Web 2.0 Applications with AJAX

Shane Trahan, Yun Mai, Nanthini Ganapathi

2007 International Field Director's & Technologies Conference (IFD&TC)
Santa Monica, CA   May 20 - 23, 2007
What is Web 2.0*

- Not a new version or update to the World Wide Web.
- Web 2.0 refers to second-generation websites that imply a significant change in web usage
- Focus on the web as a platform
- Services sold on-line and accessible via a web browser
- Create rich user experiences
- Driven by increased broadband adoption and more powerful machines

*Tim O’Reilly
Characteristics of Web 2.0 Sites

- “Network is the platform”— delivering and allowing users to use applications entirely through a browser
- Rich, interactive, user-friendly interface based on AJAX or similar framework
- Architecture of user participation and democracy encouraging users to add value as they use it
- Example of Web 2.0 sites: Google Maps, eBay, Craigslist, Wikipedia
Web 2.0 and AJAX

- AJAX enriches the user experience by changing the users interactivity
- Web pages become interactive applications and have the feel of desktop application
- Ability to integrate multiple services “mash-ups” into a single service
- AJAX based applications work best in high bandwidth environments
Asynchronous JavaScript and XML (AJAX), is a web development technique for creating interactive web applications.

AJAX is a generic concept of using JavaScript to communicate directly to a server without reloading an entire web page.

AJAX is not a new technology.
Technical History of AJAX

- 1993: WWW become public
- 1996: The IFRAME tag was introduced
- 1998: Microsoft's Remoting Scripting (MSRS) introduced
- 2002: MSRS replaced with the XMLHttpRequest object
- 2005: The term AJAX was coined
- 2006: OpenAJAX initiative was formed
Pros of AJAX

- Flashy interactivity via Javascript
- No need to have pages reload
- Faster display of data
- Ability to provide more desktop-like user interfaces.
- Utilizes JavaScript and XML which is almost universally independent of platforms.
Cons of AJAX

- Lack of browser integration.
- Web analytics issues
- Reliance on JavaScript
- Bandwidth problems - too much AJAX usage will slow site down
- Difficult to bookmark sites
- Security issues
- JS tools are limited
Conventional Click-and-Wait Model

- Clients wait for pages to reload
- Content is relatively static
- Content refreshed all at once
New AJAX Interactivity Model

- Client browsers can continue to use the form while AJAX is taking care of the data retrieval and updates
- Only necessary elements are updated
- Page doesn’t completely refresh. Gives feel of desktop based application
Examples

**AJAX Example 1**

*Find information by Zip Code*

*Looks at California and North Carolina Only*

Enter a Zipcode: ________ Find Zip Reset

County: ________

City: Select City

State: ________

Zip: ________
Example 2

AJAX Example 2
AJAX and Mashup Example using Google Map API

Go to Site
Pending Cases Only

AJAX Example 2

AJAX and Mashup Example using Google Map API
Completed Cases

AJAX Example 2

AJAX and Mashup Example using Google Map API

Case Info

Male Respondent 23 years of age
Refusals / Ineligible Cases

AJAX Example 2

AJAX and Mashup Example using Google Map API

Respondent in military, not eligible
Tools and Resources

- Google APIs and Toolkits – Create interactivity between your site and Google’s online tools and services.
- Firebug – Firefox plug-in that aids in not only AJAX troubleshooting but is a great web developer tool as well
- JSEclipse – Javascript plug in for the Eclipse IDE
- AJAX Patterns, Ajax Toolkit Framework (ATF)
- AJAX toolkit for ASP.Net (Atlas and Silverlight)
Google Developer Network – http://code.google.com/
Firebug – http://www.getfirebug.com
JSEclipse – http://labs.adobe.com/technologies/jseclipse/
Microsoft Atlas – http://ajax.asp.net/
OpenAjax – http://www.openajax.org/
Future Directions

- More AJAX based web services will become available (i.e. Google Office Suite, MS Live services etc)
- Inclusion of more multimedia
- Increase in availability of pluggable widgets and easier integration
- Data consumption and remixing from all sources including user generated data
- Greater acceptance of internet based applications
Questions & Comments

Shane Trahan
Research Computing Division
RTI International
srt@rti.org
919-541-5848