Asynchronous JavaScript And XML (AJAX)

Asst. Prof. Dr. Kanda Runapongsa Saikaew
(krunapon@kku.ac.th)
Dept. of Computer Engineering
Khon Kaen University
What is Common of these?

- Web sites
  - Google Maps [http://maps.google.com](http://maps.google.com)
  - Google Mail [http://gmail.google.com](http://gmail.google.com)
  - Google Search [http://www.google.com](http://www.google.com)
  - Flicks Photo Sharing [http://flickr.com](http://flickr.com)
  - Social Bookmarking [http://del.icio.us](http://del.icio.us)
Google Search with AJAX
Agenda

- What is AJAX?
- Why AJAX?
- The Rise of AJAX
- AJAX Characteristics
- AJAX Technologies
- AJAX Design
What is AJAX?

- AJAX: **Asynchronous Javascript And XML**
- An architectural style (a high level design pattern) for creating interactive web applications
- Make the Web become richer and response, closing the gap with the desktop
Why AJAX?

- HTTP never intend 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”
Why Web Application? (1/5)

- People are using different computers at home, at work, at school, in cafes, and on their phones
  - Hosting the data online is the most natural way
- Some people have no computer
  - A web application is the only practical way to use a particular application and safely retain all their data
Why Web Application? (2/5)

- Many homes and offices now have broadband and server hardware is more powerful than ever
  - Server-side storage is cheap enough for vast amounts of data to be held online
- The technologies behind AJAX – JavaScript, the DOM, and Web remoting have matured and become more standard
For developers, a modern web application is more productive than a conventional GUI alternative

- A single set of code for all platforms
- Allow upgrade the application incrementally
- Choose whatever programming languages and libraries they want to work with
Developing rich applications on the Web used to be considered a kind of rocket science

But it’s actually a lot easier now to develop for the Web - arguably easier than many GUI environments

As security concerns have heightened, companies are now quicker to lock down desktops and forbid browser plugins
Application developers are usually interesting in supporting as wide a user base as possible

A web application is often more a flexible way to target these newer platforms

One of the great strengths of the Internet is the ability to communicate and collaborate with remote users
AJAX: Change in Real-time

- They let you drag boxes instead of clicking on arrows and typing in numbers
- They keep page content fresh instead of forcing you to keep hitting Refresh button
- They show meaningful animations instead of verbose messages
On February 18, 2005, Jesse-James Garrett published an online article “Ajax: A New Approach to Web Applications”

Garrett introduced “Ajax to label the architecture behind the new generation of rich web applications like google Maps and Google Suggest”
AJAX Characteristics (1/3)

- Applications, Not Just Web sites
  - Keeps user engaged and productive
  - Socialize via the browser platform
- Smooth, Continuous Interactions
  - JavaScript makes AJAX interaction feel faster and more continuous
- Live
  - Can be programmed to always show the latest news, details on who else is online, or to send messages to the user
AJAX Characteristics (2/3)

- Supportive
  - AJAX Apps can monitor user actions and proactively support whatever task the user’s working on

- Visual Effects
  - Animation that gets across a message about what’s happening and what the user can do next

- Standards-based
  - AJAX makes the most of standard browser features and avoid browser-specific features and plugins where possible
AJAX Characteristics (3/3)

- **New Widgets**
  - AJAX Widgets go beyond the standard HTML controls
  - Sliders and progress indicators, built on standard HTML elements, are becoming popular

- **New Styles of Interaction**
  - Drag-and-drop
  - Keyboard shortcuts
AJAX Technologies

- Standards-based presentation using XHTML and CSS
- Dynamic display and interaction using DOM
- Data interchange and manipulation using XML and XSLT
- Asynchronous data retrieval using XMLHttpRequest
- JavaScript binding everything together
An AJAX App uses an HTML document to show the initial page. The document is continuously manipulated to change the display and set up new events.

Where possible, its XML-compliant variant, XHTML, should be used in order to make manipulation more robust.
CSS

- CSS enriches the display and, thanks to stylesheets, helps separate document structure from style details.
- CSS can easily be manipulated with JavaScript.
- With just one line of code, you can make an object disappear, move it around the page, or alter its appearance.
HTTP, CGI, Form Submission

- AJAX communicates via HTTP
- Instead of returning full pages, the server returns concise results that are then processed in the browser script
- Form submission is also used, no page refresh need take place
Server-Side Scripting

- The server is still required to perform tasks like data persistence and input validation
- In some AJAX applications, it no longer performs any duty of display or application logic
  - Leaving those things for the browser script to handle
JavaScript

- Client-side programming language that coordinates browser activity
- Not Java, but from the same family
  - Java-like syntax
  - Loosely-typed variables, dynamically interpreted
  - Functions are objects
    - Can be involved from outside a class
- Can use OOP-like style, but not required
The DOM exposes a web page to the JavaScript engine

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
Asynchronous Data Loading (1/2)

- 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
Asynchronous Data Loading (2/2)

- XMLHttpRequest: DOM extensions allowing asynchronous calls

- Issues: browser compatibility
  - Developer owns implementation, testing burden ...
  - Example: some web page displays successfully with IE but not with Firefox
Classic vs. AJAX Application Model

Classic web application model

AJAX web application model
Synchronous vs. Asynchronous

Classic web application model (synchronous):
- Client: user activity → data transmission → system processing → data transmission → user activity
- Time: sequential flow
- Server: system processing

Ajax web application model (asynchronous):
- Client: browser UI → display → input → display → input → display
- Time: concurrent flow with server-side processing
- Server: server-side processing
AJAX Design Principles (1/2)

- The browser hosts an application, not content
  - Application code delivered to the browser, mostly as JavaScript code

- The server delivers data, not content
  - Data may be plain text, JavaScript fragments, or XML documents
AJAX Design Principles (2/2)

- User/application interaction is continuous and fluid
  - UI metaphors like drag-and-drop become possible

- This is real coding and requires discipline
  - Significant developer responsibility to manage conversational state over entire web transaction
Why is it Popular?

- Google helped popularize, and legitimize it in Gmail
- Increase usability of Web applications
- Rich Internet applications without Flash
- Save bandwidth
- Download only data you need
- Faster interfaces (sometimes)
AJAX Downsides (1/2)

- Note these are not all specific to AJAX
  - Browser “back” button may not work
    - Browsers record static page visits
  - Increase browser code size
    - Response time affected
  - Difficult to debug
    - Processing logic both in client
AJAX Downsides (2/2)

- Bookmarking state becomes difficult
  - JavaScript generates the page
- Viewable source
  - Open to hackers or plagiarism
- Server Load
  - Asynchronous request may be an expensive operation
- Web analytics
  - Many web analytics solutions are based on the paradigm of a new page being loaded whenever new content is displayed to the user.
AJAX Toolkits

- Dojo Toolkit
  - [http://dojotoolkit.org/](http://dojotoolkit.org/)

- Google Web Toolkit
  - [http://code.google.com/webtoolkit/](http://code.google.com/webtoolkit/)

- PHP Ajax Frameworks
  - [http://ajaxpatterns.org/PHP_Ajax_Frameworks](http://ajaxpatterns.org/PHP_Ajax_Frameworks)

- AJAX : The Official Microsoft ASP.NET Site
  - [http://www.asp.net/ajax/](http://www.asp.net/ajax/)

- Prototype JavaScript Framework
  - [http://prototypejs.org/](http://prototypejs.org/)
Technologies Used in AJAX (1/2)

- Javascript
  - Loosely typed scripting language
  - JavaScript function is called when an event in a page occurs
  - Glue for the whole AJAX operation

- DOM
  - Represents the structure of XML and HTML documents
  - API for accessing and manipulating structured documents
Technologies Used in AJAX (2/2)

- **CSS**
  - Allows for a clear separation of the presentation style from the content and may be changed programmatically by JavaScript

- **XMLHttpRequest**
  - JavaScript object that performs asynchronous interaction with the server
What is XMLHttpRequest?

- JavaScript object
- Adopted by modern browsers
  - Mozilla™, Firefox, Safari, and Opera
- Communicates with a server via standard HTTP GET/POST
  - XMLHttpRequest object works in the background for performing asynchronous communication with the backend server
- Does not interrupt user operation
Steps of AJAX Operation

1. A client event occurs
2. An XMLHttpRequest object is created
3. The XMLHttpRequest object is configured
4. The XMLHttpRequest object makes an async. request
5. The ValidateServlet returns an XML document containing the result
6. The XMLHttpRequest object calls the callback() function and processes the result
7. The HTML DOM is updated
XMLHttpRequest Methods

- `open(“HTTP method”, “URL”, syn/asyn):` Assigns HTTP method, destination URL, mode
- `send(content):` Sends request including string or DOM object data
- `abort():` Terminates current request
- `getAllResponseHeaders():` Returns headers (labels + values) as a string
- `getResponseHeader(“header”):` Returns value of a given header
- `setRequestHeader(“label”, ”value”):` Sets Request Headers before sending
XMLHttpRequest Properties

- **Onreadystatechange**
  - Set with a JavaScript event handler that fires at each state change

- **readyState - current status of request**
  - 0 = The request is not uninitialized
  - 1 = The request has been loading
  - 2 = The request has been loaded
  - 3 = The request is in process
  - 4 = The request is complete

- **Status**
  - HTTP Status returned from server: 200 = OK
XMLHttpRequest Properties

- **responseText**
  - String version of data returned from the server

- **responseXML**
  - XML document of data returned from the server

- **statusText**
  - Status text returned from server
Using the readyState Property

xmlHttp.onreadystatechange=function()
{
    if(xmlHttp.readyState==4) {
        // Get the data from the server's response
    }
}

1. A Client event occurs

// file display.html

<html><head>
  <script src="clienthint.js"></script>
</head>
<body>
<form> Company Name :
  <input type="text" id="txt1" onkeyup="showHint(this.value)">
</form>
<p><span id="txtHint">
  Result will be list here.</span></p>
</body></html>
2. An XMLHttpRequest object is created

```
//clienthint.js (1/4)

var xmlHttp

function showHint(str) {
    if (str.length==0) {
        document.getElementById("txtHint").innerHTML="";
        return;
    }

    xmlHttp=GetXmlHttpObject()
    if (xmlHttp==null) {
        alert("Your browser does not support AJAX!");
        return;
    }
```

3. An XMLHttpRequest object is configured with a callback function

4. XMLHttpRequest object makes an async. request

//clienthint.js (2/4)

```javascript
var url = "getData.php" + "?q=" + str + "&sid=" + Math.random();

// result from calling stateChanged function
xmlHttp.onreadystatechange = stateChanged;
xmlHttp.open("GET", url, true);
xmlHttp.send(null);
```
function GetXmlHttpRequest(){
    var xmlHttp=null;
    try {
        // Firefox, Opera 8.0+, Safari
        xmlHttp=new XMLHttpRequest();
    } catch (e) {
        // Internet Explorer
        try {
            xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
        } catch (e) {
            xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
        }
    }
    return xmlHttp;
}
AJAX Sample 1 Application (5/8)

6. XMLHttpRequest object calls callback() function and processes the result
   - The XMLHttpRequest object was configured to call the stateChanged() function when there is a state change to the readyState of the XMLHttpRequest object

7. The HTML DOM is updated
   ```javascript
   function stateChanged() {
     if (xmlHttp.readyState == 4) {
       document.getElementById("txtHint").innerHTML = xmlHttp.responseText;
     }
   }
   ```
<?php
    // Connect to database and fetch result into array name.
    $name = array("Amazon 24689",
                   "Google 13579", "Holiday Inn 123456", "Bevery Hills 987665", "Rhyme & Reason Inc. 966640", "RRR Advertising Inc. 466136", "RIVIERA HOTELS 169706", "RIVIERA HOTELS 928748", "RRR Advertising Inc. 77165", "Yahoo Inc. 369864");
    $q=$_GET["q"]; // get the q parameter from URL
<!-- getData.php (2/3) -->

// lookup all hints from array if length of q>0

5. The Server-side script returns the results

```php
if (strlen($q) > 0) {
    $hint = "";
    for($i=0; $i<count($name); $i++) {
        if (strtolower($q) == strtolower(substr($name[$i], 0, strlen($q)))) {
            if ($hint == "") {
                $hint = $name[$i];
            } else {
                $hint = $hint."<br/>".$name[$i];
            }
        }
    }
}
```
<!-- getData.php (3/3) -->

//Show Hint
if ($hint == ""){
    $response="Not found!";
}
else{
    $response=$hint;
}

//output the response
echo $response;
?>
Company Name: 

Rhyme & Reason Inc. 966640
RRR Advertising Inc. 466136
RIVIERA HOTELS 169706
RIVIERA HOTELS 928748
RRR Advertising Inc. 77165
View display.html on IE

Company Name: 

Rhyme & Reason Inc. 966640
RRR Advertising Inc. 466136
RIVIERA HOTELS 169706
RIVIERA HOTELS 928748
RRR Advertising Inc. 77165
Dojo and JSON

- To build AJAX applications, we can choose to write our own verbose JavaScript code or use already tested and functional libraries.
- Dojo and JSON are two very different but complementary technologies.
- Both of them can significantly enhance the interface and usability of web applications.
What is Dojo?

- Dojo is a set of powerful Javascript libraries that provide a simple API to a plethora of features.

- Aside from providing Ajax functionality, Dojo also provides packages for:
  - String manipulation
  - DOM manipulation
  - Drag-and-drop support
  - Data structures, such as lists, queues, and stacks
What is JSON?

- JSON is a Java library that helps convert Java objects into a string representation.
- JSON is designed to be used in conjunction with JavaScript code making HTTP requests.
- Since server side can be written in a variety of languages such as C#, Python, PHP, and Java.
Development Tools on Mozilla Browser

- Mozilla FireBug debugger (add-on)
  - This is the most comprehensive and most useful JavaScript debugger
  - This tool does things all other tools do and more
- Mozilla JavaScript console
- Mozilla DOM inspector (comes with Firefox package)
- Mozilla Venkman JavaScript debugger (add-on)
- Mozilla LiveHTTPHeaders HTTP monitor (similar to NetBeans HTTP monitor)
Mozilla FireBug Debugger

- Spy on XMLHttpRequest traffic
- JavaScript debugger for stepping through code one line at a time
- Inspect HTML source, computed style, events, layout and the DOM
- Status bar icon shows you when there is an error in a web page
- A console that shows errors from JavaScript and CSS
- Log messages from JavaScript in your web page to the console (bye bye "alert debugging")
- An JavaScript command line (no more "javascript:" in the URL bar)
References (1/2)


References (2/2)

- Zarar Siddiqi, “Using Dojo and JSON to Build AJAX Applications”,