9.2 TOWARDS A WEB SERVICES INTERFACE TO THE LIVE ACCESS SERVER (LAS)

Roland H. Schweitzer*, A. Manke[1], K. O'Brien[1], J. Mclean[1], J. Callahan[1], S. Hankin[2], R. Rogers[3], J. Sirott[4]

NOAA-CIRES Climate Diagnostics Center, NOAA/OAR/CDC, University of Colorado
Boulder, Colorado

JISAO/University of Washington, Seattle, WA[1]

NOAA Pacific Marine Environmental Laboratory, NOAA/OAR/PMEL, Seattle, WA[2]

Cray Research, Inc.[3]

Independent Consultant[4]

1. INTRODUCTION

The Live Access Server is a general purpose Web-server for geo-science data sets. Data providers can use the three tiered LAS architecture (see Figure 1) to build custom Web interfaces to their scientific data. Users can then access the LAS site to search the provider's on-line data holdings, make plots of data, create sub-sets in a variety of formats, compare data sets and perform analysis on the data.

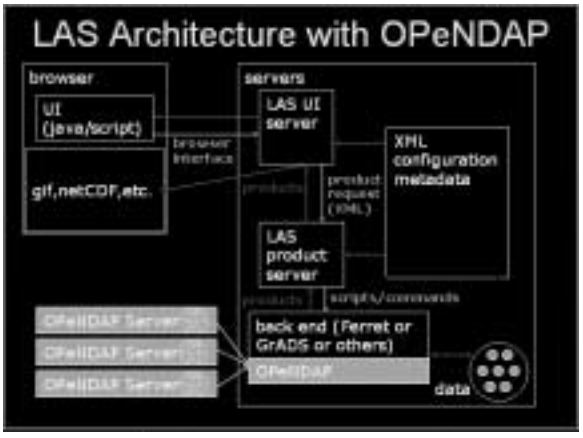

Figure 1. The LAS Architecture

An important new technology for delivering information on the Web is that of a Web service. The W3C Web Services Architecture Working Group defines a Web service as "a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols." Many Web developers extend this definition to require that the services be defined and described using the Web Services Description Language (WSDL) standard and that Simple Object Access Protocol (SOAP) be used as the communication protocol.

With the substantial number of LAS installations in place around the world, it seems natural to look for technical connections between LAS and the Web at large. The LAS development team recognized the value of having LAS servers be able operate as a Web service as defined by the Web Services Architecture Working Group.

This paper will explore the process of adapting LAS servers to operate as a formal Web service. It will give a complete technical description of the techniques used to have LAS operate as Web service in the strictest definition as well as describe the enhancements needed to satisfy the more broadly accepted definition of a Web service using the WSDL standard and the SOAP protocol.

2. The Web Service Definition Piece by Piece

In order to discuss LAS and its use as Web Service we will first take each of the key points of the Web Service definition and discuss how it relates to the current LAS implementation

2.1 Software application identified by a URI

As discussed in the introduction the Live Access Server uses a multi-tiered architecture. The piece of the architecture that interacts with the back-end analysis software to produce products for the user is known as *the product server*. The product server for each LAS installation has an associated URI. The top-tier of the architecture is the user interface server. The interactions between the UI server and the product server are all passed via this URI.

2.2 Interfaces and bindings via XML

The next requirement of the Web Services definition is that the interfaces and bindings must be described, defined and discovered via XML artifacts. In the current implementation of LAS this is accomplished via

*Corresponding author address: Roland H. Schweitzer, NOAA/ERL/CDC - (R/E/CD1), 325 Broadway, Boulder, CO 80303. E-mail: Roland.Schweitzer@noaa.gov

two XML specifications, the LAS configuration file or a THREDDS catalog.

When LAS is installed a single configuration file defines the user interface elements, operations the LAS can perform and the data the installation can interact with. A client program that wishes to interact with an LAS installation as a Web Service can request this configuration file and by parsing it can discover all of the operations the LAS is capable of performing.

A component of the NFS-funded THREDDS project is a Data Set Inventory Catalog (Davis 2002). The THREDDS catalog produced by an LAS product server also contains within it pointers to all of the data and descriptions of the operations the product server can perform.

Up to now, we can see that an LAS installation has a URI that identifies the service and has XML artifacts, via which the client can discover the interfaces and binding associated with the service.

2.3 Direct interactions… using XML-based messages

A third requirement for a Web Service is that clients wishing to use the service must interact with it using XML-based messages. The current XML architecture uses XML messages to communicate all requests to the product server. Using XML messages was a feature of the architecture before the advent of the W3C Web Services concept and now this feature can be exploited for making LAS installations into Web Services.

While requests send *to* the product server are always in the form of an XML-based message, in the current implementation results returned from the product are not necessarily XML based. As we discussed previously, the configuration XML or THREDDS catalog used to discover the bindings are XML-based messages returned from the product server, other products are returned as HTML, image data or a URL pointing to an additional service. It is this characteristic of the product server that will require the most work, in adapting the current LAS architecture to be a Web Service.

2.4 Message exchanged via Internet-based protocols

Exchanging messages via an Internet-based protocol is the final requirement for a software program to be a W3C considered a Web Service. While not specified by the definition HTTP is clearly the most commonly used protocol for Internet-based services. All of the communications between the UI server and the product server are done via HTTP.

3. Some Practical Considerations

There are several practical considerations to using LAS installations as Web services. Some of these must be addressed by the client program that is using the Web Service and others may be addresses by new LAS development work.

3.1 Using the XML-message protocol

The foremost consideration for client developers that want to use LAS installations a Web Services is the fact that the XML messages are specific to LAS. While some standards exists in related fields (such as the Open GIS Consortium's standards for Web Coverage Servers and Web Map Servers), standards for other geo-science data are still emerging.

Where ever possible, the LAS development team will work with existing standards that work well within the LAS framework. For example, it may be possible to use the Geography Markup Language as a standard way of encoding information about the location of data on the surface of the Earth. Of course, considerations like clarity, acceptance in the community and ease of use will be taken into account before any standard is adopted.

The task of learning the XML message protocol needed to communicate with LAS installations may seem daunting, but the LAS development team believes it will be worth the effort. Once the client code has been adapted to use the protocol the client can request services from any LAS installation in the world.

3.2 Discovering the contents of the server

It is likely that the first XML-message exchanged between a client program and an LAS Web Service will be a request for the XML configuration information or for the THREDDS catalog description. Once this information is received by the client the XML must be parsed and the relevant information must be extracted. Again this appears to be a complicated task, but it is made easier readily available XML parsing software. In fact, many client already have XML machinery build-in and the developer's task is limited to adapting that framework to use the new XML constructs from the configuration file or THREDDS catalog.

3.2 Improving the XML message protocol

In order to be widely adopted any Web Service must be reasonably straightforward to use and must provide clear benefits to the community that uses Web Service via a client program. In developing and enhancing the LAS Web Service framework, the LAS developers will help streamline the process for client developers where ever possible.

This will be done by using existing XML standard

grammars where applicable, improving the consistency and design of the XML message protocol and by incorporating feedback and soliciting development help from the user community.

3.3 Future development

The W3C definition does not specify that Web Services be described by the Web Services Description Language (WSDL). However, it is so common to describe Web Services in this way that it is a de facto standard. The LAS development team recognizes this and is considering if it will be possible to create a WDSL description of the LAS Web Service.

Another de facto standard of the Web Service community is the use of SOAP as the protocol to wrap the exchange of the XML messages. Again the LAS development team recognizes this as an important area for potential future development.

4. CONCLUSIONS

An LAS installation is a versatile and useful product for both data providers and scientists alike. When used as originally envisioned – as a Web-based portal to scientific data – the geo-science community has a valuable tool for the sharing and exploration of data. The addition of Web Services and the potential for new or existing client programs to use the visualization and analysis capabilities of LAS servers adds even more the value of the product to data providers and scientists interested using geo-science data.

5. REFERENCES

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

Davis, Ethan and J. Caron, 2002: THREDDS: Geophysical Data/Metadata Framework, Preprints, Eighteenth International Conference for Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, Orlando, FL., AMS