

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/compWin.html>
- ◆ *Interactivity issues*

Interactivity Issues

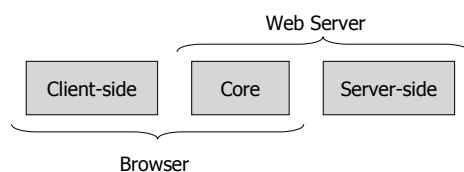
- ◆ Conventional GUI application
 - Rich event model
 - Responsive
 - No network delay
 - Partial redraw
- ◆ Web application
 - Simple request-response model
 - Not so responsive
 - Send request, wait for response
 - Full page refresh

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

JavaScript

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



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

Example: Event Handling

- ◆ `jl.html`
 - Uses X Library from <http://cross-browser.com/>
 - *Handles events*
 - *Modifies the HTML document*

Events and Event Handler

- ◆ Events
 - `onfocus`, `onblur`, `onkeypress`, `onkeydown`, `onkeyup`, `onclick`, `ondblclick`, `onmousedown`, `onmouseup`, `onmousemove`, `onmouseover` ...
- ◆ Specify event handler
 - `<element event="code">`
 - For example:

```
<button onclick="clickHandler();">click</button>
```

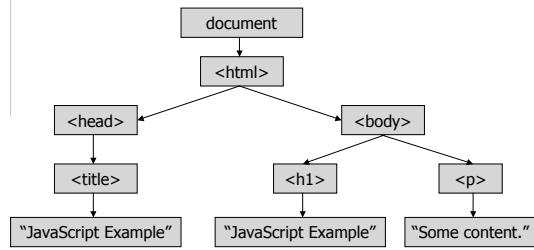
Document Object Model (DOM)

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

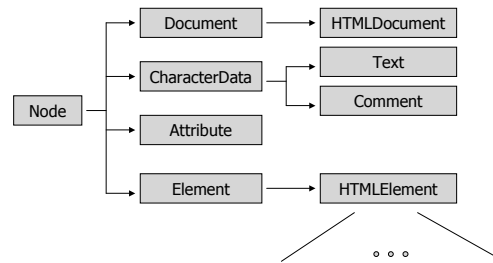
An HTML Document

```
<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.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()`, `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 HTTP request
 - Parse XML response
 - *Response can be handled asynchronously*

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)

An XMLHttpRequest Example

- ◆ a1.html
 - A client scripts sends an XMLHttpRequest
 - A servlet responses 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()

So What is Ajax?

- ◆Asynchronous JavaScript and XML
 - <http://www.adaptivepath.com/ideas/essays/archives/000385.php>
 - 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

- ◆ AJAX: Getting Started - http://developer.mozilla.org/en/docs/AJAX:Getting_Started
- ◆ Mastering AJAX, Part 1-3 - http://www-128.ibm.com/developerworks/views/web/libraryview.jsp?search_by=Mastering+Ajax
- ◆ Taconite Documentation - <http://taconite.sourceforge.net/>