Web Server Languages
Summer 2017

Thomas A. Powell
tpowell@pint.com

http://www.pint.com/classes/cse135
Housekeeping

- Syllabus review
- Class page [http://classes.pint.com/cse135/](http://classes.pint.com/cse135/)
- Updated syllabus, notes, homework specifications, etc. will always be posted there
- History of this class
- Current focus confusion and my solution
- My goals for this class
- Attendance and participation
- Teams and the “pledge”
- Standard academic issues (esp. cheating)
What’s Web Programming All About

- Isn’t web programming just a form of client-server programming?
- Yes but...
  - Huge issues with an insecure and potentially unreliable network running software built often by those lacking formal large systems design.
- If you were in 134 you heard most of this but let’s make sure we spin it for the server-side and make sure to question things if we know more now.
- If you recently weren’t in 134 I have to assume you know at least some HTML5, CSS, and some JS!
  - Sadly Street HTML ≠ Real Client Side Knowledge
  - Avoiding knowing client tech creates a barrier for you and flies in the face of current dev trends (see Ajax)
  - We’ll do our best to solve this, but keep this in mind if staying in
What’s It All About

• Ok again isn’t web programming just a form of client-server programming?
• Yes but...
  - Huge issues with an insecure and potentially unreliable network running software built often by those lacking formal large systems design.

• User constraints
  • Training and background issues
  • no common platform in OS/browser/screen size/color/multimedia/etc
  • Users do not necessarily have good intentions
It’s the Network Stupid

Fallacies of networked computing

- The network is reliable.
- Latency is zero.
- Bandwidth is infinite.
Web Development Is Hard

Douglas Crockford on browsers:

“The most hostile software development environment imaginable.”
Developing Trouble

- Lack of formal software engineering practices are common in Web projects
  - Artists as programmers, programmers as artists
  - Do everything “web masters”
  - Starting to change finally!
  - Common jargon, convention, and standards issues
- Little testing and consensus on “best practices”
- Economic “first mover” advantage and its downside
User Concerns

- Do you really think about them enough
- How do you know you are succeeding?
- Measuring and Fact Fitting
  - The Visitation Fallacy
    - More always = better?
  - The Understanding Fallacy
    - 99% rule, testing artifacts, ad hoc conventions.
  - The Bandwidth Fallacy
    - Patience goes the other way
- The human (nature and capabilities) will see to be the constant & technology is the variable.
Err...Do You Even Know What it Is?

- What is Web design / development is different things to different people
- Depending on the person Web “D” includes many things such as:
  - Visual design
  - Programming
  - HTML
  - Navigation issues
  - Usability
  - Business issues (marketing, commerce, etc.)
- Depending on the project Web design actually may draw from any of these areas so it truly can be a very multidisciplinary field.
Big Challenge

• This lack of common understanding has really hurt the field more than one can imagine
  - Poor use of jargon
    • Buzz word bingo - Web 2.0, Social, DHTML, Ajax, HTML5
  - No agreed upon site structures and UI conventions
    • Consider the equivalent in GUI design
  - Crazy marcom driven notations about trade-off less tech or 2.0 3.0 er 4.0 of anything

• Be careful though, I am not promoting a “this is Web design/dev” solution
  - Often you make false comparisons
  - There is a wide range of solutions to a problem
A Review of Some Fundamental Ideas
Best Sites?

- So given a “best practices” approach to Web design and development answer the following:
  
  - Question: What are the “best” sites you know on the Web? In other words who does it right?

  - My answers _____, _____, _____, ______
  - Now what did those look like?
The 5 Pillars

1. Content
2. Structure
3. Technology (Implementation)
4. Delivery
5. Design
Another Way to View Web Development

Structure (e.g. XHTML)

Content

Interaction (e.g. programming) <- delivery ->

Presentation (e.g. CSS)
2 Participants

1. Site Owners
   - Developers, people who pay for the site, etc.

2. Users

Rule: You must balance between what the users want/needs are and the realities of the site owners’ wants/needs
2 Participants - A Balance of power

- A balance of power
  - User in control - mistakes made?
  - Too much developer control - feel restrictive

- "Las Vegas" or "Disneyland" design

- Seen it before the old Macintosh vs. command line argument
### Web Site Types

<table>
<thead>
<tr>
<th></th>
<th>Intranets</th>
<th>Extranets</th>
<th>Public</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Info about Users</strong></td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Capacity Planning</strong></td>
<td>Possible</td>
<td>Usually possible</td>
<td>Difficult to impossible</td>
</tr>
<tr>
<td><strong>Bandwidth</strong></td>
<td>High</td>
<td>Varies</td>
<td>Varies greatly</td>
</tr>
<tr>
<td><strong>Ability to set technology</strong></td>
<td>Yes</td>
<td>Sometimes</td>
<td>Rarely</td>
</tr>
</tbody>
</table>
Range of Web Sites

- Pure Static Web Site
- Static Site with Entry Form
- Dynamically Created Site
- Dynamic Data Access Via Web Site
- Web-Based Application
Site Types Contd.

- **Static Sites**
  - Most common
  - Pages don’t change per visitor and are built to fit users generically

- **Dynamic Sites**
  - Built on the fly for users
  - Personalized sites fall into this category (myYahoo)
  - Usually stored in a database

- **Interactive sites**
  - Those that allow the user to interact with content or site features in a significant fashion beyond simple selection
Dynamic Site Overview

User requests page

Server returns static file
(Same for everybody)

User requests page

Dynamically created page returned
(May be different per person or browser)
The Medium of Development & Context

Client Side
- HTML Markup
- Images
- Java Applets, ActiveX Controls or Plug-Ins
- Browser
- Multimedia
- Scripting (e.g. JavaScript)
- Helper Applications (e.g. Excel)

Server Side
- Web Server
- Server API Programs (e.g. ISAPI, NSAPI)
- Server side scripting pages (e.g. Active Server Pages)
- CGI Program
- Backend Database

Network

Demo Company
Typical Web App Set-Up

- **Web Client**
  - HTTP request (cleartext or SSL)
  - HTTP reply (HTML, Javascript, JSON, Flash, etc)

- **Web Server**
  - Apache
  - IIS
  - Nginx
  - etc...

- **Database**
  - J2EE
  - .NET
  - JS
  - PHP etc.

- **Database connection**
  - ADO
  - ODBC, etc.

- **Firewall**
## Web Programming Toolbox

<table>
<thead>
<tr>
<th>Client Side</th>
<th>Server Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helper Applications</td>
<td>CGI scripts and programs</td>
</tr>
<tr>
<td>Netscape Plug-ins</td>
<td>Server API Programs</td>
</tr>
<tr>
<td>ActiveX Controls</td>
<td>* ISAPI</td>
</tr>
<tr>
<td></td>
<td>* NSAPI</td>
</tr>
<tr>
<td></td>
<td>* Apache Modules</td>
</tr>
<tr>
<td>Java Applets</td>
<td>Java Servlets</td>
</tr>
<tr>
<td>Scripting Languages</td>
<td>Server-side scripting</td>
</tr>
<tr>
<td>* JavaScript</td>
<td>* Active Server Pages (ASP)</td>
</tr>
<tr>
<td>* VBScript</td>
<td>* ColdFusion</td>
</tr>
<tr>
<td></td>
<td>* PHP</td>
</tr>
</tbody>
</table>
What is a browser?
Browser Side Interactions

- `<div>` `<img>` `<iframe>` `<body>` `<form>` `<input>` `<table>`
- `<object>` `<embed>` `<script>`

- HTML+CSS
- Javascript
- DOM
- ActiveX
- mime types
- toolbars
- Flash
- libraries
Simple View of Client Side Web Development

- **Markup**
  - (HTML, XHTML, XML)

- **DOM**

- **Interaction**
  - (JavaScript, Objects - applets, plugins, etc.)

- **Presentation**
  - (HTML*, CSS, Media objects, Flash)
Simple View of Server-Side Web Development

HTTP Request

Web Server
Hardware & Software

Server Side programming technology

Backend System (e.g. Database)

User Agent of some sort

HTTP Response

Apache, IIS, Zeus, etc.

CGI
Apache Module, ISAPI
Scripting Tech (PHP)
HTTP Request/Response Cycle Preview

Asks for resource by its URL:
http://www.foo.com/page.html

HTTP Request

HTTP Response

www.foo.com

Resource
/page.html

Browser decodes MIME type and determines action
maps file extension .html to appropriate MIME type: text/html
HTTP Request Example Shows Process

07/01/04 09:07:02 Browsing http://www.ucsd.edu
Fetching http://www.ucsd.edu/ ...
GET / HTTP/1.1
Host: www.ucsd.edu
Connection: close
User-Agent: Sam Spade 1.14

HTTP/1.1 200 OK
Date: Thu, 01 Jul 2004 16:07:00 GMT
Server: Apache/1.3.27 (Unix)
Last-Modified: Thu, 01 Jul 2004 16:01:00 GMT
ETag: "c992b-77df-40e4353c"
Accept-Ranges: bytes
Content-Length: 30687
Connection: close
Content-Type: text/html

<!DOCTYPE html public "-//W3C//DTD HTML 4.0 Transitional//EN">
<html lang="en">
<head>
    <base href="http://www.ucsd.edu/">
    <title>University of California, San Diego</title>
    <meta name="generator" content=""/>
    <meta name="author" content=""UCSD Libraries, Information Technology Department">
    <meta name="keywords" content=""/>
Browser Lookup for Action on Mime (or file type)
Example of file extensions and action

HTTP Request for file.php

Browser receives response with appropriate MIME type (usually text/html) and renders

Q: If you view source what do you see?

Script engine intercepts and evaluates PHP code
Range of Interface - Progressive Enhancement

- Text + Images + Animation + Audio + Video + Immersive Technology (3D)
  - Text + Images + Animation + Audio + Video
    - Text + Images + Animation + Audio
      - Text + Images + Animation
        - Text + Images
          - Plain Text
Range of Interface - Technology

- XHTML + CSS + Mandatory JavaScript + Mandatory Flash
  - XHTML + CSS + Mandatory JavaScript + Optional Flash
    - XHTML + CSS + Mandatory JavaScript
      - XHTML + CSS + Optional JavaScript
        - XHTML + CSS
          - HTML + Tables
            - Simple HTML
Range of Connectivity

Offline
Disconnected

Online
Connected

Low Latency

High Latency

Low Bandwidth

High Bandwidth

Chance network or traffic effects can alter experience
Range of Interaction

Static Site
(ex. My Yahoo!)

Custom
(ex. message board/wiki)

Participatory
(ex. virtual space)

Beyond...
Interface Style Choices

Traditional


Read, Understand, Click
(Simple links/press buttons)

Direct Manipulation

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Drag & Combine
(Select region, drag and drop, fill, etc.)
Site Types Some Groupings

- Informational
- Transactional
- Community
- Entertainment
- Other
  - Blogs, artistic, personal
- Commercial
- Government
- Educational
- Non-profit
- Personal
Visual Groupings - Text Oriented
Visual Groupings - GUI Style

Welcome to Demo Company

Domes
- Personal Space Vehicles (PSVs)
- Balloon Packs
- Robots

Don't wait for the future to arrive - buy a piece of it now with one of Demo Company's amazing products! Want to be pampered by your own robot butler while living under the ocean, commute to the moon and take long, relaxing balloon rides with your significant other (or reasonable facsimile)? Now you can, thanks to Demo Company - making the impossible possible since 1999!

For more information, choose one of our four product categories:

- Domes
- Personal Space Vehicles (PSVs)
- Balloon Packs
- Robots

Making the impossible possible since 1999!

We are the Balloon Pack of Robots. Personal Space Vehicles (PSVs) are in the known universe... and our Balloon Packs are robots. This Web site also demonstrates what our parent company, Pnt, Inc. (www.pnt.com), can do for your company. Our demos include and easy-to-use maintenance system, our amazing internet, and the customized Build Your Own Sport PSV tool. Whether you're searching for the utmost futuristic consumer products or the latest in database-driven Web sites, you've come to the right place.

Controls
- To adjust this page, you can drag the window to any location that suits you. You can also maximize or close them. Here are some other window to open:
  - Updated Product Listing
  - Search Box
  - Maintenance System Login
  - Employee Only

Jobs at Demo Company
- Interested in a career with Demo Company?
  - If so, be sure to check here for the latest job postings.

Dome Repair Technician
- Prior experience with dome repair a plus. Must have excellent knowledge of electrical systems as well as working knowledge in making sure things are air tight!

Calendar of Events
- Be sure to check back often to stay informed on the most recent Demo Company events and training seminars.
Visual Style - Metaphorical
Visual Style - Experimental

Demo Company is a fictitious company sponsored by Powell Internet Consulting that allows users to play with a variety of tools. Demo Company has four main product lines: Domes, Personal Space Vehicles (PSVs), Balloon Packs and Robots. You can use our PSV finder to figure out the best PSV for you. You can also build your own PSV and see it in several colors of your choice. There is also a maintenance system to see as well as the cool design!
Range of Underlying Info Structure - Linear
Range of Info Structures: Contd.

Grid
Range Contd: Web Site Structure and Trees

Narrow Hierarchy

Wide Hierarchy
Applied Tree

- **HOME**
  - **ABOUT**
    - HISTORY
    - BIOS
      - Tom
      - Jim
  - **PRODUCTS**
    - ROBOTS
      - Butler
      - Trainer
    - PSVS
  - **NEWS**
    - Press
      - Release 1
    - Press
      - Release 2
    - Press
      - Release 3
  - **CONTACT**
Is it a Tree?
Give Them Infinite Choice?
Just a Graph

Pure Web
Realistic User View of the Site

Diagram showing the user journey from Entry to Exit Page.
Well...it might be a bit more complicated
Goal Driven vs. Structure Driven Web Dev?

- Focusing on entry and exit and trying to get people to get what they want quickly
  - Amazon knows how many clicks you want to do...1!

- Consider Amazon’s feature other people who bought this book bought these books

- This approach to navigation is known as information foraging
  - The wild animal and food example is good to keep in mind
Site Navigation vs. Web Navigation

- Remember that your site may just be one “Island Hop” on a user’s visit to many sites trying to accomplish some overall task.
Entry Point Important: Consider This
What about Exit?

• Is there an exit?
• The idea of “closure”
• Statefulness, statelessness and the Web
  - HTTP by design
  - But...do they logout?
  - Security, Tracking, Errors,...HEADACHES!
• We are going to see that much of what happens even at user level is a side-effect of core aspects of things like HTTP
Web Development

“A multidisciplinary pursuit pertaining to the planning, and production of Web sites, including, but not limited to, technical development, information and its structure, visual design, and networked delivery.”
Common Web App Design Themes

- Generally the major themes behind modern Web design include:
  - Designer/Client needs versus user needs
  - The balance of form and function
  - The quality of execution
  - Interplay between convention and innovation
User Focused Design

- UCD - the concept of designing something (in our case Web sites) always with the user and use in mind.

- Some important rules we cover:
  - Rule: YOU are NOT the USER
  - Rule: USERS are NOT DESIGNERS
  - Rule: Design for common, account for differences
  - Rule: Users are REAL PEOPLE
Form and Function

- Favorite Catchphrase: Form follows function!

- Rule: The visual form of a site should relate to its function

- Extreme examples to illustrate the point
  - Overly flash based site for your IRS tax form
  - All text driven move promotion site

- Interesting how design is not bottom-up today but top-down if this is true?
Who does the user blame for mistakes?
- Slow speed, no JavaScript, broken links, bad render, etc.
- No limitations in most case for the user to leave - just click [no uninstall barrier]

Reason for mistakes?
- Misunderstanding the medium and its constraints
- Lack of process
- Lack of professional education
- Lack of engineering style thinking
Conformity versus Innovation

- Users come to Web sites with history
  - 99% rule, big site effect, their operating system, etc.

- There is a rich history in computer interfaces
- There is an emerging history in Web design

- Rule: Appropriately respect GUI and Web interface conventions

- You may want to break the rules when you know the rules!
Learning Web Design and Development

- Theory
- Observed Practice - Evaluations
- Your own implementations
- But hey we aren’t artists!
- Always remember there is no single form of “correct” Web ‘design’ that will fit every site or situation
  - Splash page example
We Should Know the Details

**We get away with things**

- The Web is different!?
  - Browsers fix many of your problems
    - Markup, CSS, some network configuration with MIME types and even JavaScript!
    - Imagine a C compiler trying to guess what you mean!

- We really have to do our job right particularly as the distinction between software genres melts away, but what is our job...is there something more than knowing the tech that is maybe tougher?
All About End Users?

• Users declare us good or bad for better or worse
  - They often only see results and things above the water
  - “The Iceberg Model”

• Example: Speed is all important - you can never have it fast enough!
  - User don’t care about bytes, they care about time so… implications?
    • Watch out for broadband fallacy
  - Is time and perception of slowness consistent?
  - Given the cycle of read, decide, click, wait, repeat can we play a trick? Yet to do so we have to have technical chops and an understanding of user and experience
The Inherent Trade-offs

- **Client-side**
  - No control - end user environments vary greatly and then can do what they want to your markup and code
  - Speed and scalability
    - Responsive interface since no network round-trip
    - Off load server from duties it shouldn’t need to perform

- **Server-side**
  - Control is yours - you choose the technology and approach
  - The “secrets” hopefully should be safe
  - Speed and scalability could be a problem as you round-trip all the time and you find your systems are hung in network wait states all the time
Not a “versus” Issue

- You need both CS AND SS, it is just a question of what makes sense where
  - Client-side tends to be good for interface concerns
    - Validation, UI, etc.
  - Server-side tends to be good for data concerns
    - Submission and storage of sensitive data in particular

- Sacrifices and trade-offs are made, there is no vacuum here you may be forced or encouraged to balance CS and SS differently depending on user needs, internal conditions, developer familiarity or simply time.