Distributed Systems

Browser-Based Programming
Objectives

- Look at the user-side of distributed systems
- Introduce browser-based programming techniques
- Introduce device-awareness & content adaptation
# User-side of Distributed Systems

<table>
<thead>
<tr>
<th>Distributed Systems</th>
<th>Desktop / Laptop</th>
<th>Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser-based</td>
<td>Joe XHTML CSS Javascript Ajax</td>
<td>Joe XHTML CSS Javascript Ajax</td>
</tr>
<tr>
<td>Native application</td>
<td>(not today) Java</td>
<td>Mike Android</td>
</tr>
</tbody>
</table>
Browsers

• Browsers provide a powerful, standards-based (for the most part) platform for developing user applications.

• With varying capabilities, they work on:
  – Feature phone
  – Smartphone
  – Blackberry / iPhone / Android
  – iPad
  – Netbook / Laptop
  – Desktop
  – ?
Outline

• Core browser technologies
  – XHTML
  – CSS
  – Javascript

• Ajax

• Device awareness
  – User-Agent
  – Device Description Repositories
    • WURFL
Core browser technologies

- Structure
  - XHTML
- Presentation
  - CSS
- Behavior
  - Javascript

Mostly outside the scope of this course

We will focus on the communication aspects
HTML, CSS, Javascript

- These are outside the scope of this course, but necessary for scripting browser-based user interfaces.
- If you don’t have some prior experience with HTML and CSS, you should familiarize yourself.
- As an IT professional, these are well within your grasp.
- You will regularly be expected to pick up new languages, platforms, frameworks, and architectures in your work.
- If a question / answer / primer session would be worthwhile, we could arrange one outside of class.
Resources

• Here are a few good video introductions
  – HTML & CSS – The VERY Basics
  – Google: HTML, CSS, and Javascript from the Ground Up
    • http://code.google.com/edu/submissions/html-css-javascript/
  – Douglas Crockford videos
    • http://video.yahoo.com/watch/111593/1710507

• And a few cheatsheets:
  – Javascript:
    http://www.addedbytes.com/cheat-sheets/javascript-cheat-sheet/
Ajax

• What is Ajax?
  – Shorthand for Asynchronous JavaScript and XML
  – Is a technique of designing web-based applications

• What are exemplars of an Ajax approach?
  – Gmail
  – Google Maps
  – Others?
Ajax is built on

- HTML/XHTML
- CSS
- Javascript
- Document Object Model (DOM)
- Event model (e.g. onMouseClick)
- Web remoting (e.g. XMLHttpRequest)
Some Ajax Patterns

• Dynamically update page elements
• Dynamically restructure page layout
• Dynamically update content from remote source
• Friendly interactive interface to remote services
• Responsive first-person interaction
• Collaborative interaction
• Streaming within a web-page context
• Submission throttling
• Partial completion
History of browser / server interaction

• Static web page
  – User / Browser / web server

• Dynamic web page / web application
  – User / Browser / web server / CGI
  – CGI may be PHP, Ruby, Java/JEE, etc.

• Ajax-style web application
  – User / Event Handler / Response Handler / web app
Browser – server communication

• Key to the Ajax style communication is communication between a browser and a server within the context of a page
  – I.e. without a new page load
  – Avoiding “click and wait”

• XMLHttpRequest is the primary means of facilitating this communication

• All modern browsers have a built-in XMLHttpRequest object.
  – E.g. IE7+, Firefox, Chrome, Safari, Opera...
XMLHttpRequest basics

• Get a new XMLHttpRequest
• Open it
  – Blocking, or non-blocking
• Send a request
• Process the response
New

• xhr = new XMLHttpRequest();
XMLHttpRequest open parameters

• 1st : http request method (text string)
  – i.e. get, post (most often)
  – Also head, put , delete, options

• 2nd: URL of HTTP request (text string)
  – Typically restricted to be on the same server:port

• 3rd: is the request asynchronous (boolean)
  – True means a send will not block (asynchronous)
  – False means a send will block (synchronous)
Synchronous

• If synchronous
  – Send...
    • Will block until response fully received
  – Check if response.status == 200  // OK
  – Use results
Asynchronous XMLHttpRequest

- If non-blocking (asynchronous)
- xhr.readyState values
  0. Object has been constructed
  1. Open method successful
     - headers can now be set
     - send can now be done
  2. Response headers have been received
  3. Response body is being received (in progress)
  4. Response complete (or something went wrong)
Asynchronous

- If asynchronous
  - Send...
  - In onreadystatechange handler
    - If (request.readyState == 4)
    - // response ready
  - Check if response.status == 200
  - Use results
XMLHttpRequest send parameters

• **GET**
  - Put parameters on the URL query string
  - `open("GET", "http://localhost/ajax/test.aspx?param1=x&param2=y", true);`
  - `send(null);`

• **POST**
  - Put parameters as arguments to send
  - `open("POST", "http://localhost/ajax/test.aspx", true);`
  - `send("param1=x&param2=y");`
Use the results

• Text
  – The results can be treated as text
  – responseText

• XML
  – Or, if the response is valid XML, it can be treated as such
  – responseXML will be a DOM document object
Simplest example

http://www.andrew.cmu.edu/course/95-702/examples/ajax/eg1.html

```html
<head>
  <script>
    function doXMLHttpRequest() {
      xhr = new XMLHttpRequest();
      xhr.open("GET", "data.txt", false);
      xhr.send(null);
      if(xhr.status == 200)
        document.getElementById("responseArea").innerHTML=xhr.responseText;
      else
        document.getElementById("responseArea").innerHTML="Error code " + xhr.status;
    }
  </script>
</head>

<body>
  <FORM name="eg1">
    <INPUT type="BUTTON" value="Do XMLHttpRequest" ONCLICK="doXMLHttpRequest()">
  </FORM>
  <div id="responseArea"> </div>
</body>
```

http://www.andrew.cmu.edu/course/95-702/examples/ajax/data.txt

Hello class

95-702 Distributed Systems
Synchronous example

• Blocking send
• Review Eg2.html
  http://www.andrew.cmu.edu/course/95-702/examples/ajax/eq2.html
Asynchronous example

- Non-blocking send
- Other things can continue to happen
- Review eg3.html
  
  http://www.andrew.cmu.edu/course/95-702/examples/ajax/eg3.html
Ajax on mobile

- Ajax works on mobile also
- Subject to any limits on the implementation of:
  - XHTML (XHTML-MP)
  - Javascript
- Examples with mobile DOCTYPE
  - eg1-mobile
    - [http://www.andrew.cmu.edu/course/95-702/examples/ajax/eg1-mobile.html](http://www.andrew.cmu.edu/course/95-702/examples/ajax/eg1-mobile.html)
  - eg3-mobile
    - [http://www.andrew.cmu.edu/course/95-702/examples/ajax/eg3-mobile.html](http://www.andrew.cmu.edu/course/95-702/examples/ajax/eg3-mobile.html)
WURFL


- Android emulator->

![Android Emulator Interface](image-url)