Serving North American Geologic Map Information using Open Geospatial Web Services

Boisvert, E.[1], Johnson, B.R.[2], and Brodaric, B.[3]
[1]Geological Survey of Canada- Québec, 490 de la Couronne, Québec, Québec, G1K 9A9, Canada eboisver@nrcan.gc.ca; (418) 654-3705
[2]U.S. Geological Survey, 954 National Center, Reston, VA 20192, U.S.A. bjohnson@usgs.gov (703) 648-6051
[3]Geological Survey of Canada- Central, 234B - 615 Booth St. Ottawa, Ontario *K1A 0E9, Canada brodaric@nrcan.gc.ca (613) 992-3562*
Corresponding author: eboisver@nrcan.gc.ca

ABSTRACT: Open geospatial web services are the foundation of national spatial data infrastructures in Canada, the U.S., and in several other countries. Many geological data providers are experimenting with these services, though largely in isolation, leading to a wide diversity in the format of the served data. The IUGS-CGI Data Model Collaboration (DMC) is a significant international collaboration intended to harmonize the various web service efforts of national geologic data providers. In the past two years the DMC has (1) developed GeoSciML, a standard Geographic Mark-up Language (GML) application, and (2) initiated two multi-country testbeds to evaluate GeoSciML and related Open Geospatial Consortium (OGC) compliant web service technologies. In this paper we report on progress on the North American components of the second testbed, in which six countries will eventually provide geologic map information using GeoSciML. We describe an implementation from each of the GSC and USGS, and show how the national geologic maps of Canada and the U.S. are being served in the GeoSciML format. We also discuss some challenges in transforming geologic map databases, stored in commercial GIS systems, to the GeoSciML standard. In general, we find that geospatial web service technologies do not yet easily accommodate access and use of geospatial information in public data format standards, but that once such capability is achieved, typically through the development of custom middleware, the benefits become apparent.

KEYWORDS : geologic maps, web services, Geography Markup Language, Open Geosptial Standards.

1. Introduction

GeoSciML is the result of a collaborative effort that resulted from a meeting on interoperability standards in geosciences convened by the Geological Survey of Canada and the British Geological Survey in Edinburgh in 2003. At this meeting three technical teams were developed to address specific issues, and a decision was made to carry out the work under the auspices of IUGS. One of the technical teams was tasked to develop a data transfer format as a GML application to enable exchange of geoscience data. During the last 2 years, week-long workshops were held in Perth (Australia), Ottawa (Canada) and Orléans (France) to:

- evaluate existing standards and formats pertinent to information on geologic maps,
- design a model adapted from previous work, and
- initiate an OGC WMS and WFS testbed to demonstrate the application.

These workshops have resulted in the development of GeoSciML 1.1, (Sen and Duffy, 2005; Cox, 2006, see also https://www.seegrid.csiro.au/twiki/bin/view/CGIModel/WebHome) which incorporates, extends, and expands on elements found in the data models, database schema, and exchange formats used by several national geologic organisations (see https://www.seegrid.csiro.au/twiki/bin/view/CGIModel/ModelInventory). In particular, significant concepts as well as GML encodings are adapted from the North American Data Model (**NADM, 2004**) effort and CSIRO's XMML (**http://opengis.net/xmml/**). The resulting model, limited in this first version to core geologic entities such as geologic units and structures as well as boreholes, is implemented as an application of the Geography Mark-up Language (**OGC,2004**), under the name GeoSciML.

GeoSciML is being tested in North America as part of the IUGS effort, and being evaluated as a potential data delivery standard for geologic map information by the Geological Survey of Canada and the U.S. Geological Survey. The need for such a standard is quite profound in North America, as there are more than 60 independent geological survey agencies producing data in federal, provincial, and state organisations. Each organization has its own requirements resulting in heterogeneous management of the information, as well as varied structures and content; e.g. while virtually all organisations produce the information as maps, the attributes associated with features on the map as well as the vocabularies that populate the attributes, vary considerably within and between organisations. The software platform used to distribute this information also varies, with vendor specific solutions such as ESRI's ArcIMS often utilized, though a trend is emerging in which the map information is provided using open geospatial web services such as the OGC WMS, WFS and WCS standards. Significantly, these standards rely on GML as a means of transferring information and describing the structure and content of the data. In the context of a data network in which data providers agree to deliver their data using a common schema from these web services, there arises the issue of transforming the schema from the local data provider to the schema of the common public GML format, e.g. GeoSciML. In this paper, we demonstrate our approach to this transformation. We describe the transformation mechanism and demonstrate how we deliver geologic map data in the common GeoSciML format from two WFS sources, one hosted at the Geological Survey of Canada (GSC) using the University of Minnesota MapServer technology, and the other hosted at the United States Geological Survey using ArcIMS over ESRI shape files.

2. Translating data from private schema to public schema

Although ArcIMS and MapServer both offer WMS and WFS support, and therefore can be accessed using the same web service interfaces, the content of the data repositories using these services are quite different, both in terms of structure and content. To achieve interoperability, both the schema and content must be converted from the private schema into a public schema (here GeoSciML). The translation is not limited to translating the outgoing data into GeoSciML, but also translating a query expressed in GeoSciML and sent by some client requesting data. The complete process of extracting data from the repository is decomposed into 2 steps: (1) the query from a client is translated from the public GeoSciML schema into the private schema of the local data and the request is executed by the local data store, and (2) the data resulting from the query is translated from the private schema to the public schema and delivered to the client. The core of the problem is how to express and execute this translation in a web service environment. Two approaches are regularly adopted for the placement of the translation layer within the system architecture: the translation layer is placed between the web service and the data store, or it is placed between the client requesting data and the web service delivering the data. Here we have implemented the second approach.

3. The Cocoon solution to data translation

Our implementations insert middleware wrapper software between the client and the server, which intercepts both client requests and server results and applies the necessary translations. We use Apache Cocoon (http://cocoon.apache.org/) as the middleware platform. It is an open source application development framework that connects applications via transformation steps using a logical construct called a pipeline. Alternative technologies exists, such as FME, but we chose Cocoon basically it's open source and allow a broader community to use the implementation. Cocoon is geared specifically to the processing of XML documents and offers various transformation components, the most common being W3C XSLT (**http://www.w3.org/TR/xslt**). One can then build a public-private translation mechanism by invoking XSLT transformations with in a specific pipeline. In our implementation, a first XSLT is applied to the incoming client request, the transformed result is sent to MapServer or ArcIMS expressed in a private schema where it is executed by the local store. A second XSLT is applied to the results before they are returned to the client. This solution is implemented identically in both USGS and GSC situations, with the only difference being the specific XSLTs generated for the unique local data stores. At the GSC, the Cocoon wrapper envelopes both ArcIMS and MapServer (WMS services are implemented in ArcIMS and WFS in MapServer, for performance reasons). At USGS the whole system is implemented in ArcIMS. On query, the MapServer client queries are translated to WFS web service calls, and the ArcIMS client queries are translated to ArcXML queries. On delivery, the MapServer GML is translated to GeoSciML, and the ArcXML is translated to GeoSciML.

4. Example

Cocoon pipelines are encoded in a special XML document called a sitemap, designed to intercept an incoming request and direct it to the correct pipeline. The example shown in Figure 1 is only a small portion of the GSC pipeline. In Figure 1, Step 1 is a generator that feeds the XML input into the pipeline. This specific generator reads the XML from the POST buffer (WFS GetFeature can be sent either in POST of GET; we show only the POST pipeline for brevity).

Simplified GetFeature pipeline

```
<map:match pattern="wfs/GetFeature">
  ❶<map:generate type="xmlStream"/>
   <!-- turn public query into private query -->
  ❷<map:transform src="style/gs2ms.xslt"/>
  ❸<map:transform type="PostClient">
      <map:parameter name="server" value="
http://s5-stf-ngwd.nrn.nrcan.gc.ca/WebScripts/mapserv.exe?request=GetCapabilities&amp;servi
ce=wfs&amp;map=\ngwd\map\bedrock\bedrock.map"/>
   </map:transform>
  ❹<map:transform type="stx" src="style/gsmlunit_ms.stx"/>
  ❺<map:serialize type="xml"/>
</map:match>
```

Figure 1: a fragment of the GSC pipeline.

The following step (2) is a XSLT transformation (the default transformer) that applies a stylesheet (gs2ms.xslt) to the incoming request. The result is streamed to the next transformer (3) a PostClient transformer, which is a custom made transformer written in Java. The sole role of this transformer is to catch the incoming XML and send it to a server identified in a parameter passed to it. In the GSC implementation, the query is then sent to a WFS service, in the USGS implementation the query is sent to ArcIMS using ArcXML. The result is then parsed and sent down the pipeline to the next transformer. This last step (5) is not an XSLT transformation, but is an STX transformation (Streaming Transformations for XML, see http://stx.sourceforge.net/documents/spec-stx-20040701.html), which is a variant of XSLT specially designed to deal with large XML streams. STX transformations are applied on streams of XML entities, as opposed to XSLT which applies to a DOM (Document Object Model) which requires the whole document to be loaded in memory Although STX has limitations, it works perfectly with large lists of isolated XML fragments, which is typically what is returned from a WFS service. The benefit of STX is a small memory footprint on the server, as large XML documents are not uploaded in memory prior to transformation.

5. Conclusions

The Cocoon solution we built does not cover all the syntactic richness of WFS at this point, but does address the most common WFS queries that minimal clients typically send. This includes identifying the feature type name, the bounding box and the maximum number of features to be returned. Adding more functionality is a matter of writing a more sophisticated XSLT transformation, which is not a trivial task, but which would be complex in any environment. The main benefits of this approach are the ease of maintenance of the architecture (it’s only a matter on maintaining XSLTs/STXs), the quick development and prototyping time.

References

Cox, S.J.D., Boisvert, B., Brodaric, B., Duffy, T.R., Johnson, B.R., Laxton, J.L., Richard, S.M., and Simons, B., 2006. GeoSciML: a standards-based encoding for transfer of geoscience information from IUGS/CGI. These Proceedings.

North American Geologic Map Model (NADM) Steering Committee, Data Model Design Team (DMDT), 2004 : NADM Conceptual Model 1.0, A conceptual model for for geologic map information. Geological Survey of Canada, Open File 4737, U.S. Geological Survey Open File Report 2004-1334, 1 CD ROM (http://nadm-geo.org/)

OpenGIS Consortium, 2004, OpenGIS Geography Markup Language (GML) Implementation specification version 3.1, Cox et. al., ed.: OGC Document 03–105r1, accessed at http://portal.opengis.org/files/?artifact_id=4700.

Sen, M. & Duffy, T., 2005. GeoSciML: Development of a generic GeoScience Markup Language. *Computers and Geosciences* (31): 1095-1103.