Using Ajax for Desktop-like Geospatial Web Application Development

Weiguo Han, Liping Di, Peisheng Zhao, Xiaoyan Li
Center for Spatial Information Science and Systems
George Mason University
Greenbelt, USA
whan@gmu.edu, ldi@gmu.edu, pzhao@gmu.edu, xlia@gmu.edu

***Abstract*—As one of key components of Web 2.0 architecture, Ajax brings web applications with more responsive, interactive, intuitive and dynamic features. With its rich combination of technologies, Ajax provides a strong foundation for the development of geospatial web applications. An Ajax-enabled and desktop-like online geospatial analysis system is built to provide Geosciences community an easy web access to Open Geospatial Consortium (OGC) standards compliant geospatial data, information and services from multiple sources. In this system, Ajax-enabled Graphic User Interface (GUI) components, Servlet calling and web service invocations are integrated and implemented to create a better web experience for the end users. Ajax helps this data-rich and service-centric geospatial web application be increasingly used.**

Keywords-Ajax; Rich Internet Application; Online Geospatial Analysis; Open Geospatial Consortium Specification; Web Service

I. Introduction

The growing popularity of Ajax, one of the basic components of Web 2.0 architecture, leads to the delivery of appealing web applications in a completely new way which brings user interface and functionality with responsive, interactive, intuitive and dynamic features. A couple of powerful Ajax frameworks which provide suitable libraries to construct rich internet applications (RIA) are freely available for download and use by web developers. Moreover, some of them have been integrated successfully into open source JavaScript mapping libraries like OpenLayers and MapBuilder. With its rich combination of technologies, Ajax provides a strong foundation for developing next generation geospatial web application.

The GeoBrain project funded by NASA aims to provide an easy web access to geospatial data services and efficient geospatial web services from diverse sources for the Geosciences community within an effective, user-friendly and online environment. The GeoBrain research group has implemented and built an Ajax-enabled and desktop-like geospatial application, GeoBrain Online Analysis System (GeOnAS, http://geobrain.laits.gmu.edu:81/OnAS/) [1].

This paper describes Ajax use in the implementation of GeOnAS. Section 2 introduces briefly Ajax fundamentals and Ajax frameworks. We present the integration of Ajax-based rich user interfaces, Servlet calling and web service invocations in GeOnAS in section 3. Section 4 discusses how to strengthen and optimize performance of this Ajax-powered system. Finally, section 5 summarizes the lessons learned from the development of this Ajax-driven web application and gives the future direction on what need to be improved.

We acknowledge the support of NASA's Research, Education and Applications Solution Network (REASoN) program (No. NNG04GE61A).

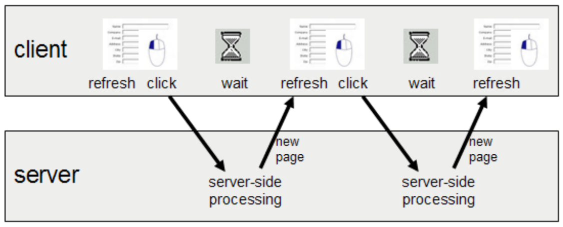
II. Ajax Fundemenetals and Frameworks

2.1 Ajax

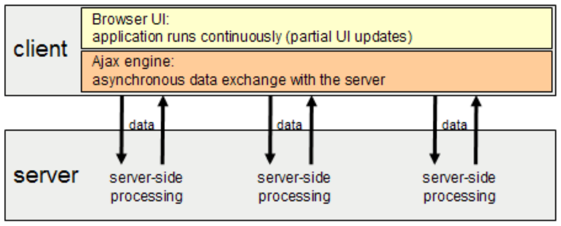
Ajax is really several familiar technologies, which are bundled together in powerful new ways [2], including Document Object Model (DOM), Cascading Style Sheets (CSS), Dynamic HyperText Markup Language (DHTML), Extensible Markup Language (XML) and JavaScript. DOM represents the structure of XML and HTML documents, and DOM APIs provide a way for JavaScript to handle the returned document from server and update the displayed page. DTHML and CSS are adopted to create the interactive and dynamic web pages. XML is for data manipulation and conversion. JavaScript is the client-side script for dynamically caching and displaying information that has been received using XML. The *XMLHttpRequest* object in JavaScript is utilized to perform asynchronous interaction with the backend server via standard HTTP GET/POST requests.

In comparison with those old web applications constrained by HTML, Ajax based applications bring user interface and functionality with rich, responsive, intuitive, interactive and dynamic features entirely inside the browser without plug-in or other software required [3]. Besides rendering HTML and executing script blocks, the browser plays a more active role in processing HTTP requests and responses in these applications. Instead of traditional "click, wait, and refresh" user interaction, these rich internet applications show better performance and web experience since they could add or retrieve users' requests asynchronously without reloading web pages. Figure 1 illustrates the interactions between client and server in classic HTML and Ajax-driven web applications.

Ajax is commonly used in the scenarios like HTML enhancement of web sites, implementation of advanced GUI widgets and controls, and development of desktop-like web applications. Ajax has led to the delivery of rich, appealing and popular web applications, such as Google Maps, Google Suggest, Gmail, Flickr, Yahoo News, etc. Currently, Ajax works well in most popular browsers like Internet Explorer, Firefox, Safari, and Opera.



(1)



(2)

Figure 1. Interaction of Classic HTML and Ajax-driven web application (source: Open Ajax Alliance [4])

Ajax related websites like AJAXIAN (http://ajaxian.com), AJAX.org (http://www.ajax.org) provide Web developers with plenty of useful technical documents, JavaScript libraries introduction, Ajax related development tools and other helpful resources. To accelerate the adoption of Ajax-based Web technologies, some Ajax vendors (e.g. Microsoft, IBM, Google, Oracle, Adobe, Eclipse, etc.), open source initiatives and developers founded an organization named Open Ajax Alliance (http://www.openajax.org/) to offer software community with both technical and marketing fronts [4]. A set of standard JavaScript functionality has been defined by this organization to solve the interoperability issues which caused by using multiple Ajax libraries [5].

Lawton discusses Rich Internet Applications (RIA) development methods including Ajax, Adobe's Flash, Microsoft's Silverlight, and Sun's JavaFX [6]. In comparison with other alternatives, Ajax is more stable, viable and flourishing so far, and presents fewer problems for clients to use.

2.2 Ajax Toolkits/Frameworks

Ajax frameworks/toolkits provide all the necessary functions for Ajax engine in server-side (e.g. Google Web Toolkit) and client-side (e.g. Dojo toolkit). In GeOnAS, server-side uses Servlets and Java Server Pages (JSP). Client-side Ajax frameworks are introduced as follows.

The browser-side Ajax frameworks provide a set of prepackaged controls, components, utilities and JavaScript APIs that help developers to build web sites and applications easily and flexibly. A good many of Ajax frameworks are available for free download and use, and the most favorite ones are ExtJS, Prototype/Script.aculo.us, jQuery, Dojo Toolkit and Yahho! User Interface Library (YUI). These frameworks make the development of RIA much more akin to that of desktop applications.

Wikipedia compares features like data retrieval, visual effects, GUI components, and license of sixteen Ajax frameworks in one specific table [7]. Chandler Project of non-profit organization Open Source Application Foundation (OSAF) also details the description, benefits and drawbacks of Ajax/JavaScript libraries for developers' reference [8]. The selection criteria of Ajax framework for the implementation of dynamic Web applications should include widget availability, file size to browser, ease of maintenance and quality of documentation [9]. In addition, quality of community support, popularity and performance are also important factors be considered.

Moreover, AJAX-enabled open source Web mapping frameworks like OpenLayers and MapBuilder are widely used to build rich Web-base geographic applications. These promising geospatial frameworks are compliant with OGC standards of Web Map Services (WMS), Web Feature Services (WFS), Web Feature Service - Transactional (WFS-T), Web Map Context (WMC) and Geography Markup Language (GML). They also support layers from Google Map and Yahoo Map.

All in all, Ajax provides a strong foundation for developing next generation web geospatial applications. In consideration of its advantages and tremendous industry momentum, the GeoBrain development team leverages the power of Ajax to implement functions of GeOnAS.

III. Building Desktop-like GeOnAS

Multiple-Protocol Geospatial Client (MPGC) [10], the predecessor of GeOnAS, was widely installed by GeoBrain project partners and users. However, it was not easily to deploy and upgrade according to their feedback. So an Ajax-enabled, browser-based and desktop-like online geospatial analysis application is developed to satisfy their requirements and meet project goal.

Graphical user interface components including *Layout*, *Menu*, *Toolbar*, *Dialog*, *Tree*, *Tab,* etc are integrated in GeOnAS which looks and behaves more like traditional Windows applications as shown in Figure 2. The appearance parameters including caption, color, font, position, images and tip of these components are loaded from the configuration files in XML format which are got from server. These components are easily initialized in web page by calling APIs from the libraries when user enters GeOnAS.

Like those components of desktop software, *Layout* helps developers to define user interface structure, and specify web page elements along with their sizes programmatically, classic layout of desktop GIS software is adopted in GeOnAS for convenience and consistency; by clicking on text and/or symbol, *Menu*/*Toolbar* offers a quick, direct and flexible shortcut of execution of each function or command, in response to the click event of HTML element, associated callback functions are used to perform an operation; *Tree* of the map layers list allows users to show or hide the selected layer and control display order, the tree list is generated dynamically via calling APIs of *Tree* component when geospatial dataset being added, and dataset information is presented in a hierarchical view; when clicking associated HTML element, popup modeless or modal *dialog* with the content of user choice enables him to input parameter values or specify other attributes for next operation; *Status Bar* with three sections displays current state of users operation, mouse position and copyright information; *tab* is used to switch between the map layers tree list and the tasks list; *Tooltip* provide helpful information regarding the element which is hovered over by the mouse cursor.

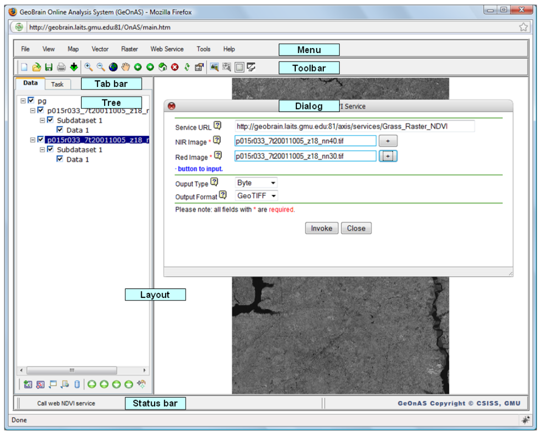

GeoBrain Online Analysis System (GeOnAS) - Mozilla Firefox
http://geobrain.laits.gmu.edu:81/OnAS/main.htm
Service URL: http://geobrain.laits.gmu.edu:81/axis/services/Grass_Raster_NDVI
NIR Image *: p015r033_7t20011005_z18_nn40.tif
Red Image *: p015r033_7t20011005_z18_nn30.tif
Ouput Type: Byte
Output Format: GeoTIFF
Please note: all fields with * are required.

Figure 2. GeOnAS User Interface

Ajax is also used to call functions implemented as Servlets or JSPs in server-side of GeOnAS. For example, it is employed to obtain the bounding box of specified location and interact with Google Map when creating new project, as displayed in Figure 3. *GetCounties* Servlet is developed to generate the counties list of selected state of the United States, and *GetBBOX* Servlet to obtain the bounding box of specified location (country, state, or county). When *XMLHttpRequest* get the correct response text from these Servlets, the counties list will be displayed in the dropdown combo box, and the bounding box values will be shown in text boxes and displayed in Google Map form.

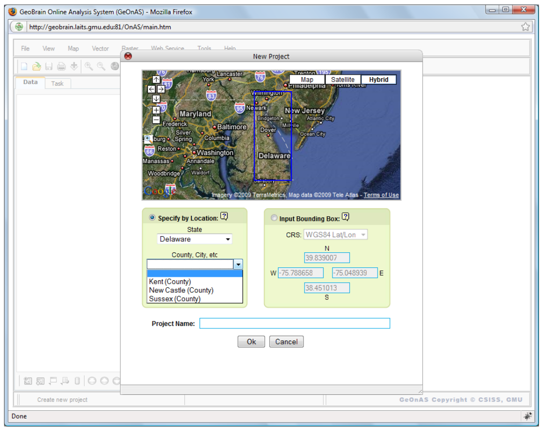
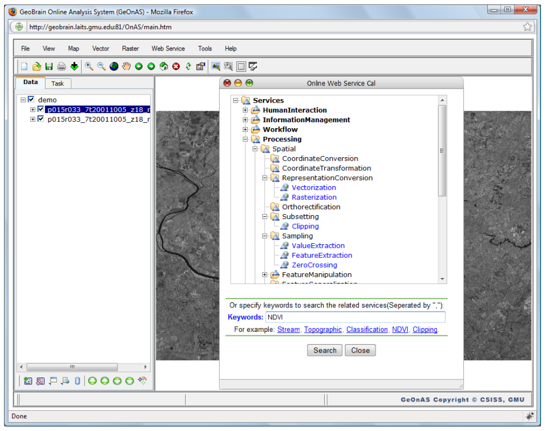
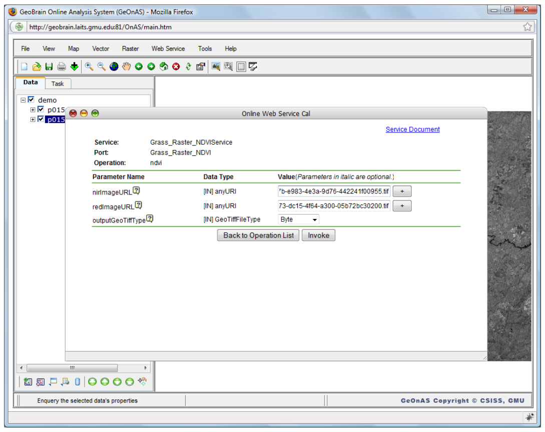

Specify by Location: State: Delaware; County, City, etc: Kent (County), New Castle (County), Sussex (County)
Input Bounding Box: CRS: WGS84 Lat/Lon; N 39.839007; W -75.788658; E -75.048939; S 38.451013
Project Name:

Figure 3. New Project Window

Moreover, Ajax is adopted to communicate with Web Service Caller Client, one middleware package which has been built to discover, select, and invoke Web geospatial services in GeOnAS. The GeoBrain team has implemented many Web geospatial services by wrapping the associated functions from Geographic Resources Analysis Support System (GRASS) [11] or developing new geospatial functions in coordination with project partners. The request and response messages are exchanged in Simple Object Access Protocol (SOAP) format between the middleware and Web geospatial services. This middleware processes the Web services results and return them as the response text of *XMLHttpRequest* to the browser client asynchronously. E-mail is sent to notify the progress or status of invocation when user exits GeOnAS. Figure 4 demonstrates the process of invoking web NDVI services asynchronously via this middleware and Ajax.


Or specify keywords to search the related services(Seperated by ","):
Keywords: NDVI
For example: Stream Topographic Classification NDVI Clipping

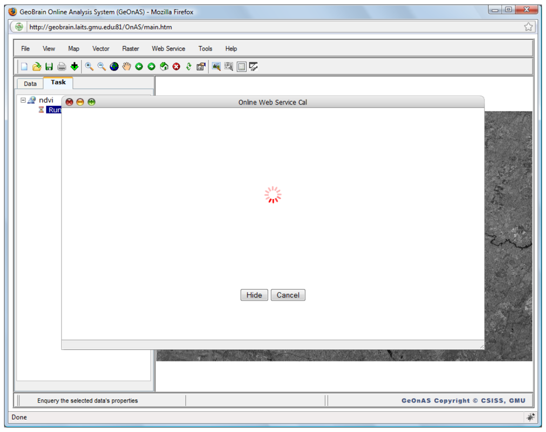
(1) Search Web NDVI Service


Service: Grass_Raster_NDVIService
Port: Grass_Raster_NDVI
Operation: ndvi

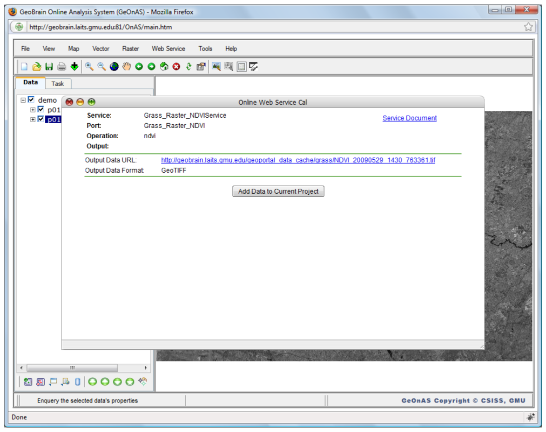
Parameter Name	Data Type	Value(Parameters in italic are optional.)
nirImageURL	[IN] anyURI	"b-e983-4e3a-9d76-44224100955.tif
redImageURL	[IN] anyURI	73-dc15-4f64-a300-05b72bc30200.tif
outputGeoTiffType	[IN] GeoTiffFileType	Byte

Online Web Service Cal
Service Document
Back to Operation List
Invoke
Enquery the selected data's properties
GeOnAS Copyright © CSISS, GMU

(2) Input Parameters

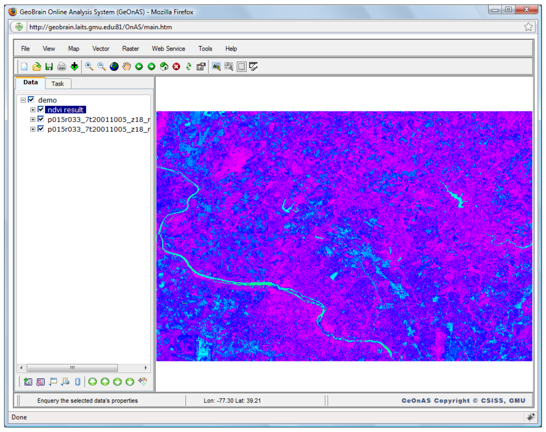

Online Web Service Cal
ndvi
Hide
Cancel
Enquery the selected data's properties
GeOnAS Copyright © CSISS, GMU
Done

(3)Wait Result


GeoBrain Online Analysis System (GeOnAS) - Mozilla Firefox
http://geobrain.laits.gmu.edu:81/OnAS/main.htm
File View Map Vector Raster Web Service Tools Help
Data Task
Online Web Service Cal
Service Document
Add Data to Current Project
Enquery the selected data's properties
GeOnAS Copyright © CSISS, GMU
Done

Service: Grass_Raster_NDVIService
Port: Grass_Raster_NDVI
Operation: ndvi
Output:
Output Data URL: http://geobrain.laits.gmu.edu/geoportal_data_cache/grass/NDVI_20090529_1430_763361.tif
Output Data Format: GeoTIFF

(4)Add Rseult


GeoBrain Online Analysis System (GeOnAS) - Mozilla Firefox
http://geobrain.laits.gmu.edu:81/OnAS/main.htm
File View Map Vector Raster Web Service Tools Help
Data Task
demo
ndvi result
p015r033_7t20011005_z18_r
p015r033_7t20011005_z18_r
Enquery the selected data's properties
Lon: -77.30 Lat: 39.21
GeOnAS Copyright © CSISS, GMU
Done

(5) Display Result

Figure 4. Web NDVI Service Invocation

IV. System Performance Improvement

System performance related issues (e.g. network transfer time, server response time, and client processing time) should be analyzed carefully when building Web applications with Ajax [12].

GeOnAS is deployed in a cluster environment with high speed interlinks which could load balance HTTP requests to achieve high scalability and improve system stability. And we will replace four old Apple Xserve G5 servers with 2GB memory and Dual 2G Hz PowerPC 970 processor with four new Apple Intel Xserve servers with 32GB memory and two 2.8G Hz Quad-Core Intel Xeon processors. These devices will improve server processing speed dramatically.

Although the use of Ajax could leads to better overall user experience, it also could come with downsides. Analysis tools like Firebug (a Firefox extension) and Yahoo! YSlow (Firefox add-on integrated with Firebug) are useful to analyze HTML, CSS and JavaScript code of GeONAS and suggest solutions to improve its performance based on a set of rules for high performance sites [13]. Firebug Lite could be installed in other browsers such as Internet Explorer and Safari to simulate some Firebug features.

According to the suggested solutions, JavaScript code, CSS and HTML are shrunk by removing all unnecessary spaces and comments from them to optimize network and runtime performance. Obfuscation code (like Google Map) is not adopted in GeOnAS because it changes the names of variables, functions and members to shorter strings that are harder to understand and debug. And JavaScript Packer is selected to decrease file size by compression. The *prototype* object of JavaScript is involved in improving the modularity of source code. In GeOnAS, WCS, WFS and WMS layer objects are inherited from this object, and attributes/methods could be easily accessed from their instances. Through these optimizations, the overall performance score evaluated by YSlow increased from F(58) to C(74). Plus we will improve those items with poor grade in the Performance Grade table.

V. Lessons and Conclusion

Lessons learned from the implementation of GeOnAS are summarized as follows:

Ajax-enabled application especially online analysis system is difficult to debug and test because of the complex interaction between browser and server. In our development process, Firebug is used as an effective JavaScript debugging tool to find and fix small issues, and JsUnit is employed in unit testing for client-side JavaScript code.

A number of Ajax toolkits from open source community provide seamless Ajax features and make it easy to develop Ajax-driven web applications. The adoption of these toolkits saves us considerable time and effort. However, JavaScript conflicts should be pay more attention when using multiple Ajax libraries in the same page. Using namespace is a good choice for this.

Since various browsers support Ajax differently, developers should make the implementations and test their accessibility for different platforms and browsers. Currently, GeOnAS is well supported in IE and Firefox, its support for other browsers is being tested and validated.

Concatenating all JavaScript and CSS files into one file could avoid the costs of multiple HTTP request. We will combine the source code following the modularity principle in the near future.

In addition, enabling gzip is another effective way to reduce text-based resources of Ajax application. And system security of Ajax application should be paid more attention.

In conclusion, AJAX offers GeoBrain users a better interactive, powerful, and responsive web experience, and it helps this data-rich and service-centric geospatial web applications be utilized by Geosciences researchers.

References

[1] L. Di, P. Zhao, W. Han, Y. Wei, X. Li, "GeoBrain Web Service-based Online Analysis System (GeOnAS)", Proceedings of NASA Earth Science Technology Conference 2007, College Park, Maryland, USA, 2007.

[2] J. J. Garret, "Ajax: A New Approach to Web Applications", http://www.adaptivepath.com/publications/essays/archives/000385.php, 2005.

[3] L. D. Paulson, "Building Rich Web Applications with AJAX", Computer, Vol. 38, No.10, pp. 14-17, 2005.

[4] Open Ajax Alliance, “Introcuding Ajax and OpenAjax”, http://www.openajax.org/whitepapers/Introducing%20Ajax%20and%20 OpenAjax.php, 2008.

[5] Open Ajax Alliance, “OpenAjax Hub 2.0 Specification”, http://www.openajax.org/member/wiki/OpenAjax_Hub_2.0_Specificatio n, 2009.

[6] G. Lawton, "New Ways to Build Rich Internet Applications", Computer, Vol. 41, No.8, pp. 10-12, 2008.

[7] Wikipedia, “Comparison of JavaScript frameworks”, http://en.wikipedia.org/wiki/Comparison_of_JavaScript_frameworks, 2008.

[8] Open Source Application Foundation, “Survey of Ajax/JavaScript Libraries”, http://chandlerproject.org/Projects/AjaxLibraries, 2009.

[9] Turner A., Wang C., Journal D., “Ajax: Selecting the Framework that Fits”, http://www.ddj.com/web-development/199203087, 2007.

[10] P. Zhao, D. Deng, L. Di, “Geospatial Web Service Client”, Proceedings of ASPRS 2005 Annual conference, Baltimore, Maryland, USA, 2005.

[11] X. Li, L. Di, P. Zhao, Y. Wei, W. Han, "Development of GRASS based Geospatial Web Services", Proceedings of the XXI Congress the International Society for Photogrammetry and Remote Sensing, Beijing, China, 2008.

[12] Zyp K. W., “Ajax Performance Analysis”, http://www.ibm.com/ developerworks/web /library/wa-aj-perform/, 2008.

[13] Yahoo! Developer Network, “Best Practices for Speeding Up Your Web Site”, http://developer.yahoo.com/performance/rules.html, 2008.