Introduction to Ajax

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.
- Interactivity issues

Interactivity Issues

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

Example: Event Handling

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

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](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](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, onmousemove, onmouseout ...
- Specify event handler
  - `<element event="code">`
  - For example:
    - `<button onclick="clickHandler();">click</button>`

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

<html>
  <head>
    <!-- Some content. -->
  </head>
  <body>
    <title>JavaScript Example</title>
    <h1>JavaScript Example</h1>
    <!-- Some content. -->
  </body>
</html>

Nodes

- Document
- HTMLDocument
- Node
- CharacterData
- Text
- Comment
- Attribute
- Element
- HTMLElement

Manipulate a Document

- Find Elements
- Modify Elements
- Create Elements

Find Elements

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

Modify Elements ...

- HTMLElement properties and methods
  - IE
    - `innerHTML`
    - `innerText`
    - `insertAdjacentHTML()`
    - `insertAdjacentText()`
  - Netscape/Mozilla
    - `innerHTML`
  - Element-specific

... Modify Elements

- Node
  - `setAttribute()`, `removeAttribute()`
  - `appendChild()`, `removeChild()`
  - `insertBefore()`, `replaceAll()`
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 HTTP request
    - Parse XML response
  - Response can be handled asynchronously

An XMLHttpRequest Example

- `a1.html`
  - A client sends an XMLHttpRequest
  - A servlet response 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`
- `readystatechange`
  - 0 – uninitialized
  - 1 – loading
  - 2 – loaded
  - 3 – interactive
  - 4 – complete
- `status`
- `statusText`
- `responseBody`
- `responseStream`
- `responseText`
- `responseXML`
XMLHttpRequest - Methods

- abort()
- getAllResponseHeaders()
- 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
  - Vs. Yahoo Maps (The Old Version)
  - The beginning of “Web 2.0” (or 3.0)

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
  - Simplifies request creation
  - Response generated by JSP
  - No JavaScript required to update page
- 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/HTML5
  - Proprietary
  - Require browser plugins
  - Powerful GUI features
  - Good development tool support
  - Desktop development experience