

CEOS WGISS: Interoperability between OGC CS/W and WCS Protocols

Wenli Yang, George Mason University

Min Min, George Mason University

Daniel Holloway, OPeNDAP.org

Yonsook Enloe, SGT, Inc.

Christopher Lynnes, NASA/GSFC

DRAFT! Sept 23, 2008

Background

In the past several years, interoperability gaps have made cross-protocol and cross-community data access a challenge within the Earth science community. One such gap is between two protocol families developed within the geospatial and Earth science communities. The Earth science community has developed a family of related geoscience protocols that includes OPeNDAP for data access and the Thematic Real-time Environmental Distributed Data Services (THREDDS) catalog capability. The corresponding protocols in the geospatial community are the Open Geospatial Consortium (OGC) protocols Web Coverage Service for geospatial data access and Catalog Services for Web (CSW) for data search. We have developed a catalog gateway to mediate client/server interactions between OGC catalog clients and THREDDS servers.

CSW Server Implementation

Catalog Services for Web(CS/W) is designed for providing catalog service for clients to find the needed data and services. CS/W specifies the interfaces, bindings, and a framework for defining application profiles required to publish and access digital catalogues for geospatial data and services. The CSW specification does not require the use of a specific catalogue schema. However, it encourages the adoption of standard schemas for maximum interoperability. Specifically, OGC developed two application profiles for CSW: the ISO19115/19119 profiles and the ebRIM profile. The ISO19115/19119 profile explains how catalogue services based on the profile are organized and implemented for the discovery and management of geospatial data and service metadata which are compliant with the ISO19115 and 19119 standards. The ebRIM profile explains how services based on the

CEOS WGISS: Interoperability between OGC CS/W and WCS Protocols

more general OASIS ebXML Registry Information Model are organized and implemented. CSISS CS/W is compliant with OpenGIS Catalogue Services Specification 2.0.2 -ISO Metadata Application Profile. It specifies an application profile for ISO 19115/ISO 19119 metadata with support for XML encoding per ISO/TS19139 and HTTP protocol binding. Currently, it supports OGC_Service.GetCapabilities, CSW Discovery.GetRecords, CSW Discovery.DescribeRecord, and CSW Discovery.GetRecordById. A CSW client starts with sending a GetCapabilities request to the server and getting a response showing the capability of CSW server. The client then constructs a GetRecords request, following the specification, based on the inputs of the user (who is using the client) and sends the request to the server. Based on the GetRecords request, the server generates a response xml and sends it back to the client.

Before the ingestion tool, or ingestor, and the server can be implemented, a metadata mapping scheme must be developed. The ingestor reads the THREDDS catalog and converts related metadata items into ISO19115 counterparts based on the mapping scheme and ingests them into the CSW server database. The CSW server provides either real-time or pre-stored THREDDS catalog information to the clients. The following flow of process is therefore involved in the middleware:

1. It ingests THREDDS catalog information to CSW server, pre-configured schedule.
2. It receives a request in compliant with CSW specification from a CSW client.
3. It searches the CSW database which includes ingested THREDDS catalog information.
4. It sends the translated response to the requesting CSW clients.
5. The Ingestor includes four components: Ingesting Component, Parsing Component, Mapping Component, and Registration Component.

The role of Ingesting component is a THREDDS client: it sends a request to the THREDDS server, and sends the response to the Parsing component.

The parsing component is responsible for parsing the catalogue content. The THREDDS catalog is hierarchical. There are two types of datasets in the THREDDS catalogue, direct dataset and dataset collection. A dataset without a nested "dataset" is a direct dataset, otherwise, it is a dataset collection. "catalogRef" element also be parsed as a dataset collection. If a "catalogRef" element is found in the processing catalog document, it will generate a new catalogue xml URL reference, and then give this URL back to Ingesting component, and Ingesting component sends a new request to the server for

CEOS WGISS: Interoperability between OGC CS/W and WCS Protocols

the corresponding new catalog xml document. If "dataset" element is found, it pass the metadata to Mapping component.

One THREDDS catalog may, and often does, contain many thousands or even more individual datasets. The ingestor was designed and implemented to do hierarchical ingestion. It ingests more stable, but much less, data collections at a pre-configured regular and relatively longer time period, such as daily and weekly. The more frequently updated datasets can be ingested at shorter time period such as hours or minutes. In our current integration with the Unidata THREDDS catalog, we adopted on-demand ingesting for datasets, i.e., performing ingesting only when a client wants to search inside a particular datacollection, at which point the datasets under the needed datacollection are ingested real-time.

Mapping component translates the metadata from the schema of THREDDS InvCatalog to the schema that compliant with ISO 19115.

Registration component registers the metadata it into CS/W database.

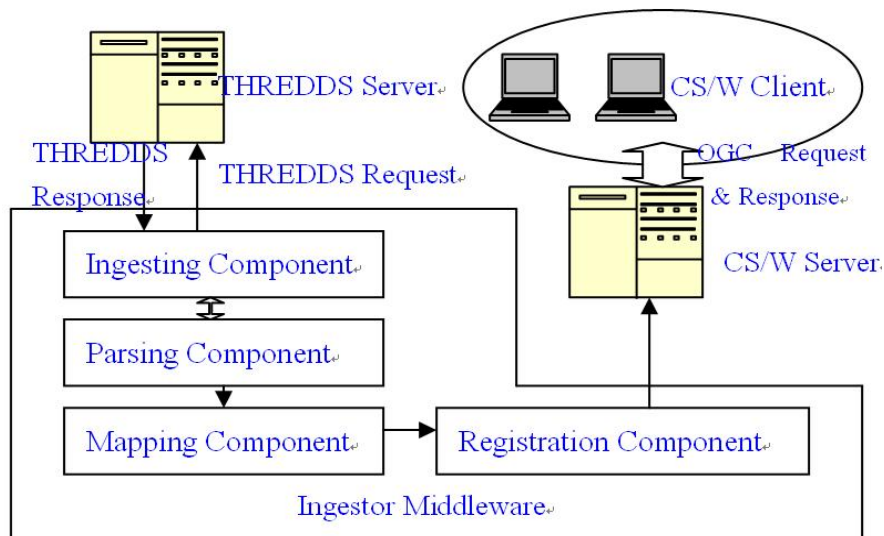


Figure 1. Architecture of Ingestor

Interoperability Challenges

The OGC Catalog Services for the Web (CS/W) interface can work with a variety of metadata profiles. A catalog system implementing the OGC CS/W interface standard must also implement a metadata profile. The common metadata profiles being used with the CS/W include the EbRIM family and

CEOS WGISS: Interoperability between OGC CS/W and WCS Protocols

the ISO 19115 family. Because there is a proliferation of metadata profiles that can be used with the OGC CS/W standard, interoperability between a CS/W Server and Client, developed independently, is difficult. For meaningful, basic searches to be performed, the CS/W client and the CS/W server must understand the same mandatory set of metadata attributes. For comprehensive, detailed searches to be performed, the CS/W client and the CS/W server must agree on a similar set of optional metadata attributes. This requires coordination and agreement between the CS/W client and CS/W server developers.

Deep Dataset Hierarchies

THREDDS servers nearly always provide access to collections of datasets so the THREDDS technology allows for catalogs of catalogs. For example, the Unidata "motherlode" prototype server serves real-time output of several different NCEP weather forecast model runs (among them are the Rapid Update Cycle or RUC, North American Model or NAM, Global Forecast System or GFS). Each of these models are run at regular intervals ranging from every hour to twice a day. THREDDS catalogs are available at various levels:

- Top level catalogs the different types of datasets available (e.g., NCEP model output, radar, station obs, radar scans, satellite images):
<http://motherlode.ucar.edu:8080/thredds/catalog.html>
- In the NCEP branch, the next level catalogs different models (e.g., RUC, NAM, GFS, NDFD)
<http://motherlode.ucar.edu:8080/thredds/idd/models.html>
- The next level catalogs the output options for a specific model such as the NCEP-NAM-CONUS_12km-conduit (e.g. forecast model run or individual "file access")
http://motherlode.ucar.edu:8080/thredds/catalog/fmrc/NCEP/NAM/CONUS_12km/conduit/catalog.html
- For the "file access," there is an inventory list of the grib files with about 90 such files on motherlode for NCEP-NAM-CONUS_12km-conduit runs over about 3 weeks.
http://motherlode.ucar.edu:8080/thredds/catalog/fmrc/NCEP/NAM/CONUS_12km/conduit/files/catalog.html
- For any one of those NCEP-NAM-CONUS_12km-conduit runs, there are four access method options (OPeNDAP, HTTPServer, WCS, NetcdfSubset)
http://motherlode.ucar.edu:8080/thredds/catalog/fmrc/NCEP/NAM/CONUS_12km/conduit/files/catalog.html?dataset=fmrc/NCEP/NAM/

CEOS WGISS: Interoperability between OGC CS/W and WCS Protocols

CONUS_12km/conduit/files/
NAM_CONUS_12km_20080820_0600.grib2

This hierarchy of catalogs begs the question of which level should be harvested for discovery in the CS-W search facility. At the top level, there are only 7 categories of data in the catalog on motherlode. However, these branch out into thousands of inventory, leaf node possibilities. If one chooses to list the catalogs at one of the higher collection levels for searching in the CS-W in order to avoid returns with hundreds of hits, it then is necessary to be able to drill down from that level to the data access level where a data access interface such as WCS is available. And, even if that capability is provided in the CS-W server, it is not useful unless clients are able to take advantage of it and make it possible for the user to drill down via the client interface.

As noted below in the "Last Mile Problem" section, there is the added challenge that users may actually be searching for specific fields (e.g., vorticity) in any given forecast. Addressing that possibility compounds the granularity issue in that the various models output a set of fields that numbers in the many dozens. If each such field in each dataset is viewed as a separate coverage, one can envision a simple CS-W query that would result in thousands or tens of thousands of hits in terms of individual coverages for different model runs and forecast times on a given server.

Freetext Search

Freetext searches are by far the most common search types on the Internet today (cf. Google). One of the keys to their popularity is the utter simplicity: there is no data model to learn in order to use it, just a single field. However, the CS/W search/response protocol is highly structured, with a number of fields containing potentially searchable text. As a result, there is much ambiguity as to which of these many fields should be searched by the client when presenting a simple freetext field to the client. Some clients (e.g. ESRI) search only the title; others search [which fields does GI-GO search?] As a result, the same keywords in two separate clients may generate different results even when submitted to the same server.

"The Last Mile Problem"

The ultimate goal is to return a search result that a client can use to issue a Web Coverage Service request, thus taking advantage of the WCS capabilities of the THREDDS Data Server. This should be relatively transparent to the user, without the need to cut and paste URLs. Initially, the THREDDS metadata were inventoried at a server level, meaning that the CSW server provided enough information for the client to issue a GetCapabilities request, followed eventually by a GetCoverage request. However, the GetCapabilities response overlaps

CEOS WGISS: Interoperability between OGC CS/W and WCS Protocols

significantly with the CSW response in information content, forcing the client to essentially repeat a significant amount of search work. For example, if the user searched on, say, "vorticity", this response would force the user or the client to search through the GetCapabilities response (or a subsequent DescribeCoverage response) in order to find the vorticity coverage. One alternative is to instead inventory at a Coverage level, so that a client could immediately issue a GetCoverage request. Enrico Boldrini and Lorenzo Bigagli of the Earth and Space Science Informatics Laboratory suggested that sufficient information be provided in the fields of the ISO 19115 CI_OnlineResource element to allow the client to issue a specific direct GetCoverage request without having to use additional information, as the followings:

- linkage -> the WCS endpoint (without request parameters)
- protocol -> something like "**OGC:WCS-1.0.0**" or "**OGC:WCS-1.1.1-get-coverage**", following the "style" NAMESPACE:PROTOCOL-VERSION-OPERATION used in the examples from OGC: <http://schemas.opengis.net/csw/2.0.2/profiles/apiso/1.0.0/examples-ISO19139/collectiondata4ESA.xml> and in GeoNetwork: <http://trac.osgeo.org/geonetwork/wiki/ISO19119impl>.
- name -> the name of the coverage
- description -> brief description of the coverage
- function -> download

Currently, the WCS access urls in the THREDDS catalog include getCapabilities but not getCoverage requests. Thus a WCS client component is implemented in the CS/W server. This WCS component performs the following functionalities: a) sending a getCapabilities request for a specific dataset using the WCS url provided by THREDDS; b) parsing the getCapabilities response; c) construct a describeCoverage request based on the getCapabilities response; d) parsing the describeCoverage response; e) construct a getCoverage request based on the describeCoverage response; and finally f) place the getCoverage request in the linkage field in the CI_OnlineResource. With such information, the user of a CS/W client can directly download a specific dataset or a specific variable in a multi-variable dataset.

Field	Description	Example
linkage	WCS endpoint (without request parameters)	[wcs getCoverage request]
protocol	Protocol and version	OGC:WCS-1.1.1-get-coverage
name	Name of the coverage	
description	Brief description of the coverage	
function	"download"	"download"