Using Semantic Web Technologies to Streamline the Implementation of the OGC Web Service Interface Specifications for Coverage and Feature Data within OPeNDAP

Daniel Holloway1 (d.holloway@opendap.org), Benno Blumenthal2 (bennorn@irri.columbia.edu), Nathan Potter2 (ndp@opendap.org), Patrick West3 (pwst@ucar.edu)

1 OPeNDAP Inc., 2 IRIS/Columbia University, 3 NOAA/UCAR

I. Abstract

The OPeNDAP Data Access Protocol has seen widespread adoption within the science community. OPeNDAP servers are currently deployed by individual investigators, academic institutions, and national and international data systems to provide distributed data access for their respective user communities. Many of these data providers anticipate that there will be significant demand for data access by applications using the suite of OGC web service specifications. Supporting multiple data access protocols can be expensive, both in the initial acquisition and deployment cost for the software components as well as for the potentially redundant maintenance and security costs required when supporting several server implementations concurrently. To provide a cost-effective solution to these problems, OPeNDAP is developing extensions to its data access protocol to enable the use of semantic web technologies for data and metadata transformations, and extensions to its server architecture to support request and response operations simultaneously for multiple data access protocols.

The OGC Web Coverage Service Interface Specification is the initial data access protocol to be layered onto the OPeNDAP server for this multi-protocol support. Supporting data access through the OGC service interface compromises operations that are both mechanical and semantic. The OPeNDAP server architecture (fpCache) uses a Lightweight Front-end Server (LFS) that is responsible for interacting with the requesting client application. The LFS is extensible and in this project has been extended to support the OGC web service interface specifications.

Coalesped with the LFS the fpCache architecture uses a Back-End-Server (BES) to do data access, processing, and response generation that are then returned through the LFS to the requesting client. Similar to the LFS, the BES is extensible and for this project has been extended to support various mechanical operations required in support of the OGC service’s request and response interface specifications. In addition to the simpler, mechanical aspects of the OGC service requirements, the LFS has been extended to support multiple protocols, semantic operations are required in order to interpret request elements and for constructing well-formed OGC response elements. To support these semantic operations, we’ve developed representations of the OGC, OPeNDAP and NetCDF data models, and the relationships between these models. The OPeNDAP has been extended to support XSLT operations transforming OPeNDAP’s XML data descriptor (DDX) representation to a Human Readable Description Framework (HDF) representation. Modules executing during server initialization ingest the RDF representations and use the ontologies to crosswalk the remaining elements between the protocols.

II. Methodology

III. Semantic Mapping

In order to have geo-located objects, we need additional semantics on top of netCDF, e.g. the CF conventions. NetCDF-3, and netCDF-4 have multi-dimensional arrays with attributes, one-dimensional mapping vectors. NetCDF-3 has multi-dimensional arrays with attributes, one-dimensional mapping vectors, and common dimension names, which form the corresponding netCDF objects. These objects usually detail an implicit, but the netCDF-object model gives explicit name/value associations to these parts of the netCDF conventions, and it is that higher-level objects which correspond to objects in other conventions.

In order to have geo-located objects, we need additional semantics on top of netCDF, e.g. the CF conventions. NetCDF has a special attribute “Conventions” to name the convention that governs the meaning of the attributes in a particular file, and the corresponding netCDF convention for these data elements, the cfobject model describes the convention at the attribute level, i.e. which attributes are defined, and what are their possible values. However, these attributes imply higher-level objects (e.g. geo-located objects), which are given explicitly in the cfoef model, along with the rules for setting these objects given the information in the cfoed model.

RDF is sufficiently flexible to be able to hold both the information in a DAP DDX document, and the information in a WCS Capabilities XML document. In this diagram we show the semantic path to convert DAP DDX documents into WCS Capabilities.

We organize the ontology documents into three classes: objects, models and crosswalks. Objects are collections of entities, e.g. DAP datasets or WCS Covers. Models are the structure (schema) for these entities, e.g. OPeNDAP or WCS. And crosswalks are the connections between those structures, containing rules for expressing objects in another model’s frame (the model that a document is built from).

In the figure we have two sample datasets (EUMWF ERA-40 dataset and 208003010009_001_000_030_030_38_300 which are structured according to the DAP Model. The goal is to use the multi-protocol support to return WCS Server Responses.

IV. DAP Internals

V. Server Internals

VI. Development Plan

We can now generate WCS Server Responses.

Changes:
- **RDF response:** A new DAP service, the RDF response, has been introduced. A request for this response will return an RDF representation of the data.
- **DDX response:** In support of the RDF representation of the data, we added a DDX response (which has been available from fpCache for several years) has been extended by adding both WMT and WCS metadata attributes to the DAP Dataset element. This version of the DDX is only available to clients which indicate that they accept DAP version 3.2. The client does this by setting the HTTP header XDAP-Accept, “XDAP-Accept: 3.2”.

Proposed Changes:
- Allow XML content from other namespaces as semantic metadata in the DDX, essentially allowing the DIX to be more flexible.
- In order to maintain DDS usability some combination of rules may need to be created:
  - Elements of type dayBlueType (data set variable) may contain elements of other namespaces.
  - Elements of type dayBlueType (data set variable) may not be the children of elements in another namespace and be as part of the top level dayBlueType element’s regular DAP semantics.

Acknowledgements

Funding for this effort is provided by NOAA’s National IOOS Development effort initiated in 2007. This effort initiated a competitive funding process to continue building capacity for regional ocean observing systems, and in particular the effort contributes toward developing applications for regional stakeholders, and enhancing regional and national data management and communication protocols.

References

