INCREASING INTEROPERABILITY WITH CEONET TECHNOLOGY USING WSDL AND SOAP

N. Grabovac

MacDonald Dettwiler, 13800 Commerce Pkwy, Richmond, BC V6V 2J3, Canada – ngrabova@mda.ca

KEY WORDS: Interoperability, Spatial Infrastructures, Databases, Software, Internet/Web, CEONet, SOAP, WSDL

ABSTRACT:

The future World Wide Web will be composed of interoperable, distributed software components called Web Services. These services will be capable of automatically discovering and invoking one another, allowing complex applications to be created from collections of interacting Web Services. Two new Web technologies that help make this possible are Web Services Description Language (WSDL) and the Simple Object Access Protocol (SOAP). A network of interoperable Web Services that dynamically interact with one another to perform a host of geoprocessing activities will likely form the technological building blocks enabling the next generation of Spatial Data Infrastructures. Although CEONet Technology already provides programmatic access to its services, this access is based on a mix of proprietary and standard mechanisms. Recasting CEONet Technology and its partners as a collection of Web Services that use WSDL and SOAP would increase interoperability in the Canadian Geospatial Data Infrastructure while reducing dependencies on proprietary technology. This paper describes WSDL and SOAP and how they can be used to transform CEONet Technology from a web application to a collection of standards-based, interoperable Web Services.

1. INTRODUCTION

The first step toward building a Spatial Data Infrastructure (SDI) involved placing databases of geospatial data online and making them accessible via standard protocols. Once there were many such databases, a service was required to find them. Under the direction of the GeoConnections program, and in conjunction with the Canada Centre for Remote Sensing (CCRS), the Canadian Earth Observation Network (CEONet) was created to serve as a directory of these geospatial data sets. Over time, it has evolved into the technology that drives the Discovery Portal for the Canadian Geospatial Data Infrastructure (CGDI), enabling registration, discovery and access to a variety of geospatial resources (Smith and Fraser, 1999; Fraser and Smith, 1998; Smith, 1998).

In addition to access via a Web browser, CEONet Technology (CT) also supports programmatic access through an application programming interface (API) and a set of Re-Usable Components (RUC). The development of CT has, as much as possible, been based on existing and emerging technologies and standards (Smith and Fraser, 1999; Fraser and Smith, 1998). The API and RUCs allow software developers to create web applications that use CT services (Katic, 1998), which is the first step toward supporting interoperability with other Spatial Data Infrastructures (SDI). As the capabilities of CT have expanded, so too has the demand to provide greater programmatic access to these capabilities. If a network of cooperating and interacting SDIs is to become a reality, the ability of applications to interoperate must increase dramatically.

The World Wide Web is evolving. It is no longer simply a massively distributed information repository. The Web is now also a platform for internet-based applications, such as the Discovery Portal, and will soon support large numbers of distributed software components, or Web Services. These Web Services are “self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web Services perform functions, which can be anything from simple requests to complicated business processes...Once a Web Service is deployed, other applications (and other Web Services) can discover and invoke the deployed service.” (Tidwell, 2000).

This vision of the Web as a collection of distributed interacting software components is one that is shared by the geospatial community (Doyle *et al*, 2001). A network of interoperable Web Services that dynamically interact with one another to perform a host of geoprocessing activities will likely form the technological building blocks enabling the next generation of Spatial Data Infrastructures. These Web Services will be capable of performing a wide variety of tasks, such as discovery of data, coordinate transformations, map and imagery visualization, etc. Powerful geospatial applications could be built by chaining together collections of such services.

The first part of this paper describes two technologies that are helping to make the Web Services vision a reality: Web Services Description Language (WSDL) and the Simple Object Access Protocol (SOAP). WSDL provides the means for a web service to describe its capabilities, location and method of invocation, whereas SOAP is a protocol for invoking Web Services.

The second part of this paper describes how these technologies can be used to transform CT from a largely monolithic internet application into a Web Services-based collection of geospatial services. Doing so would allow CT to support greater interoperability with components of the CGDI, other Spatial Data Infrastructures (SDI), and any Web applications that require the discovery and access services provided by CT.

2. CEONET TECHNOLOGY TODAY

In many ways CEONet Technology already bears some resemblance to a set of interacting Web Services. A typical set of user actions on the Discovery Portal involves: 1) discovering remote data repositories containing geospatial data of interest,

2) finding individual data products within those repositories, followed by 3) accessing that data in some manner – downloading, viewing, ordering, etc. Along the way, the user can make use of a map viewer, postal code look-up tool and place name look-up tool to refine the geographical area of interest for their search. The map viewer can also be used to display the geographic areas covered by the data products that were discovered (Figure 1).

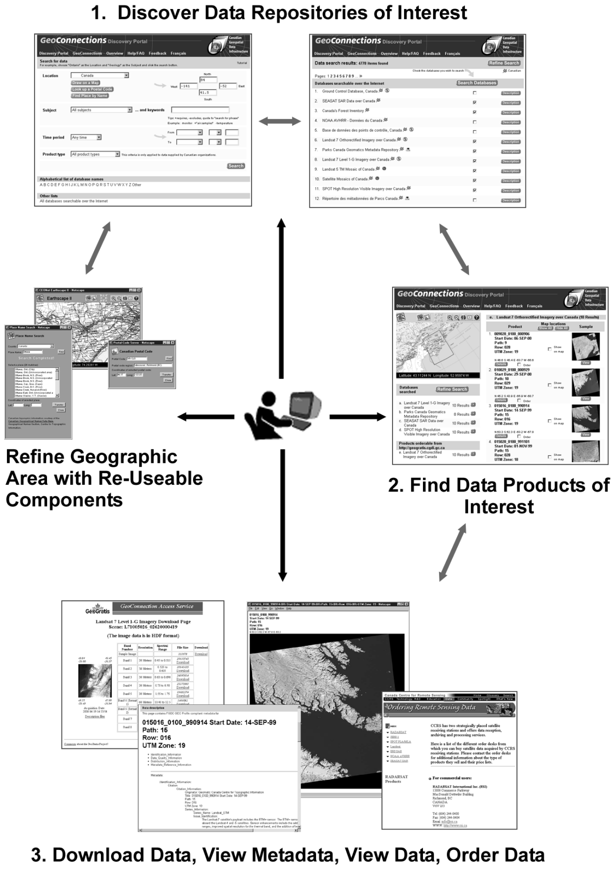
1. Discover Data Repositories of Interest
Refine Geographic Area with Re-Useable Components
2. Find Data Products of Interest
3. Download Data, View Metadata, View Data, Order Data

Figure 1. Discovery and access of geospatial data using CEONet Technology services

We can think of steps 1 to 3 as separate services that are chained together to provide the user with discovery and access services for geospatial information. The discovery services provided by CT are chained with the access services provided by the data repository. This integration of data discovery services with distributed data access services is performed using a mixture of standard and proprietary protocols.

When a data provider registers their information with the Discovery Portal, they provide the location of their data repository and the protocol to use to access their data. These connectivity details are collected and stored in a proprietary format by CT and are used to chain together the CT data discovery service with the data access services supported by the provider. Access types can include the display of browse images, thumbnail images, metadata, downloading of data, online ordering and viewing the data provider's website, all of which may potentially utilize different protocols and return their results in different formats. CT hides this complexity from the user and provides a consistent user interface (Smith and Fraser, 1999).

Web technologies and specifications have advanced to the point where a proprietary solution to chaining together disparate services, such as discovery and access, is – or will soon become – unnecessary. In order to support greater interoperability with other components of the CGDI, as well as other SDIs, these CT discovery and access services can be transformed into Web Services that are described using a standard specification and can be invoked using standard protocols. The next section will focus on two of these technologies: Web Services Description Language (WSDL) and the Simple Object Access Protocol (SOAP).

3. WSDL AND SOAP

The Web Services architecture is based on the dynamic discovery and invocation of relevant services. This requires that Web Services are both self-describing and accessible via a standard protocol. Some of the largest software companies in the world, including IBM and Microsoft (in addition to numerous smaller hi-tech companies) have jointly developed a set of specifications for this purpose. Two of these specifications: Web Services Description Language (WSDL) and the Simple Object Access Protocol (SOAP) have gained widespread industry acceptance and vendor support. As a result, they are considered de-facto standards in the Web Services community. Both WSDL and SOAP have been submitted to the World Wide Web Consortium (W3C), an organization concerned with developing standards for the Web.

3.1 Web Services Description Language

When a Web Service goes live, it is registered in a service registry, with a description of the service that includes what it can do and where and how to access it. Web Services can then use these registries to discover the other services they need to collaborate with. The invoking service uses the description of the other service in the registry to determine how to invoke that service and what response to expect in return. (Zaev, 2001). This description is specified in a standard way by using Web Services Description Language (WSDL).

A WSDL document contains sections that describe what a Web Service can do – the operations it can perform – the protocols used to access the service, and where to access the service. These descriptions are written in XML. The operations are defined in an abstract manner, without reference to the access protocol or URL used to access the operation. This allows operations to be bound to different physical access protocols – such as HTTP GET/POST or SOAP (see below) – without changing or duplicating the operational definition. The binding of the abstract operation to a concrete protocol and URL are specified in a different section of the WSDL document. A single operation can be bound to more than one protocol and access point, thus increasing flexibility in how a service can be invoked (Cauldwell, 2001; Christensen *et al*, 2001).

Descriptions written using WSDL allow an application to dynamically determine the protocol and URL for accessing a Web Service, as well as the parameters required by the operation and the format the results will be returned in. This allows Web Services to chain themselves together in real time to perform actions on behalf of a user.

3.2 Simple Object Access Protocol

Once a Web Service has been discovered and its WSDL description used to determine its capabilities, where the service resides and how to access it, the next step is to access, or invoke, the service. Although there are many possible access protocols that could be defined in the WSDL description of the service, the Web Services community has adopted the Simple Object Access Protocol (SOAP) as the standard method for invoking a Web Service.

Like WSDL, SOAP is written in XML. It defines a specification for how XML messages can be passed over a network. Because these messages can contain descriptions of operations and their parameters, SOAP can be used like a remote procedure call to invoke the functionality of a Web Service over the internet (Dix, 2001; Box *et al*, 2000).

Using SOAP as the standard protocol for accessing Web Services does not tie the user to any particular transport method. SOAP can be used over HTTP, SMTP, FTP or any other protocol for which a SOAP binding has been specified and implemented. Current implementations of SOAP use all of the above transport methods. Because SOAP was designed to be extensible, it should be able to bind to any future protocols as well.

A SOAP message consists of a body, an optional header and an envelope. The body is the payload of the message. It contains the XML document to transmit, or the details about the Web Service operation to invoke. The header is used to encapsulate any extensions to the SOAP protocol, such as transaction support, encryption, support for binary attachments, etc. It also contains routing instructions that describe the chain of intermediaries the SOAP message must pass through and what actions those intermediaries should take. The envelope serves as the container of the header and body.

The Simple Object Access Protocol allows for one-way communication, request-response communication as well as arbitrarily complex chains of communication, with multiple intermediaries between the sender of the message and the receiver. At each step in the chain, processing can take place, making SOAP an ideal protocol for constructing Web Service chains.

4. CEONET TECHNOLOGY TOMORROW

Moving from a proprietary method of service chaining to a standards-based approach using WSDL and SOAP could be accomplished as follows:

1. CEONet Technology (CT) services such as data discovery, place name look-up, postal code look-up and map viewing are described using WSDL. These WSDL descriptions are made available to other applications, perhaps using service registry.
2. The CT discovery services are SOAP-enabled, allowing other applications and Web Services to access these services using SOAP.
3. When a data provider registers their information with the Discovery Portal, they also provide a WSDL description that defines the access capabilities of the data provider, how those capabilities should be invoked, and the URL to send requests to.
4. The data provider provides a SOAP interface for invoking these access services.

Imagine the following scenario:

A market researcher needs to produce a map of housing price changes in Vancouver, BC over the last 10 years. Using a custom-built application, she enters the postal code of interest and how much she is willing to pay for the data. The application sends a message to CT asking for its capabilities. CT returns a WSDL document to the application. The application parses the document and discovers the postal code look-up service. It invokes this service to get the Lat/Long coordinates it needs to perform a search for air-photos of Vancouver in that region. It then sends a SOAP request to the CT discovery service (also described in the WSDL document) to search for air photos of Vancouver that cover the geographic area specified by the Lat/Long coordinates provided by the postal code look-up service. A SOAP response is sent back to the application, detailing the data repositories that have air photos with the required coverage, which repositories support online access to the data, how much data access costs and how reliable their access services are. The application selects a data repository based on cost and reliability criteria. At this point, the application could send another SOAP request to CT to download the air-photos (in this case, CT acts as an intermediary between the application and the data repository), or it could request the WSDL description of the data repository of interest and invoke the access service itself. The application then retrieves housing price data for the same postal code from a different Web Service. Finally, the application sends a request to a visualization Web Service to display the price data overlaid on the air photo. The resulting map is displayed to the researcher.

During this scenario, once the research application has a description of CT's capabilities as a WSDL document, it is able to invoke the services it desires without using any proprietary protocols. The application creates a service chain that uses CT Web Services to discover and access the geospatial data of interest, and uses other Web Services to combine this geospatial data with other relevant data for display to the user. A complex chain of Web Services has been created to perform an activity that has value to its user.

5. CONCLUSIONS

Web Services are changing how people and applications will make use of the World Wide Web. The increase in interoperability provided by new Web technologies such as WSDL and SOAP will also be a boon to the geospatial community and Spatial Data Infrastructures. Existing geospatial web applications, such as CEONet Technology, can be recast as Web Services that use WSDL and SOAP. Doing so makes it possible to create complex geoprocessing applications by dynamically chaining together a group of interoperable geospatial Web Services, thus moving a step closer toward implementing some of the goals envisioned the Canadian Geospatial Data Infrastructure and other Spatial Data Infrastructures.

6. REFERENCES

Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H. F., Thatte, S., Winer, D., 2000. W3C Note "*Simple Object Access Protocol (SOAP) 1.1"*, DevelopMentor, International Business Machines Corporation, Lotus Development Corporation, Microsoft, UserLand Software. http://www.w3c.org/TR/SOAP (accessed 24 Feb. 2002).

Cauldwell, P., 2001. WSDL: The Web Services Definition Language. In: Cauldwell P, Chawla R, Chopra V, Damschen G, Dix C, Hong T, Northon F, Ogbuji U, Olander G, Richman MA, Saunders K, Zaev Z, *Professional XML Web Services*, Wrox Press, Birmingham, pp. 109-147.

Christensen, E., Curbera, F., Meredith, G., Weerawarana, S., 2001. W3C Note "Web Services Description Language (WSDL) 1.1", Ariba, International Business Machines Corporation, Microsoft. http://www.w3.org/TR/wsdl (accessed 24 Feb. 2002).

Dix, C., 2001. SOAP Basics. In: Cauldwell P, Chawla R, Chopra V, Damschen G, Dix C, Hong T, Northon F, Ogbuji U, Olander G, Richman MA, Saunders K, Zaev Z, *Professional XML Web Services*, Wrox Press, Birmingham, pp. 77-107.

Doyle, A., Reed, C. (eds), Harrison, J., Reichardt, M., 2001. OGC Interoperability Program White Paper, "Introduction to OGC Web Services", Open GIS Consortium. http://ip.opengis.org/ows/010526_OWSWhitepaper.doc (accessed: 24 Feb. 2002).

Katic, N., 1998. Reusable Components In Development of Geo-spatial Software. In: *Proceedings Earth Observation and Geo-Spatial Web and Internet Workshop (EOGEO'98)* 17-19 February, 1998, Salzburg, Austria. http://www.sbg.ac.at/geo/eogeo/ (accessed 24 Feb. 2002).

Smith, S., 1998. Low Cost Distributed Geospatial Searching. In: *Proceedings Earth Observation and Geo-Spatial Web and Internet Workshop (EOGEO'98)* 17-19 February, 1998, Salzburg, Austria. http://www.sbg.ac.at/geo/eogeo/ (accessed 24 Feb. 2002).

Smith, S., Fraser, D., 1999. Rising Above a Quagmire of Search Protocols. In: *Proceedings of the Earth Observation and Geo-Spatial Web and Internet Workshop (EOGEO'99*), 9-11 February, 1999, Washington D.C., USA. http://webtech.ceos.org/eogeo99/Papers/rolker/rolker_eogeo99.html (accessed 24 Feb. 2002).

Tidwell, D., 2000. IBM devloperWorks Web services tutorial "Web services – the Web's next revolution". http://www-105.ibm.com/developerworks/education.nsf/webservices-onlinecourse-bytitle/BA84142372686CFB862569A400601C18?OpenDocument (accessed 24 Feb. 2002).

Zaev, Z., 2001. Evolution of Web Services. In: Cauldwell P, Chawla R, Chopra V, Damschen G, Dix C, Hong T, Northon F, Ogbuji U, Olander G, Richman MA, Saunders K, Zaev Z, *Professional XML Web Services*, Wrox Press, Birmingham, pp. 11-57.