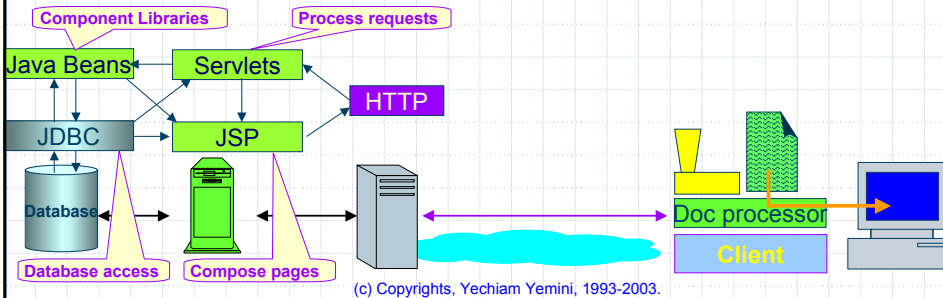
<%@language=JavaScript%>
<html>
<head>
  <title>My Page</title>
</head>
<%
  add-dynamic-component() =...;
%>
<body>
  .....
  <% =add-dynamic-component(cookie) %>
  .....
</body>
</html>
    
```



(c) Copyrights, Yechiam Yemini, 1993-2003.

J2EE Overview

- Fundamentals
 - Scaling for a large number of pages, services, sessions, wire-speeds....
 - Components-based architecture
- Servlets: control application workflow from request to response
- (Enterprise) Java Beans: components to compose application logic
 - Data,relationships, events...
- JDBC provides raw database access
- Java Server Pages (JSP) compose pages
 - Incorporate Java code in document



Summary: Recasting Computing

- Tagged-document processing
 - Data: tagged objects
 - Programs: tag interpreters
 - Sequential flow-processing
- Scripts/VM provide active extensibility of tag language
 - Client-side: interactive presentations
 - Server/proxy-side: dynamic composition, manipulations

