5C.4 ADDING MODERN AJAX WEB-SERVICES TO EXISTING DATA DISTRIBUTION SYSTEMS

Roland H. Schweitzer*
Weathertop Consulting, LLC, College Station, Texas

Steve Hankin, Ansley Manke
NOAA/Pacific Marine Environmental Laboratory, Seattle, Washington

Jonathan Callahan, Kevin O'Brien, Jeremy Malczyk
JISAO, University of Washington, Seattle, Washington

Jing Li
Macrostaff, Seattle, Washington

1. INTRODUCTION

The Live Access Server (LAS); Hankin (1998) is celebrating its tenth year as a freely available Web-application software system for display and analysis of geo-science data sets. The software, available for anyone to download and install, gives data providers an easy way to establish services for their on-line data holdings so their users can make plots, create and download sub-sets in a variety of formats, and compare and analyze data.

Modern web applications are taking advantage of novel applications of web technology – the so called Web 2.0 revolution. In this paper we discuss our experiences adding AJAX (Asynchronous Javascript and XML) services to the next generation of the Live Access Server. In order to build clients which give the user the utmost in responsiveness we had to carefully consider the factoring and the implementation of the services that will feed the client program the information it needs. In conjunction with the discussion of software infrastructure issues related to Live Access Server development, we will provide an overview of the latest advances in the capabilities and configurability of the Live Access Server itself.

We recently re-factored our server architecture for other purposes. The new architecture is faster at extracting and processing configuration information needed to address each request. The speed of this re-factored architecture has meant that we can use many of the underlying configuration processing classes to implement services for an AJAX client. We believe this retooling has been successful and that others will benefit from our experience.

*Corresponding author address: Roland H. Schweitzer, Weathertop Consulting, LLC, 2802 Cimarron Ct, College Station, TX 77845. E-mail: weathertop.consulting@gmail.com

2. AJAX – AN OLD IDEA WITH A FANCY NAME

AJAX, Asynchronous JavaScript and XML, is a short-hand name for collection of technologies and programming techniques that allows JavaScript running inside a browser to fetch data from the server "behind the scenes" so the user unaware of the server access. Contrast this with the other extreme of interface where the user interacts with the site via a series of "pages" where the entire browser window is refreshed after each interaction.

The concepts now commonly in use and referred to as AJAX have as ancestors concepts such as "Inner-Browsing" Gallias (2003) as the technique was called by Netscape and Microsoft's Remote Scripting.

Now that the client-side of the AJAX technology (the ability for browsers to make requests from the server via JavaScript) has standardized enough to make implementing and maintaining an AJAX application reasonable for a small group with limited resources, we decided to base our new user interface for the Live Access Server on AJAX technologies.

3. ADVANTAGES FOR THE LIVE ACCESS SERVER

Of course, we hope to gain many of the advantages in user interaction that AJAX sites offer. LAS sites will have increased responsiveness using this type of interface. Controls for selections can be refreshed without sending the user back through a series of individual pages.

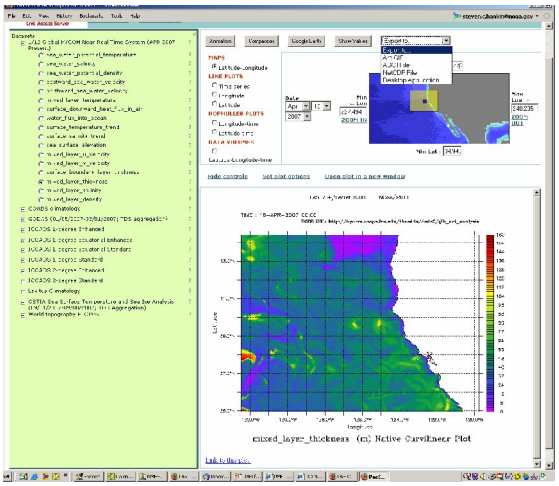

Figure 1. The new user interface.

Interestingly, once we started down this path we found that the components we constructed both on the server-side and on the client-side could be adapted to create a collection of new and novel data analysis and display products that were difficult to implement using the old technology. Section 5 has details on two of those new products.

4. OUR IMPLEMENTATION

Many AJAX implementations use XML as the exchange format. However, for our implementation we choose to stream our AJAX responses from the server as JSON, rather than XML. JSON, the JavaScript Object Notation, is a lightweight data-interchange format. The advantage of JSON when received by a JavaScript client is that the text of a JSON object can be evaluated as JavaScript code and the contents of the object can be accessed directly as JavaScript data structures.

On the server-side, all of our configuration information is ingested from XML files. The internal objects used to contain the information stores the XML as JDOM objects and exposes convenience methods for extracting the information needed by the server in the form most convenient to the class using the information (such as collections of human readable date Strings). Because there is a nearly direct translation possible between the XML and a JSON object, the job of shipping the requested information to the AJAX client is quite simple to implement and fast to run.

One word of caution, however. When translating XML to JSON some implementations will "flatten" the resulting XML when there is only one sub-element in a parent element and eliminate the parent element as redundant. To keep the data structures consistent, we had to modify this translation to JSON from XML to keep the parent element.

5. EXAMPLE PRODUCTS

As mentioned previously the new AJAX aware client-side components made it easy to implement new products for LAS.

5.1 ANIMATION

For example, we have created an animation product. When the product is first requested, we calculate some reasonable default animation parameters on the server-side, like the number of time steps, consistent contour levels to use for all frames and the size of the images. This information is delivered to the client as a list of individual product requests that it should make to build the animation. The client makes these requests via AJAX using the client-side components created for that purpose. This means that all of the error handling and other features built into the AJAX component are seamlessly reused in the animation product.

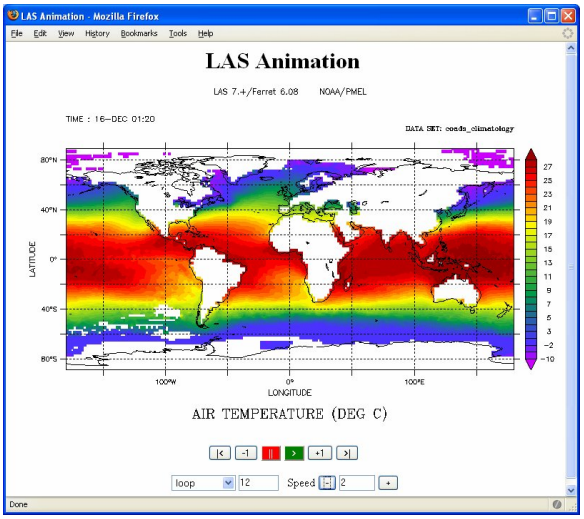

Figure 2. A snap-shot of the animation interface.

5.2 SLIDE-SORTER

A second new product that depends on the new AJAX infrastructure is a plot comparison interface we call the slide sorter. The content of each individual frame in the interface is independently controlled by selecting the value(s) for the axes orthogonal to the plot plane. Each frame refreshes itself independently by making an AJAX call to request the new plot image after value(s) of the orthogonal axes have been changed.

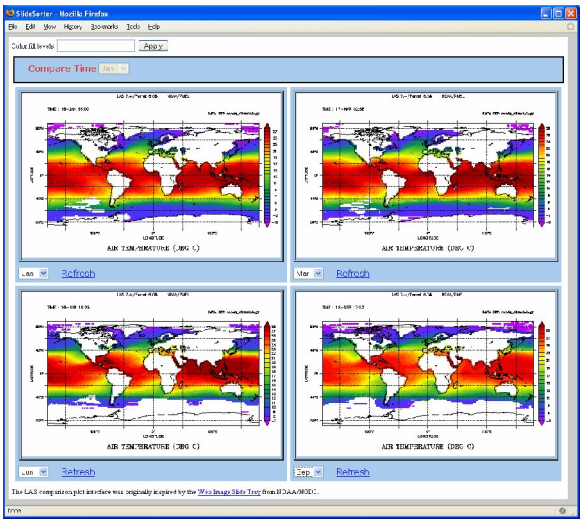

Figure 3. The Slide-sorter Interface in LAS

6. CONCLUSTIONS

Not only has the new AJAX implementation made the main user interface more robust and user friendly, it has made it easy for us to implement many new and useful data analysis and exploration products.

7. REFERENCES

Galli, M., R. Soares, I Oeschger, 2003: Inner-Browsing: Extending Web Browsing the Navigation Paradigm, http://devedge-temp.mozilla.org/viewsource/2003/inner-browsing/index_en.html.

Hankin, Steve, J. Davison, J. Callahan, D. E. Harrison and K. O'Brien, 1998: A Configurable Web Server for Gridded Data: A Framework for Collaboration, Preprints, Fourteenth International Conference for Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, Phoenix, AZ., AMS, 417-418.