CS520 Web Programming
Introduction to Ajax

Chengyu Sun
California State University, Los Angeles

Browser As The New OS
- Application can be used from anywhere
- Easy application distribution and deployment
- Greatly simplifies system administration
  - No software to download, install, and update
  - Centralized data management

So why it didn't happen??

Disadvantages of Web Applications
- Usually requires high bandwidth
- Storing data remotely
  - Privacy
  - Reliability
- Limited number of GUI components
  - Compared to, e.g.:
    http://java.sun.com/docs/books/tutorial/ui/features/complWin.html
- Interactivity issues

Interactivity Issues
- Conventional GUI application
  - Rich event model
  - Responsive
    - No network delay
    - Partial redraw
- Conventional Web application
  - Simple request-response
  - Not so responsive
    - Send request, wait for response
    - Full page refresh

So how do we make web applications behave like desktop applications??

HTML Event Models
- HTML 4 Event Model
  - HTML 4.01 Specification -
    http://www.w3.org/TR/REC-html40/interact/scripts.html#h-18.2.3
  - Limited but widely supported
- Standard Event Model
  - DOM Level 2 HTML Specification -
    http://www.w3.org/TR/DOM-Level-2-Events/events.html
- Browser specific event models

Events and Event Handler
- Events
  - onfocus, onblur, onkeypress, onkeydown, onkeyup, onclick, ondbclick, onmousedown, onmouseup, onmouseover, onmousemove ...
- Specify event handler
  - <element event="code">  
  - For example:

  <button onclick="clickHandler();">click</button>
**Example: Event Handling**

- j1.html
  - Uses X Library from http://cross-browser.com/
  - Handles events
  - Modifies the HTML document

**JavaScript**

- Interpreted language
- Originally developed by Netscape
- Syntax is similar to Java

```
<script>
  // Code here
</script>
```

**Core JavaScript**

- Mainly covers language syntax, which is kind of similar to Java
- Global Object
  - Created by a JavaScript interpreter
  - Global variables and global methods are simply variables and methods of this object

**Client-Side JavaScript**

- Embed JavaScript in HTML
  - `<script>`
    - `type="text/javascript"`
    - `language="JavaScript"`
    - `src="path_to_script_file"`
- Run inside a browser
- `Window` is the global object

**Document Object Model (DOM)**

- Representing documents as objects so they can be manipulated in a programming language.

**An HTML Document**

```html
<html>
  <head>
    <title>JavaScript Example</title>
  </head>
  <body>
    <h1>JavaScript Example</h1>
    <p>Some content.</p>
  </body>
</html>
```
DOM Representation

Nodes

Manipulate a Document
- Find Elements
- Modify Elements
- Create Elements

Find Elements
- `document.getElementById()`
- `document.getElementsByTagName()`
- `document.getElementsByName()`

Modify Elements ...
- HTML Element properties and methods
  - IE
    - `innerHTML`
    - `innerText`
    - `insertAdjacentHTML()`
    - `insertAdjacentText()`
  - Netscape/Mozilla
    - `innerHTML`
  - Element-specific

... Modify Elements
- `node`
  - `setAttribute()`, `removeAttribute()`
  - `appendChild()`, `removeChild()`
  - `insertBefore()`, `replaceChild()`
Create Elements

- `document`
  - `createElement()`
  - `createTextNode()`

Example: Document Manipulation

- `j2.html`
  - Read and display the text input
  - Display "Hello <name>"??
  - Add text input to table??

Communicate with Server

- The request-response model is still a limiting factor in user interactivity
- Solution: XMLHttpRequest
  - A JavaScript object
  - Send request and receive response
  - Response can be handled *asynchronously*
    - Do not need to wait for the response

Understand Asynchronous

- **Synchronous**
  - `send( request );`
  - // wait for response
  - `process( response );`
  - // do other things
- **Asynchronous**
  - `send( request );`
  - // don’t wait for response
  - `process( response );`
  - // do other things
  - What’s the problem??
  - What’s the solution??

An XMLHttpRequest Example

- `a1.html`
  - A client script sends an XMLHttpRequest
  - A servlet responds with an XML message
  - When the message arrives on the client, a *callback function* is invoked to update the document

About the Example

- `clickHandler()`
- `newXMLHttpRequest()`
- `updateDocument()`
- `getReadyStateHandler()`
XMLHttpRequest - Properties

- onreadystatechange
- readyState
  - 0 – uninitialized
  - 1 – loading
  - 2 – loaded
  - 3 – interactive
  - 4 – complete
- status
- statusText
- responseBody
- responseStream
- responseText
- responseXML

XMLHttpRequest - Methods

- abort()
- getAllResponseHeaders()
- getResponseHeader( header )
- open( method, url, asyncFlag, username, password )
  - asyncFlag, username, password are optional
- send( messageBody )
- setRequestHeader( name, value )

So What is Ajax?

- Asynchronous JavaScript and XML
  - JavaScript + XMLHttpRequest
- Characteristics of Ajax
  - Non-blocking – the server response is handled asynchronously with a callback function
  - Partial page update using JavaScript

More About AJAX

- XMLHttpRequest used to be an IE specific feature that received little attention
- It’s all started by Google Maps
- The beginning of “Web 2.0” (or 3.0)

Key Elements of an Ajax Operation

<table>
<thead>
<tr>
<th>Client</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>Process the request</td>
</tr>
<tr>
<td>Event handler</td>
<td>Send back a response</td>
</tr>
<tr>
<td>Create a XMLHttpRequest</td>
<td>Attach a callback function</td>
</tr>
<tr>
<td>Attach a callback function</td>
<td>Send the request</td>
</tr>
<tr>
<td>Callback function</td>
<td>Process the response</td>
</tr>
<tr>
<td>Process the response</td>
<td>Update the HTML Page</td>
</tr>
</tbody>
</table>

AJAX Frameworks and Libraries

- http://ajaxpatterns.org/Ajax_Frameworks
More Widgets, Less JavaScript

- Simplifies XMLHttpRequest creation and response handling
  - E.g. X Library, Taconite
- AJAX widgets libraries
  - E.g. Ajax JSP Tag Library, YUI
- Full-fledged web development frameworks
  - E.g. ZK, GWT
- AJAX widgets for existing web development frameworks
  - E.g. ASP, JSF

More Ajax Examples

- a2.html - a Taconite example
  - http://taconite.sourceforge.net/
  - Simplifies request creation
  - Response generated by JSP
  - No JavaScript required to update page
  - Identify the key Ajax elements in this example
- CSNS
  - Toggle file public
  - Add section

Readings


What’s in the Future? – RIA vs. Ajax

- Rich Internet Application (RIA) platforms
  - Flex, Silverlight, JavaFX
- vs. Ajax
  - Proprietary
  - Require browser plugins
  - Powerful GUI features
  - Good development tool support
  - Desktop development experience
- HTML5??