

AJAX Overview

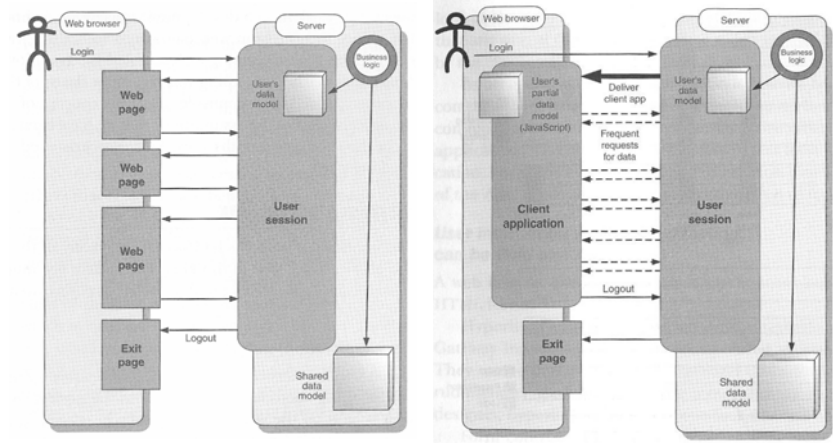
MSIS 531 – Spring 2006

2

AJAX Overview

- AJAX: “Aynchronous JavaScript and XML”
- New technique for creating interactive web applications
 - Not a new *Technology* but more of a *Pattern*
- Motivation:
 - HTTP never intended to dynamically serve content
 - Pages always reload, but never get updated
 - Users wait for the entire page to load even if a single piece of data is needed
 - Single request/response restrictions: no middle ground between “this page” and “next page”

Web Applications and AJAX

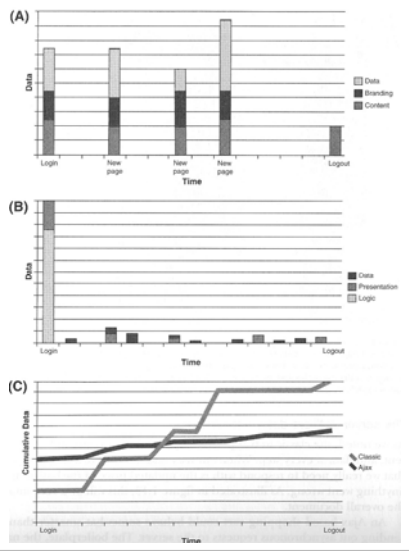


Standard web app—many web pages, each refreshing the screen. Conversational state on server

AJAX app – client code delivered at login, requests processed on client or server without interrupting workflow

Content Delivery

- Classic webapp – everything delivered for each page
 - Browser caching reduces, but does not remove, this overhead
- AJAX webapp – initial large delivery, subsequent activity low
 - Front-loaded client load less expensive than you might think
- Cumulative data delivered
 - Higher at first, but cumulative data cost lower
 - Clear benefit if WAN involved

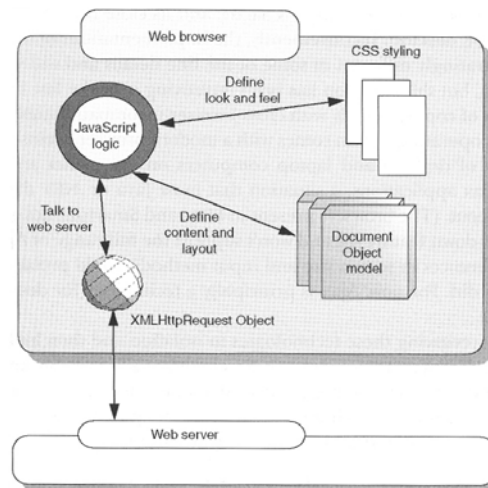


AJAX Design Principles

1. The browser hosts an application, not content
 - Application code delivered to browser, mostly as JavaScript code
2. The server delivers data, not content
 - Data may be plain text, JavaScript fragments, or XML documents
3. User/application interaction is continuous and fluid
 - UI metaphors like drag-and-drop become possible
4. This is real coding and requires discipline
 - Significant developer responsibility to manage conversational state over entire web transaction

AJAX Components

- HTML and CSS
 - Presentation, with standardized style information
- Document Object Model (DOM)
 - Dynamic display of and interaction with the HTML page
- XMLHttpRequest object
 - Asynchronously retrieves data from web server
- Javascript
 - Client-side code controls actions (controller pattern)



Cascading Style Sheets (CSS)

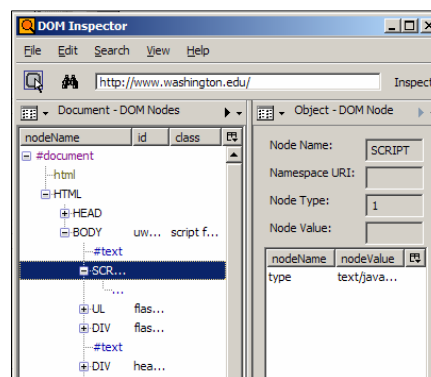
- Well-established design technique for classic and Ajax web applications
- Goal: standardize formatting (“styles”) of web content
 - In classic web applications, promotes resuability
 - In Ajax, allow predefined looks, applied w/minimum code
- CSS selectors are applied to HTML tags
 - Example: make this heading red


```
.redhead { color: red; }
<h1 id="head1" class="redhead">A red headline</h1>
```
- Many other options available to CSS designers

Document Object Model (DOM)

- The DOM exposes a web page to the JavaScript engine
- Tree-structured format
 - DOM structure well defined by World-Wide Web Consortium (W3C)
 - Each DOM element is a node
- DOM can be read, and also updated (change, insert)
- Programming recommendations:
 - Use of “id” attribute to make finding elements easy
 - CSS styles applied using node’s className attribute:


```
head1.className = 'redhead';
```
 - Use of XHTML



JavaScript

- Solid browser-based programming language with a very bad reputation
 - Source of pop-up windows, back button hacks, image rollovers
- Not Java, but from the same family
 - Java-like syntax
 - Loosely-typed variables, dynamically interpreted
 - Functions are objects
 - Can be invoked from outside a class
- Can use OOP-like style, but not required

Asynchronous Data Loading

- Two techniques: IFrames, XMLHttpRequest
- IFrames (Inline Frames) are an old technique given new life with Ajax
 - Invisible frame, in line with other HTML
 - IFrame owns the processing
- XMLHttpRequest: DOM extensions allowing asynchronous calls
- Issue with both: browser compatibility
 - Developer owns implementation, testing burden...
 - Or we implement using a toolkit and make it someone else's problem

AJAX Downsides

- Note these are not all specific to AJAX:
 - Browser “back” button may not work
 - Browsers record static page visits
 - Bookmarking state becomes difficult
 - Javascript generates the page
 - Increase browser code size
 - Response time affected
 - Difficult to debug
 - Processing logic both in client and server
 - Viewable Source
 - Open to hackers or plagiarism
 - Server Load
 - Asynchronous request may be an expensive operation

AJAX Toolkits

- Lots of frameworks out there
 - Commercial, open source
 - Java/.NET/PHP
 - Server-side or client-side bias
- We'll look at one framework, dojo, which fits nicely into the Java/Netbeans environment
 - Nice examples of what's possible in AJAX
 - No requirement (though straightforward implementation) for Java (or .NET, or PHP)