

Testbed-12 WCS Profile Update Engineering Report

Table of Contents

1. Introduction	7
1.1. Scope	7
1.2. Document contributor contact points	7
1.3. Future Work	7
1.4. Foreword	7
2. References	9
3. Terms and definitions	10
3.1. DAP	10
3.2. NetCDF	10
3.3. OPeNDAP	10
4. Conventions	11
4.1. Abbreviated terms	11
5. Overview	12
6. Status Quo & Annotated Requirements Statement	13
6.1. Status Quo	13
6.2. Annotated Requirements Considered by the OPeNDAP and EOxServer Teams	13
6.3. Targeted Solutions	15
7. WCS support for OPeNDAP	17
7.1. WCS/OPeNDAP Comparison	17
7.1.1. Comparative examples of the OPeNDAP and EO-WCS Coverage representations	18
7.2. Extension for WCS <i>DescribeCoverage</i>	29
7.2.1. Coordinate Dimensions Beyond Latitude and Longitude	29
7.2.2. Time Is Another Dimension	33
7.2.3. Order of indices, order of coordinate dimensions	35
7.3. Extension for EO-WCS <i>DescribeEOCoverageSet</i>	36
7.3.1. Issue: Missing Range Of Results API	36
7.3.2. Issue: Search Requires Prior Knowledge Of Content.	36
7.4. Extension for WCS <i>GetCoverage</i>	37
7.4.1. Scaling Extension [OGC 12-039]	37
7.4.2. Regarding SlicePoint Subsetting Restrictions	38
7.4.3. Data Request API Examples	38
7.5. Demonstration	41
7.5.1. Index base subsetting	41
7.5.2. Value based subsetting	43
8. EO-WCS Update to Support 3D Coverages	46
8.1. EO-WCS Update	46
8.2. Use Case - Fire Emissions, San Francisco	46
8.2.1. Data	46

8.2.2. <i>GetCapabilities</i> Request	46
8.2.3. <i>DescribeEOCoverageSet</i> Request	47
8.2.4. <i>GetCoverage</i> Request	50
8.2.5. Demonstration	51
9. WCS PYXIS Client Testing Discussion	52
9.1. Additional metadata information on WCS <i>GetCapabilities</i>	52
9.2. WCS 2.0.1 request simplified by using only native grid coordinates	52
9.3. WCS 2.0.1 additional metadata for displaying meaningful native grid coordinates.	52
9.4. Detecting lat/lon of axes in WCS	53
9.5. Non clear treatment of XY and lat/lon axes	53
10. Future Recommendations	56
10.1. GRIB Encoding	56
10.2. Coverage Collections Extension	56
10.3. MetOcean Profile	56
10.4. Others	56
Appendix A: Revision History	57
Appendix B: Bibliography	59

Publication Date: 2017-04-28

Approval Date: 2016-12-07

Posted Date: 2016-11-11

Reference number of this document: OGC 16-033r1

Reference URL for this document: <http://www.opengis.net/doc/PER/t12-NA004>

Category: Public Engineering Report

Editor: Ranjay Shrestha, Liping Di, Eugene G. Yu

Title: Testbed-12 WCS Profile Update Engineering Report

OGC Engineering Report

COPYRIGHT

Copyright © 2017 Open Geospatial Consortium (OGC). To obtain additional rights of use, visit <http://www.opengeospatial.org/>

WARNING

This document is an Open Geospatial Consortium (OGC) Public Engineering Report created as a deliverable of an initiative from the OGC Innovation Program (formerly OGC Interoperability Program). It is not an OGC standard and not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by

destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Abstract

This engineering report capture the work to extend the existing Web Coverage Service (WCS) profiles, particularly the Earth Observation Application Profile (EO-WCS [OGC 10-140r1]) to support multi-dimensional subsetting of 3D space and 1D time. The updated EO-WCS (EO-WCS1.1 [OGC 10-140r2]) have removed the requirement for the 2D coverages so that it can explicitly allow coverages with more dimensions as long as they have geographic footprint. Furthermore it also clarified the use of *rangeType* when non-NCNAME characters are present in a band identifier. The example of *GetCapabilities*, *DescribeEOCoverageSet*, and *_GetCoverage* request in the updated EO-WCS1.1 is shown with use case on fire emission data in San Francisco.

Following the recommendation for EO-WCS to fully embrace the N-D, multi-dimensional, concept of Coverages as a function of time and other coordinates alongside the geospatial ones, the proposed recommendations/changes in the extension for WCS *DescribeCoverage*, EO-WCS *DescribeEOCoverageSet*, and WCS *GetCoverage* are discussed with use case example using National Centers for Environmental Prediction (NCEP) Global 0.25 deg wind data. Based on the mutual recommendation from the US National Aeronautics and Space Administration (NASA) and Baart et. al (2012), Network Common Data Form (NetCDF) was the output format due to presence of its libraries in multiple languages to lower the burden in changing on developers of WCS-compliant servers and clients.

For the extension of the WCS *DescribeCoverage*, it is recommended that CIS1.1 should be considered adopting a scheme for transmitting coordinates similar to the *_cis:rangeSet* where data are referred to as an attached Multipurpose Internet Mail Extensions (MIME) part. Time, as much as possible, be treated as just another coordinates dimension so that it could be access with the same tools used for other coordinate dimensions. To tackle the issue on order of coordinate dimensions, it is recommended to add implementation note to the EO-WCS specifications so that implementers are aware of the mismatches between dataset coordinate reference systems (CRSs) and actual axis order.

For the extension of EO_WCS *DescribeEOCoverageSet*, the issue on missing range of results API needed to be resolved by adding a request mechanism for requesting a range of matching results. It is also recommended that *DescribeEOCoverageSet* activity might be of more use to the client if the client need to supply only the subset conditions, and not a list of identifiers.

For the extension of WCS *GetCoverage*, it was discovered that for the *GetCoverage* operation for higher dimensioned datasets, existing WCS-2.0 request interface provided adequate syntax for subsetting higher dimensional data. Scaling (re-gridding) operation appears to be a natural fit for the EO-WCS subsetting, specifically SCALEEXTENT activity, however simpler explanation might be needed to fully understand its use as it appears other scaling and subsetting commands may be more than adequate for the desired outcomes. Additionally allowing *SlicePoint* subsetting is also recommended.

After performing the testing in the client side, there were few potential recommendations for improvements. More information on whether the coverage is 2D or 3D from the *GetCapabilities* request might be helpful to client so it can limit the number of *DescribeCoverage* requests to construct a list of available coverage on the server. Furthermore additional metadata information for displaying meaningful native grid coordinates is also recommended for clarification. Finally automatic detection of lat/lon axes along with clear treatment of XY and lat/lon axes ordering would be an improvement in the existing operations.

Business Value

This Engineering Report addresses the requirement to expand the existing EO-WCS profile to enable interoperability especially to the profile's treatment (or exclusion) of higher dimensions, non-geographic coordinates and aggregation.

What does this ER mean for the Working Group and OGC in general

This engineering report is important to the Web Coverage Service (WCS) Standard Working Group as it expands the existing EO-WCS profile to support 3D space and 1D time data and furthermore the profile will also include DAP4 and NetCDF responses.

How does this ER relate to the work of the Working Group

The work highlighted in the ER related to the WCS profile update will be related to the future work of the WCS.SWG in two ways:

1. Updating the existing WCS profile to better aligned with the specific needs of the EO data user community
2. Building on the EO-WCS application profile to capture the WCS access to EO data as a function of 3D space and 1D time along with the operations for subsetting the range of the coverage. As the WCS core already supports the N-

D spatio-temporal data, this ER specifically address the dimensionality restriction on EO-WCS.

Keywords

ogcdocs, testbed-12, WCS, EO-WCS, NetCDF, OPeNDAP, Multi-dimension

Proposed OGC Working Group for Review and Approval

This Engineering report will be submitted to the Web Coverage Service (WCS) Standard Working Group (WCS.SWG) for review.

Chapter 1. Introduction

1.1. Scope

This document reports the suggested expansion in the existing EO-WCS profile to support multi-dimensional subsetting of 3D space and 1D time data. The extended profile encoding will be in NetCDF responses. The presence of NetCDF libraries in multiple language provides better interoperability on developers of WCS-compliant servers and clients. This profile embraces the concept of Coverage with multi-dimensionality as function of time and other spatial and geospatial dimension.

1.2. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Table 1. Contacts

Name	Organization
Ranjay Shrestha	George Mason University, CSISS
Liping Di	George Mason University, CSISS
Eugene G. Yu	George Mason University, CSISS
Stephan Meissl	EOX IT Services GmbH
Nathan Potter	OPeNDAP
James Gallagher	OPeNDAP
Idan Shatz	PYXIS
Perry Peterson	PYXIS
Peter Baumann	Jacobs University

1.3. Future Work

Detail discussion of the future work is presented in the "Future Recommendations" section. The summary of the recommendations are: WCS General Regularly-distributed Information in Binary form (GRIB) Encoding, WCS Coverage Collection Extension, MetOcean Profile Support, CITE testing, and rangeType descriptions.

1.4. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any

relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Chapter 2. References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

- OGC 06-121r9, OGC® Web Services Common Standard
- OGC 09-110r4, OGC® WCS 2.0 Interface Standard- Core: Corrigendum
- OGC 09-146r2, OGC® Coverages Implementation Schema (CIS) (renamed from OGC® GML Application Schema - Coverages)
- OGC 10-140r1, OGC® Web Coverage Service 2.0 Interface Standard - Earth Observation Application Profile
- OGC 10-140r2, OGC® Web Coverage Service 2.0 Interface Standard - Earth Observation Application Profile (proposed version 1.1)

Chapter 3. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9] as well as those listed in the [reference documents above](#), particular EO-WCS [OGC 10-140r1] shall apply. In addition, the following terms and definitions apply.

3.1. DAP

DAP4 (The Data Access Protocol) is a NASA Community standard computer protocol which is designed to rely on stable HTTP and MIME standards to provide data access to facilitated gridded, relational and time series data along with user define data types [5].

3.2. NetCDF

NetCDF (Network Common Data Form) is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data [3].

3.3. OPeNDAP

OPeNDAP (Open-source Project for a Network Data Access Protocol) is a non-profit organization as well as commonly-used protocol name [4].

Chapter 4. Conventions

4.1. Abbreviated terms

- WCS OGC Web Coverage Service
- EO-WCS OGC® Web Coverage Service 2.0 Interface Standard - Earth Observation Application Profile
- DAP Data Access Protocol

Chapter 5. Overview

This engineering report describes extensions of the EO-WCS profile to address the data multi-dimensionality requirements from the EO data user community by utilizing DAP4 as a Coverage encoding as well as the WCS interface protocol and implementation. More specifically, to define and test the EO-WCS access with 3 dimensional space (spatial and geospatial data) and time along with operations for sub-setting the range of the coverages. Due to the presence of netCDF libraries in multiple languages, netCDF will be the output format.

Chapter 6. Status Quo & Annotated Requirements Statement

6.1. Status Quo

The subsetting request in the existing specification on the service core of the WCS [OGC 09-110r4] is carried out through the *DimensionSubset* request of the WCS *GetCoverage* interface. The subsetting is evaluated against the *gml:Envelope* element in the *boundedBy* element of the coverage address. The two approaches on subsetting are the *trim* and *slice* operations. In the *trim* operation, the pair of *trimLow* and *trimHigh* values define the axis coordinate's bound of cutout along the dimension. Similarly, in the *slice* operation the *slicePoint* value defines the slicing point along dimension. Fig 1 below [OGC 09-110r4] shows the UML diagram of the subset request of the WCS *GetCoverage* operation.

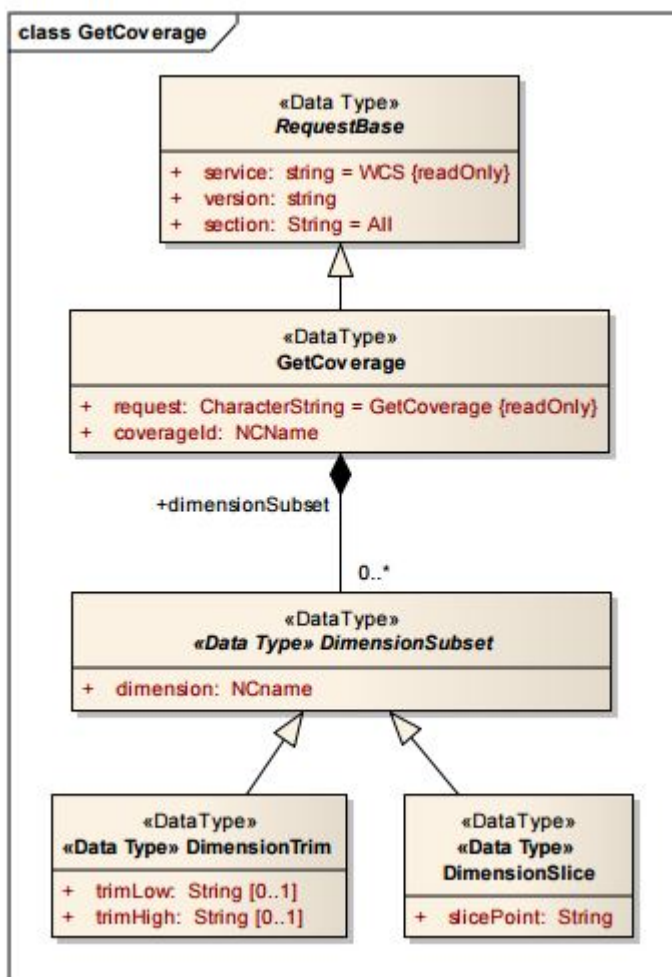


Figure 1: UML-GetCoverage-Subsetting (Source: [OGC 09-110r4])

6.2. Annotated Requirements Considered by the OPeNDAP and EOxServer Teams

In this section we examine each of the seven original requirements of Annex B of the Testbed 12 RFQ. For each we provide a brief description of the work done and outcome/result of that work. In

some cases we have started work outside of the Testbed 12 process, such as proposing that WCS 2.0 support DAP4 as a streaming data response or the EO-WCS 1.1 revision. In other cases it seems that requirements were too vague (see #4) or duplicates (see #3 and #7).

Requirement 1: *WCS-EO profile update. Update or create a new profile that is better aligned with the specific needs of the EO data user community following recommendations of Baart et al. and the NASA study cited above.*

The EOxServer team tackled this requirement by proposing several changes to the existing EO-WCS 1.0 as 1.1 revision. The specific changes related to this ER are introduced in the section [EO-WCS Update](#) below. The proposed EO-WCS 1.1 includes further changes and additions like the new GetEOCoverageSet operation as well but since those are not originating from the work described herein they are not detailed further.

The OPeNDAP team contributed considerably to the discussion and evaluation of the proposed changes.

Requirement 2: *EO Data encoding for WCS. Specify and test access to EO Data using WCS in order to identify a recommended mandatory encoding for WCS access to EO data. The form of such standardization remains to be defined, but it might include DAP4 as a Coverage encoding as well as a WCS interface protocol and implementation.*

There are two basic issues here: netCDF is already a standard Response Encoding for WCS and also well understood within the OGC community. It's also clear that the netCDF data model can store EO data and supports an open-ended metadata model (i.e., any type-name-value information can be used as metadata, although specific domains often have to develop or adopt standards to achieve semantic interoperability). Thus, as was suggested by OPeNDAP in a NASA-sponsored study, netCDF should be a *required* response format for EO-WCS. However, netCDF files cannot, in general, be streamed as responses back to clients, while both DAP2 and DAP4 responses have been designed specifically for this purpose. OPeNDAP, with support from NASA, has started the process of establishing DAP4 as a standard Response Encoding for WCS 2.0.

The current EO-WCS requires each implementation to support at least one of the GeoTIFF, netCDF, or JPEG2000 format encodings (requirement 71 or 188 in version 1.1). It is suggested that this requirement is revisited once DAP4 standardization for WCS 2.0 is adopted.

Requirement 3: *N-D Arrays. Building on the EO-WCS Application Profile, specify and test WCS access to EO data as a function of 3D space and 1D time along with operations for subsetting the range of the coverage, e.g., query data using an index-based approach along arbitrary domain and range dimensions. Consider OGC Web Coverage Processing Service (WCPS) Language and DAP4 in this task.*

The OPeNDAP team looked at WCPS and concluded that while it is very powerful, it is not needed to accomplish this requirement. The WCS and EO-WCS request operations are sufficient to subset 3D+Time data. Similarly, the DAP4 request syntax could be used for this, but is not needed. Both teams felt that WCS/EO-WCS's interface should remain as stable as possible and thus introducing unnecessary changes should be avoided.

A separate task we undertook was an analysis of the ease of implementing WCPS using the server processing capabilities of DAP4. It is clearly possible to implement WCPS using the DAP4 server function system, but to do so is a major undertaking. WCPS is a very complex language and a full

implementation would require a parser, software to implement the language's special forms and operators as well as the various mathematical operations WCPS includes. One aspect of WCPS that differs from DAP4 and its server functions is that DAP4 considers a given *dataset* (synonymous with *coverage* in WCS) to be the *environment* in which the server functions are evaluated. WCPS considers the collection of *coverages* known to the server to be the evaluation environment for its expressions. Thus the DAP4 function evaluator would have to be extended in a nontrivial fashion to support all legal WCPS expressions/programs.

Requirement 4: *Aim for alignment with DAP4 specifications.* Requirements are as follows:

- The DescribeCoverage request of WCS should be able to support all metadata describable information in DAP4.
- The GetCoverage request also should be able to support all operations specified in DAP4.

Requirement 5: *Developing of multilingual, open-source libraries as tools for promulgating use of its standards*

It is recommended to develop a set of multilingual, open-source libraries and toolkits to demonstrate the mechanism to implement and enable the specification.

Requirement 6: *Consideration of formal adoption of DAP v4.0 with modifications/extensions as necessary*

We are working with the Coverages Working Group and WCS Standards Working Group to establish the DAP4 data response as a streaming response for WCS 2.0 and/or EO-WCS (see also requirement #2 above).

Requirement 7: *Specify and test WCS access to EO data as a function of 3D space and 1D time along with operations for subsetting the range of the coverage, e.g., query data using an index-based approach along arbitrary domain and range dimensions. Consider OGC Web Coverage Processing Service (WCPS) Language and DAP4 in this task.*

This requirement is identical to the requirement #3 with the exception that the 'S' in 'specify' is capitalized in one and not the other. We are not sure what the original authors' intent was and have ignored requirement #7.

6.3. Targeted Solutions

Build on the EO-WCS Application Profile, specify and test WCS access to EO data as a function of 3D space and 1D time along with operations for subsetting the range of the coverage. The implementation work and use case is provided in [chapter 8](#) below.

Extend the profile of the EO-WCS particularly to add support for the third dimensions which include one or all of, height, depth, pressure, temperature. Height and Pressure are mainly associated with the atmospheric data where the positive height value and decreasing pressure indicate 'up'. Similarly the Depth and Temperature are utilized in Ocean data where positive value of depth as well as generally decreasing temperature are indicated 'down'.

NetCDF should be among the mandatory formats due to presence of its libraries in multiple

languages which lowers the burden in changing on developer of WCS-compliant servers and clients. The detail work on EO-WCS profile extension to add 3D support and mandatory output format specifications are carried out in [chapter 7](#) below.

Chapter 7. WCS support for OPeNDAP

We begin with a comparison of WCS and OPeNDAP followed by a description of the work that was done in the LSA thread to serve 3D+Time data. Our work was done using an existing OPeNDAP server that is widely deployed to serve as a *compute engine* for a WCS 2.0 service that conforms to the EO-WCS profile.

7.1. WCS/OPeNDAP Comparison

The WCS and OPeNDAP parallels are remarkable and close, especially at the mathematical level. Most or all aspects of WCS have been considered in developing the latest OPeNDAP protocol (DAP4), although some parts of Core WCS 2.0 (such as polygonally-meshed domains) are handled only via DAP4 extensions. Even WCS extensions (netCDF encoding and Coverage processing, e.g.) have parallels in DAP4 or its extensions. These parallels are reinforced by the successful addition of WCS support to OPeNDAP's own Hyrax server system under DAP2. The OPeNDAP and EO-WCS services both offer data response encodings in NetCDF, and when employed with suitable geospatial conventions, such as CF-1.6 or higher, the DAP2 and DAP4 data models and the associated service protocol have very similar capabilities and the OGC Coverage Model and the associated service standards.

The primary differences between the latest EO-WCS and OPeNDAP are as follows:

- Where EO-WCS is defined and understood within the larger OGC framework of standards for all things geospatial, OPeNDAP is largely self-contained and has few external references beyond netCDF, HDF, XML and other standards widely used in commercial software and hardware.
- Where EO-WCS emphasizes its geospatial orientation, OPeNDAP is domain neutral, providing subset operations based on the syntactic metadata (i.e., the 'computer science data types) except by extensions to the protocol. To some degree this is just a matter of terminology, such as OPeNDAP's use of generic "map" vectors or arrays to define sampling geometries in arbitrary (rather than geospatial) domain coordinates.
- In contrast to EO-WCS the OPeNDAP DAP2 and DAP4 offer straight forward representations and access to N-dimensional data. Through extensions, the integration and subsetting of "unstructured grid" (a.k.a. polygonally-meshed) domains is also available.
- Aggregation of datasets, a feature supported by multiple DAP server implementations, allows these servers to reduce their catalog complexity by exposing the aggregated datasets in lieu of their child constituents. By adding the composite object *wcseo:DatasetSeries* and the request interface *wcseo:DescribeEOCoverageSet* EO-WCS adds similar catalog content reduction to catalog requests. However this notion is not carried into the data access activity. The DAP2/4 services deliver the full data access capabilities to every exposed dataset. Every aggregated dataset appears as a single dataset endpoint and the associated data access responses are each returned as a single data object constructed from the aggregation's constituent datasets. With the use of a bundled extension the Hyrax DAP server can also respond to a syntactically different aggregation request which will return the result as a collection of NetCDF files packaged as a compressed tar file. In contrast, the EO-WCS *wcseo:DescribeEOCoverageSet* command can be employed as an inventory search over a known set of collections, including *wcseo:DatasetSeries* dataset collections. When the search has multiple results the service

returns a collection of *wcs:CoverageDescriptions*. It is left to the client to make individual *wcs:GetCoverage* requests to achieve data access for each coverage.

- Where EO-WCS is laid out by making formal and extensive use of GML, including definitions employed for compliance verification, OPeNDAP relies on plain-English specifications aimed for ease of understanding by software developers (working in multiple computer languages). Further emphasizing this point, OPeNDAP accompanies its specifications with software libraries, geared toward developers of clients and servers, that realize the protocol and the underlying data model via well-documented APIs.
- EO-WCS service offers CRS-based interpolation/re-projection, without extensions, DAP4 does not.

In summary, the potential for OPeNDAP-WCS congruence appears to be very high, and there are some simple and very useful integrations of the two protocols such as combining an EO-WCS service with an aggregating DAP server. This could provide the benefit of catalog reduction by exposing an aggregation of DAP data sets (each a coverage on it's own) as a single coverage and at the same time reduce the number of requests that a client has to make to achieve data access to the various components of the aggregated dataset. This scheme can coordinate thousands of individual datasets into a single coverage in EO-WCS.

7.1.1. Comparative examples of the OPeNDAP and EO-WCS Coverage representations

The following example is from the US National Weather Service, National Centers of Environmental Prediction (NCEP) forecast data of hourly wind over an isobaric surface.

OPeNDAP

The DAP protocol currently supports two formal data responses, DAP2 and DAP4. DAP4, while more verbose, is the richer response in that it transmits the syntactic (structural) metadata and all of the semantic metadata (that which is not structural) with each data response. In DAP2 a separate request must be made to retrieve the semantic metadata. The guarantee in both DAP data transmissions is that the syntactic metadata will be absolutely correct.

Additionally several DAP servers support retrieving alternate encodings of the DAP data response, most notably as NetCDF which, much like DAP4, delivers both the syntactic and semantic metadata in a single data object.

DAP4 Representation

Here is a DAP4 combined metadata response (DMR) for the NCEP model output dataset `Global_0p25deg_best_hs002.nc`

Request **URL:** http://testbed-12.opendap.org:8080/opendap/testbed-12/ncep/Global_0p25deg_best_hs002.nc.dmr

DAP4 DMR Response for NCEP Dataset

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Dataset xmlns="http://xml.opendap.org/ns/DAP/4.0#"
```

```

xml:base="http://localhost:8080/pendap/testbed-12/ncep/Global_0p25deg_best_hs002.nc"
dapVersion="4.0" dmrVersion="1.0" name="Global_0p25deg_best_hs002.nc">
  <Dimension name="time" size="112"/>
  <Dimension name="isobaric" size="31"/>
  <Dimension name="lat" size="361"/>
  <Dimension name="lon" size="720"/>
  <Float64 name="time">
    <Dim name="/time"/>
    <Attribute name="units" type="String">
      <Value>Hour since 2016-06-17T00:00:00Z</Value>
    </Attribute>
    <Attribute name="standard_name" type="String">
      <Value>time</Value>
    </Attribute>
    <Attribute name="long_name" type="String">
      <Value>GRIB forecast or observation time</Value>
    </Attribute>
    <Attribute name="calendar" type="String">
      <Value>proleptic_gregorian</Value>
    </Attribute>
    <Attribute name="_CoordinateAxisType" type="String">
      <Value>Time</Value>
    </Attribute>
  </Float64>
  <Float32 name="isobaric">
    <Dim name="/isobaric"/>
    <Attribute name="units" type="String">
      <Value>Pa</Value>
    </Attribute>
    <Attribute name="long_name" type="String">
      <Value>Isobaric surface</Value>
    </Attribute>
    <Attribute name="positive" type="String">
      <Value>down</Value>
    </Attribute>
    <Attribute name="Grib_level_type" type="Int32">
      <Value>100</Value>
    </Attribute>
    <Attribute name="_CoordinateAxisType" type="String">
      <Value>Pressure</Value>
    </Attribute>
    <Attribute name="_CoordinateZisPositive" type="String">
      <Value>down</Value>
    </Attribute>
  </Float32>
  <Float32 name="lat">
    <Dim name="/lat"/>
    <Attribute name="units" type="String">
      <Value>degrees_north</Value>
    </Attribute>
    <Attribute name="_CoordinateAxisType" type="String">

```

```

    <Value>Lat</Value>
  </Attribute>
  <Attribute name="standard_name" type="String">
    <Value>latitude</Value>
  </Attribute>
</Float32>
<Float32 name="lon">
  <Dim name="/lon"/>
  <Attribute name="units" type="String">
    <Value>degrees_east</Value>
  </Attribute>
  <Attribute name="_CoordinateAxisType" type="String">
    <Value>Lon</Value>
  </Attribute>
  <Attribute name="standard_name" type="String">
    <Value>longitude</Value>
  </Attribute>
</Float32>
<Float32 name="u-component_of_wind_isobaric">
  <Dim name="/time"/>
  <Dim name="/isobaric"/>
  <Dim name="/lat"/>
  <Dim name="/lon"/>
  <Attribute name="long_name" type="String">
    <Value>u-component of wind @ Isobaric surface</Value>
  </Attribute>
  <Attribute name="units" type="String">
    <Value>m/s</Value>
  </Attribute>
  <Attribute name="abbreviation" type="String">
    <Value>UGRD</Value>
  </Attribute>
  <Attribute name="missing_value" type="Float32">
    <Value>NaN</Value>
  </Attribute>
  <Attribute name="grid_mapping" type="String">
    <Value>LatLon_Projection</Value>
  </Attribute>
  <Attribute name="coordinates" type="String">
    <Value>reftime time isobaric lat lon lat lon</Value>
  </Attribute>
  <Attribute name="Grib_Variable_Id" type="String">
    <Value>VAR_0-2-2_L100</Value>
  </Attribute>
  <Attribute name="Grib2_Parameter" type="Int32">
    <Value>0</Value>
    <Value>2</Value>
    <Value>2</Value>
  </Attribute>
  <Attribute name="Grib2_Parameter_Category" type="String">
    <Value>Momentum</Value>

```

```

</Attribute>
<Attribute name="Grib2_Parameter_Name" type="String">
  <Value>u-component of wind</Value>
</Attribute>
<Attribute name="Grib2_Level_Type" type="String">
  <Value>Isobaric surface</Value>
</Attribute>
<Attribute name="Grib2_Generating_Process_Type" type="String">
  <Value>Forecast</Value>
</Attribute>
<Map name="time"/>
<Map name="isobaric"/>
<Map name="lat"/>
<Map name="lon"/>
</Float32>
<Float64 name="reftime">
  <Dim name="/time"/>
  <Attribute name="standard_name" type="String">
    <Value>forecast_reference_time</Value>
  </Attribute>
  <Attribute name="long_name" type="String">
    <Value>GRIB reference time</Value>
  </Attribute>
  <Attribute name="calendar" type="String">
    <Value>proleptic_gregorian</Value>
  </Attribute>
  <Attribute name="units" type="String">
    <Value>Hour since 2016-06-17T00:00:00Z</Value>
  </Attribute>
  <Attribute name="_CoordinateAxisType" type="String">
    <Value>RunTime</Value>
  </Attribute>
  <Map name="time"/>
</Float64>
<Float64 name="time">
  <Dim name="/time"/>
  <Attribute name="units" type="String">
    <Value>Hour since 2016-06-17T00:00:00Z</Value>
  </Attribute>
  <Attribute name="standard_name" type="String">
    <Value>time</Value>
  </Attribute>
  <Attribute name="long_name" type="String">
    <Value>GRIB forecast or observation time</Value>
  </Attribute>
  <Attribute name="calendar" type="String">
    <Value>proleptic_gregorian</Value>
  </Attribute>
  <Attribute name="_CoordinateAxisType" type="String">
    <Value>Time</Value>
  </Attribute>

```



```

</Float64>
<Float32 name="isobaric">
  <Dim name="/isobaric"/>
  <Attribute name="units" type="String">
    <Value>Pa</Value>
  </Attribute>
  <Attribute name="long_name" type="String">
    <Value>Isobaric surface</Value>
  </Attribute>
  <Attribute name="positive" type="String">
    <Value>down</Value>
  </Attribute>
  <Attribute name="Grib_level_type" type="Int32">
    <Value>100</Value>
  </Attribute>
  <Attribute name="_CoordinateAxisType" type="String">
    <Value>Pressure</Value>
  </Attribute>
  <Attribute name="_CoordinateZisPositive" type="String">
    <Value>down</Value>
  </Attribute>
</Float32>
<Float32 name="lat">
  <Dim name="/lat"/>
  <Attribute name="units" type="String">
    <Value>degrees_north</Value>
  </Attribute>
  <Attribute name="_CoordinateAxisType" type="String">
    <Value>Lat</Value>
  </Attribute>
  <Attribute name="standard_name" type="String">
    <Value>latitude</Value>
  </Attribute>
</Float32>
<Float32 name="lon">
  <Dim name="/lon"/>
  <Attribute name="units" type="String">
    <Value>degrees_east</Value>
  </Attribute>
  <Attribute name="_CoordinateAxisType" type="String">
    <Value>Lon</Value>
  </Attribute>
  <Attribute name="standard_name" type="String">
    <Value>longitude</Value>
  </Attribute>
</Float32>
<Float32 name="v-component_of_wind_isobaric">
  <Dim name="/time"/>
  <Dim name="/isobaric"/>
  <Dim name="/lat"/>
  <Dim name="/lon"/>

```

```

<Attribute name="long_name" type="String">
  <Value>v-component of wind @ Isobaric surface</Value>
</Attribute>
<Attribute name="units" type="String">
  <Value>m/s</Value>
</Attribute>
<Attribute name="abbreviation" type="String">
  <Value>VGRD</Value>
</Attribute>
<Attribute name="missing_value" type="Float32">
  <Value>NaN</Value>
</Attribute>
<Attribute name="grid_mapping" type="String">
  <Value>LatLon_Projection</Value>
</Attribute>
<Attribute name="coordinates" type="String">
  <Value>reftime time isobaric lat lon lat lon</Value>
</Attribute>
<Attribute name="Grib_Variable_Id" type="String">
  <Value>VAR_0-2-3_L100</Value>
</Attribute>
<Attribute name="Grib2_Parameter" type="Int32">
  <Value>0</Value>
  <Value>2</Value>
  <Value>3</Value>
</Attribute>
<Attribute name="Grib2_Parameter_Category" type="String">
  <Value>Momentum</Value>
</Attribute>
<Attribute name="Grib2_Parameter_Name" type="String">
  <Value>v-component of wind</Value>
</Attribute>
<Attribute name="Grib2_Level_Type" type="String">
  <Value>Isobaric surface</Value>
</Attribute>
<Attribute name="Grib2_Generating_Process_Type" type="String">
  <Value>Forecast</Value>
</Attribute>
<Map name="time"/>
<Map name="isobaric"/>
<Map name="lat"/>
<Map name="lon"/>
</Float32>
<Attribute name="NC_GLOBAL" type="Container">
  <Attribute name="Originating_or_generating_Center" type="String">
    <Value>US National Weather Service, National Centres for Environmental
Prediction (NCEP)</Value>
  </Attribute>
  <Attribute name="Originating_or_generating_Subcenter" type="String">
    <Value>0</Value>
  </Attribute>

```

```

<Attribute name="GRIB_table_version" type="String">
  <Value>2,1</Value>
</Attribute>
<Attribute name="Type_of_generating_process" type="String">
  <Value>Forecast</Value>
</Attribute>
<Attribute
name="Analysis_or_forecast_generating_process_identifier_defined_by_originating_centre
" type="String">
  <Value>Analysis from GFS (Global Forecast System)</Value>
</Attribute>
<Attribute name="Conventions" type="String">
  <Value>CF-1.6</Value>
</Attribute>
<Attribute name="history" type="String">
  <Value>Read using CDM IOSP GribCollection v3</Value>
</Attribute>
<Attribute name="featureType" type="String">
  <Value>GRID</Value>
</Attribute>
<Attribute name="History" type="String">
  <Value>Translated to CF-1.0 Conventions by Netcdf-Java CDM
(CFGridWriter2)\012Original Dataset =
/data/ldm/pub/native/grid/NCEP/GFS/Global_0p25deg/GFS-Global_0p25deg.ncx3; Translation
Date = 2016-06-20T21:22:16.485Z</Value>
</Attribute>
<Attribute name="geospatial_lat_min" type="Float64">
  <Value>-90.00000000000000</Value>
</Attribute>
<Attribute name="geospatial_lat_max" type="Float64">
  <Value>90.00000000000000</Value>
</Attribute>
<Attribute name="geospatial_lon_min" type="Float64">
  <Value>0.00000000000000</Value>
</Attribute>
<Attribute name="geospatial_lon_max" type="Float64">
  <Value>359.5000000000000</Value>
</Attribute>
</Attribute>
</Dataset>

```

DAP2 Syntactic Metadata

Here is a DAP2 syntactic metadata response (DDS) for the NCEP model output dataset Global_0p25deg_best_hs002.nc This is transmitted with every DAP2 data (.dods) response.

Request **URL:** http://testbed-12.opendap.org:8080/opendap/testbed-12/ncep/Global_0p25deg_best_hs002.nc.dds

```

Dataset {
  Grid {
    Array:
      Float32 u-component_of_wind_isobaric[time = 112][isobaric = 31][lat = 361][lon
= 720];
    Maps:
      Float64 time[time = 112];
      Float32 isobaric[isobaric = 31];
      Float32 lat[lat = 361];
      Float32 lon[lon = 720];
  } u-component_of_wind_isobaric;
  Grid {
    Array:
      Float64 reftime[time = 112];
    Maps:
      Float64 time[time = 112];
  } reftime;
  Float64 time[time = 112];
  Float32 isobaric[isobaric = 31];
  Float32 lat[lat = 361];
  Float32 lon[lon = 720];
  Grid {
    Array:
      Float32 v-component_of_wind_isobaric[time = 112][isobaric = 31][lat = 361][lon
= 720];
    Maps:
      Float64 time[time = 112];
      Float32 isobaric[isobaric = 31];
      Float32 lat[lat = 361];
      Float32 lon[lon = 720];
  } v-component_of_wind_isobaric;
} Global_0p25deg_best_hs002.nc;

```

Equivalent EO-WCS CoverageDescription

Request http://testbed-12.opendap.org:8080/WCS-2.0?service=WCS&version=2.0.1&request=DescribeCoverage&coverageId=eo_ncep_model_example

EO-WCS CoverageDescription (flawed) for NCEP Dataset

```

<?xml version="1.0" encoding="UTF-8"?>
<wcs:CoverageDescriptions xmlns:wcs="http://www.opengis.net/wcs/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:gmlcov="http://www.opengis.net/gmlcov/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/2.0
http://schemas.opengis.net/wcs/2.0/wcsAll.xsd http://www.opengis.net/gml/3.2
http://schemas.opengis.net/gml/3.2.1/gml.xsd http://www.opengis.net/swe/2.0

```

```

http://schemas.opengis.net/sweCommon/2.0/swe.xsd http://www.opengis.net/gmlcov/1.0
http://schemas.opengis.net/gmlcov/1.0/gmlcovAll.xsd ">
  <wcs:CoverageDescription xmlns="http://www.opengis.net/gml/3.2"
xmlns:ows="http://www.opengis.net/ows/2.0"
xmlns:wcseo="http://www.opengis.net/wcs/wcseo/1.0"
xmlns:eop="http://www.opengis.net/eop/2.0" xmlns:om="http://www.opengis.net/om/2.0"
xmlns:xlink="http://www.w3.org/1999/xlink" gml:id="eo_ncep_model_example"
xsi:schemaLocation="http://www.opengis.net/wcs/wcseo/1.0
http://schemas.opengis.net/wcs/wcseo/1.0/wcsEOAll.xsd">
  <gml:boundedBy>
    <gml:Envelope axisLabels="latitude longitude" srsDimension="2"
srsName="http://www.opengis.net/def/crs/EPSG/0/4326" uomLabels="deg deg">
      <gml:lowerCorner>-90.0 0.0</gml:lowerCorner>
      <gml:upperCorner>90.0 359.5</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <wcs:CoverageId>eo_ncep_model_example</wcs:CoverageId>
  <gmlcov:metadata>
    <gmlcov:Extension>
      <wcseo:EOMetadata>
        <eop:EarthObservation gml:id="eop_eo_ncep_model_example">
          <om:phenomenonTime>
            <gml:TimePeriod gml:id="tp_eo_ncep_model_example">
              <gml:beginPosition>2016-06-17T00:00:00.000Z</gml:beginPosition>
              <!-- 108 days -->
              <gml:endPosition>2016-07-02T00:00:00.000Z</gml:endPosition>
            </gml:TimePeriod>
          </om:phenomenonTime>
          <om:resultTime>
            <gml:TimeInstant gml:id="archiving_date_eo_ncep_model_example">
              <gml:timePosition>2016-07-03T11:02:47.999</gml:timePosition>
            </gml:TimeInstant>
          </om:resultTime>
          <om:procedure>
            <eop:EarthObservationEquipment gml:id="eo_ncep_model_example_eoe">
              <eop:platform>
                <!-- 0..1 -->
                <shortName>NCEP</shortName>
              </eop:platform>
              <!-- <eop:instrument /> 0..1 -->
              <!-- <eop:sensor /> 0..1 -->
              <!-- <eop:acquisitionParameters /> 0..1 -->
            </eop:EarthObservationEquipment>
          </om:procedure>
          <om:observedProperty xlink:href="http://cfconventions.org/Data/cf-
standard-names/35/src/cf-standard-name-table.xml#wind" />
          <!-- 1..n -->
          <om:featureOfInterest>
            <eop:Footprint gml:id="footprint_someEOCoverage1">
              <eop:multiExtentOf>
                <gml:MultiSurface gml:id="multisurface_someEOCoverage1"

```

```

srsName="EPSG:4326">
  <gml:surfaceMember>
    <gml:Polygon gml:id="polygon_someEOCoverage1">
      <gml:exterior>
        <gml:LinearRing>
          <gml:posList>-90.0 0.0 90.0 0.0 90.0 360.0 -90.0 360.0
-90.0 0.0</gml:posList>
        </gml:LinearRing>
      </gml:exterior>
    </gml:Polygon>
  </gml:surfaceMember>
</gml:MultiSurface>
</eop:multiExtentOf>
<eop:centerOf>
  <gml:Point gml:id="someEOCoverage1_p" srsName="EPSG:4326">
    <gml:pos>43.190833 2.374167</gml:pos>
  </gml:Point>
</eop:centerOf>
</eop:Footprint>
</om:featureOfInterest>
  <!-- om:result - This is required to be
om:result/eop:EarthObservationResult but there are no required children of
eop:EarthObservationResult. I added one child (eop:parameter) in order to not to
nilify the eop:EarthObservationResult -->
  <om:result>
    <eop:EarthObservationResult>
      <!-- <eop:browse /> 0..1 -->
      <!-- <eop:product /> 0..1 -->
      <!-- <eop:mask /> 0..1 -->
      <eop:parameter> <!-- 0..1 -->
        <eop:ParameterInformation>
          <eop:unitOfMeasure>m/s</eop:unitOfMeasure>
        </eop:ParameterInformation>
      </eop:parameter>
      <!-- <eop:coverage /> 0..1 -->
    </eop:EarthObservationResult>
  </om:result>
  <eop:metaDataProperty>
    <eop:EarthObservationMetaData>
      <eop:identifier>eop_eo_ncep_model_example</eop:identifier>
      <!-- eop:acquisitionType Used to distinguish at a high level the
appropriateness of the acquisition for "general" use, whether the product is a nominal
acquisition, special calibration product or other. AllowedValues: NOMINAL |
CALIBRATION | OTHER -->
      <eop:acquisitionType>NOMINAL</eop:acquisitionType>
      <!-- eop:status - Refers to product status and the allowed values have
little relevance for model output. AllowedValues: ARCHIVED | ACQUIRED | CANCELLED |
FAILED | PLANNED | POTENTIAL | REJECTED | QUALITYDEGRADED -->
      <eop:status>ARCHIVED</eop:status>
    </eop:EarthObservationMetaData>
  </eop:metaDataProperty>

```

```

    </eop:EarthObservation>
  </wcseo:EOMetadata>
</gmlcov:Extension>
</gmlcov:metadata>
<gml:domainSet>
  <gml:RectifiedGrid dimension="3" gml:id="eo_ncep_model_example_grid">
    <gml:limits>
      <gml:GridEnvelope>
        <gml:low>0 0 0</gml:low>
        <gml:high>30 360 719</gml:high>
      </gml:GridEnvelope>
    </gml:limits>
    <gml:axisLabels>isobaric latitude longitude</gml:axisLabels>
    <gml:origin>
      <gml:Point gml:id="grid_origin_eo_ncep_model_example">
srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
        <gml:pos>100.0 90.0 0.0</gml:pos>
      </gml:Point>
    </gml:origin>
    <!-- isobaric is not regular!! Values: 100, 200, 300, 500, 700, 1000, 2000,
3000, 5000, 7000, 10000, 15000, 20000, 25000, 30000, 35000, 40000, 45000, 50000,
55000, 60000, 65000, 70000, 75000, 80000, 85000, 90000, 92500, 95000, 97500, 100000
-->
      <gml:offsetVector srsName="http://www.opengis.net/def/crs/IRREGULAR">100 0.0
0.0</gml:offsetVector>
      <gml:offsetVector srsName="http://www.opengis.net/def/crs/EPSG/0/4326">0.0 0.5
0.0</gml:offsetVector>
      <gml:offsetVector srsName="http://www.opengis.net/def/crs/EPSG/0/4326">0.0 0.0
-0.5</gml:offsetVector>
    </gml:RectifiedGrid>
  </gml:domainSet>
<gmlcov:rangeType>
  <swe:DataRecord>
    <swe:field name="u_component_of_wind_isobaric">
      <swe:Quantity
definition="urn:ncep:def:independantVariable:u_component_of_wind_isobaric">
        <swe:description>u-component of wind @ Isobaric surface</swe:description>
        <swe:uom code="m/s" />
        <swe:constraint>
          <swe:AllowedValues>
            <swe:interval>-60.0 161.0</swe:interval>
          </swe:AllowedValues>
        </swe:constraint>
      </swe:Quantity>
    </swe:field>
    <swe:field name="v_component_of_wind_isobaric">
      <swe:Quantity
definition="urn:ncep:def:independantVariable:vu_component_of_wind_isobaric">
        <swe:description>v-component of wind @ Isobaric surface</swe:description>
        <swe:uom code="m/s" />
        <swe:constraint>

```

```

        <swe:AllowedValues>
            <swe:interval>-71.0 79.0</swe:interval>
        </swe:AllowedValues>
    </swe:constraint>
</swe:Quantity>
</swe:field>
</swe:DataRecord>
</gmlcov:rangeType>
<wcs:ServiceParameters>
    <wcs:CoverageSubtype>RectifiedDataset</wcs:CoverageSubtype>
    <wcs:CoverageSubtypeParent>
        <wcs:CoverageSubtype>RectifiedGridCoverage</wcs:CoverageSubtype>
    </wcs:CoverageSubtypeParent>
    <wcs:nativeFormat>application/octet-stream</wcs:nativeFormat>
</wcs:ServiceParameters>
</wcs:CoverageDescription>
</wcs:CoverageDescriptions>

```

7.2. Extension for WCS *DescribeCoverage*

7.2.1. Coordinate Dimensions Beyond Latitude and Longitude

In its native form the NCEP example dataset's measurement/model data values are held as a 4 dimensional array of IEEE floating point values:

```
Float32 u-component_of_wind_isobaric[time = 112][isobaric = 31][lat = 361][lon = 720];
```

The corresponding "Map" arrays which hold the relevant coordinates are *time*, *isobaric* (atmospheric pressure), *lat*(itude), and *lon*(gitude). One of the interesting wrinkles in the dataset is that while the *time*, *latitude* and *longitude* coordinate maps conform to the EO-WCS definition of a *RectifiedGrid* the *isobaric* dimension does not, as its values are irregularly spaced. The result of this is that the existing EO-WCS options for representing the dataset are inadequate.

In certain circumstances the *wcseo:DatasetSeries* could be utilized to deal with this n-dimensionality issue in EO-WCS. If, as in the NCEP dataset example, the data are 3D+time, the latitude and longitude coordinates are rectified, but the dataset's 3rd dimension is small and irregular then one might consider representing the coverage as an *wcseo:DatasetSeries* composed of a different coverage for each value of the third dimension. It may be possible to accomplish this without restructuring the underlying data. (This idea could be applied to nested *wcseo:DatasetSeries* to produce a representation of an n-dimensional dataset or nested tabular data.)

CIS-1.1

We looked at a draft of CIS-1.1 [OGC 09-146r3] and found that it offered a possible solution to the representational problems posed by the NCEP model output example.

In the following example an attempt is made to represent the NCEP dataset using CIS-1.1.

Possible CIS-1.1 Representation Of The NCEP Model Output Dataset


```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<generalGridCoverage xmlns="http://www.opengis.net/cis/1.1"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:cis="http://www.opengis.net/cis/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/cis/1.1 ../cisAll.xsd"
gml:id="cis_eo_ncep_model_example">
  <domainSet>
    <!-- EPSG 4979 is "geographic 3D" geodetic CRS which defines the
         third axis as "ellipsoidal height" the author is not
         aware of an EPSG that defines an axis based on atmospheric
         pressure, but he imagines there may be such a thing
    -->
    <generalGrid srsName="http://www.opengis.net/def/crs/EPSSG/0/4979"
axisLabels="Lat Lon isobaric">
      <cis:regularAxis axisLabel="Lat" lowerBound="-90.0" upperBound="90.0"
uomLabel="deg" resolution="0.5"/>
      <cis:regularAxis axisLabel="Lon" lowerBound="0.0" upperBound="360.0"
uomLabel="deg" resolution="0.5"/>
      <irregularAxis axisLabel="isobaric" uomLabel="Pa">
        <c>100</c><c>200</c><c>300</c><c>500</c><c>700</c><c>1000</c><c>2000</c>
          <c>3000</c><c>5000</c><c>7000</c><c>10000</c><c>15000</c><c>20000</c>
        <c>25000</c><c>30000</c><c>35000</c><c>40000</c><c>45000</c><c>50000</c>
        <c>55000</c><c>60000</c><c>65000</c><c>70000</c><c>75000</c><c>80000</c>
        <c>85000</c><c>90000</c><c>92500</c><c>95000</c><c>97500</c><c>100000</c>
      </irregularAxis>
      <gridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index4D"
axisLabels="Lat Lon isobaric">
        <indexAxis axisLabel="Lat" lowerBound="0" upperBound="360"/>
        <indexAxis axisLabel="Lon" lowerBound="0" upperBound="719"/>
        <indexAxis axisLabel="isobaric" lowerBound="0" upperBound="30"/>
      </gridLimits>
    </generalGrid>
  </domainSet>
  <rangeSet>
    <gml:File>
      <gml:rangeParameters xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="cid:eo_ncep_model_example"
xlink:role="http://www.opengis.net/spec/WCS_coverage-encoding_opendap/req/dap2"
xlink:arcrole="fileReference"/>
      <gml:fileReference>cid:eo_ncep_model_example</gml:fileReference>
      <gml:fileStructure/>
    </gml:File>
  </rangeSet>
  <rangeType>
    <swe:DataRecord xmlns:swe="http://www.opengis.net/swe/2.0">

```

```

    <swe:field name="u_component_of_wind_isobaric">
      <swe:Quantity
definition="urn:ncep:def:independantVariable:u_component_of_wind_isobaric">
        <swe:description>u-component of wind @ Isobaric
surface</swe:description>
        <swe:uom code="m/s"/>
        <swe:constraint>
          <swe:AllowedValues>
            <swe:interval>-60.0 161.0</swe:interval>
          </swe:AllowedValues>
        </swe:constraint>
      </swe:Quantity>
    </swe:field>
    <swe:field name="v_component_of_wind_isobaric">
      <swe:Quantity
definition="urn:ncep:def:independantVariable:vu_component_of_wind_isobaric">
        <swe:description>v-component of wind @ Isobaric
surface</swe:description>
        <swe:uom code="m/s"/>
        <swe:constraint>
          <swe:AllowedValues>
            <swe:interval>-71.0 79.0</swe:interval>
          </swe:AllowedValues>
        </swe:constraint>
      </swe:Quantity>
    </swe:field>
  </swe:DataRecord>
</rangeType>
</generalGridCoverage>

```

This CIS-1.1 representation seems well suited for the NCEP dataset.

Satellite swath data is more complex because the latitude and longitude coordinates are irregular and are typically stored as two dimensional arrays:

```

Float32 Latitude[Cell_Along_Swath = 203][Cell_Across_Swath = 135];
Float32 Longitude[Cell_Along_Swath = 203][Cell_Across_Swath = 135];

```

And these data, while manageable, would begin to strain the CIS-1.1 representation due the 27,405 coordinate point elements that would be required to populate a CIS-1.1 displacementAxisNest element (which the author believes matches the swath coordinate example):

Abbreviated Swath Coordinates in CIS-1.1 Example

```
<displacementAxisNest
  axisLabels="Lat Long"
  uomLabels="deg deg"
  xmlns="http://www.opengis.net/cis/1.1">
  <p><c>-90</c><c>0</c></p> <p><c>-87</c><c>1</c></p> <p><c>-82</c><c> 2</c></p>
  <p><c>-89</c><c>4</c></p> <p><c>-86</c><c>5</c></p> <p><c>-81</c><c> 6</c></p>
  <p><c>-88</c><c>8</c></p> <p><c>-85</c><c>9</c></p> <p><c>-80</c><c>10</c></p>
</displacementAxisNest>
```

But for significantly larger datasets with similarly structured coordinates it may be unworkable without custom parsing. Consider these coordinate arrays:

```
Float32 Latitude[Cell_Alone_Swath = 4320][Cell_Across_Swath = 8640];
Float32 Longitude[Cell_Alone_Swath = 4320][Cell_Across_Swath = 8640];
```

Encoding the 37,324,800 coordinate pairs into the CIS-1.1 XML representation is impractical, simply because the use of anything but a custom SAX parser translating the coordinate elements directly into array values in memory will likely cause significant document consumption problems on the client side. In practice, most client developers will already be utilizing an XML library with a DOM implementation. These implementations typically parse an XML document into in-memory representation of the XML objects that are then navigated through an API. The result of this is that ingesting very large XML documents can be costly, if not impossible.

Recommendation: The authors feel that CIS-1.1 should consider adopting a scheme for transmitting coordinates similar to the one used in the `cis:rangeSet` in which the data are referred to as an attached MIME part. It would be preferable if DAP2, DAP4, and/or NetCDF could be used to encode the coordinate arrays in the attachment.

Example `gml:File` reference to an additional MIME part

```
<rangeSet xmlns="http://www.opengis.net/cis/1.1">
  <gml:File>
    <gml:rangeParameters xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="cid:eo_ncep_model_example"
xlink:role="http://www.opengis.net/spec/WCS_coverage-encoding_opendap/req/dap4"
xlink:arcrole="fileReference"/>
    <gml:fileReference>cid:eo_ncep_model_example</gml:fileReference>
    <gml:fileStructure/>
  </gml:File>
</rangeSet>
```

Example: DomainSet in the CIS 1.1 XML schema have addressed similar issue

```
<complexType name="DomainSetType">
  <sequence>
    <choice>
      <element ref="cis:generalGrid"/>
      <element ref="cis:directMultiPoint"/>
      <element name="fileReference" type="anyURI"/>
    </choice>
  </sequence>
</complexType>
```

7.2.2. Time Is Another Dimension

It is also of note that in this NCEP dataset, and in virtually every other oceanographic and atmospheric dataset that the author has worked with in the past 25 years, *time* is simply another coordinate dimension. It is the authors experience that in practice *time* values are stored as simple integer or floating point values in which the units are defines as (day|hours|seconds|milliseconds) since an epoch date. In the NCEP example the units of the *time* dimension are: Hour since 2016-06-17T00:00:00Z This is relevant because it underscores the authors claim that *time* can be treated identically to the other coordinate dimensions, i.e. it can be interpolated and scaled. There are a number of use cases (for example the aviation domains use of wind forecasts) in which there is a direct need to allow *time* to be handled in this manner.

Recommendation: As much as possible treat time as just another coordinate dimension which can be acted upon with the same tools as the other coordinate dimensions. (Change RequestL http://ogc.standardstracker.org/show_request.cgi?id=410)

The important point to be consider here is not that WCS 2.0 can represent 3D or 4D data, but that EO-WCS can represent time x latitude x longitude data where time is a dimension and not an attribute. This will allow an EO-WCS service to work with more data as it natively exists (although the 'time-as-an-attribute' model works well with time-series observational data like satellite data sets).

Hypothetical CIS-1.1 representation of the NCEP dataset in which the time coordinate is "just" another coordinate dimension.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<generalGridCoverage xmlns="http://www.opengis.net/cis/1.1"
xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:swe="http://www.opengis.net/swe/2.0"
xmlns:cis="http://www.opengis.net/cis/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/cis/1.1 ../cisAll.xsd"
gml:id="cis_eo_ncep_model_example">
  <domainSet>
    <generalGrid srsName="http://www.opengis.net/def/crs/EPSG/0/4979"
axisLabels="Lat Long h">
      <cis:regularAxis axisLabel="Lat" lowerBound="-90.0" upperBound="90.0"
uomLabel="deg" resolution="0.0417"/>
    </generalGrid>
  </domainSet>
</generalGridCoverage>
```

```

    <cis:regularAxis axisLabel="Lon" lowerBound="0.0" upperBound="360.0"
uomLabel="deg" resolution="0.0417"/>
    <irregularAxis axisLabel="isobaric" uomLabel="Pa">
<c>100</c><c>200</c><c>300</c><c>500</c><c>700</c><c>1000</c><c>2000</c>
    <c>3000</c><c>5000</c><c>7000</c><c>10000</c><c>15000</c><c>20000</c>
<c>25000</c><c>30000</c><c>35000</c><c>40000</c><c>45000</c><c>50000</c>
<c>55000</c><c>60000</c><c>65000</c><c>70000</c><c>75000</c><c>80000</c>
<c>85000</c><c>90000</c><c>92500</c><c>95000</c><c>97500</c><c>100000</c>
    </irregularAxis>

    <cis:regularAxis axisLabel="time"
    lowerBound="0.0"
    upperBound="360.0"
    uomLabel="Hour since 2016-06-17T00:00:00Z"
    resolution="3"/>

    <gridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index4D"
    axisLabels="time isobaric lat lon">
    <indexAxis axisLabel="time" lowerBound="0" upperBound="111"/>
    <indexAxis axisLabel="isobaric" lowerBound="0" upperBound="30"/>
    <indexAxis axisLabel="lat" lowerBound="0" upperBound="360"/>
    <indexAxis axisLabel="lon" lowerBound="0" upperBound="719"/>
    </gridLimits>

    </generalGrid>
</domainSet>
<rangeSet>
    <gml:File>
    <gml:rangeParameters xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="cid:eo_ncep_model_example"
xlink:role="http://www.opengis.net/spec/WCS_coverage-encoding_opendap/req/dap2"
xlink:arcrole="fileReference"/>
    <gml:fileReference>cid:eo_ncep_model_example</gml:fileReference>
    <gml:fileStructure/>
    </gml:File>
</rangeSet>
<rangeType>
    <swe:DataRecord xmlns:swe="http://www.opengis.net/swe/2.0">
    <swe:field name="u_component_of_wind_isobaric">
    <swe:Quantity
definition="urn:ncep:def:independantVariable:u_component_of_wind_isobaric">
    <swe:description>u-component of wind @ Isobaric
surface</swe:description>
    <swe:uom code="m/s"/>
    <swe:constraint>
    <swe:AllowedValues>
    <swe:interval>-60.0 161.0</swe:interval>

```

```

        </swe:AllowedValues>
      </swe:constraint>
    </swe:Quantity>
  </swe:field>
  <swe:field name="v_component_of_wind_isobaric">
    <swe:Quantity
definition="urn:ncep:def:independantVariable:vu_component_of_wind_isobaric">
      <swe:description>v-component of wind @ Isobaric
surface</swe:description>
      <swe:uom code="m/s"/>
      <swe:constraint>
        <swe:AllowedValues>
          <swe:interval>-71.0 79.0</swe:interval>
        </swe:AllowedValues>
      </swe:constraint>
    </swe:Quantity>
  </swe:field>
</swe:DataRecord>
</rangeType>
</generalGridCoverage>

```

7.2.3. Order of indices, order of coordinate dimensions

One issue that comes up often in discussions of processing geo-spatial data is the order of axes—should the axis order be *longitude, latitude* or the other way around? The answer to this question is “It depends.” However, the issue becomes more complex in light of questions we were asked to examine in the Testbed 12 Large Scale Analytics thread. In this thread we looked at extending EO-WCS to support ‘3D+Time’ data. These data are effectively *stored* as four-dimensional arrays and processing them requires either writing custom code or adapting existing libraries (e.g., GDAL) to work with *slices* of these hyper-cubes. This is not so much a feature of WCS or EO-WCS as it is a pragmatic issue of how best to build services that work reliably and leverage existing software.

We have found that GDAL can be used to code the functions needed to implement EO-WCS and the 3D+Time extension, but that it requires an understanding of the library and the underlying data. A completely general solution would be either inefficient or fairly complex, so OGC should be aware that implementors will be faced with crafting code that is likely specific to data in certain sub-domains of Earth science.

Because many EO datasets do not explicitly include metadata about either their projections or coordinate reference systems (CRS, aka spatial reference systems - SRS), EO-WCS implementors should be aware that CRSs often include longitude and latitude axis order and that a global data set might rightly be thought of as using the WGS84 CRS yet not manifest with the ‘correct’ axis order. A lot of these EO data presents its axes as time, b,c,z,y,x where are other (like model data) are organized as time, b,c,z,x,y and there is no applicable widely-used CRS. In this case, either a different CRS should be used *or* the axis order of the data transformed.

Recommendation: Add an implementation note to the EO-WCS specification to this affect so that implementors will know to look out for mismatches between dataset CRSs and actual axis order.

7.3. Extension for EO-WCS *DescribeEOCoverageSet*

The *DescribeEOCoverageSet* operation embodies a simple form of inventory search. *DescribeEOCoverageSet* provides mechanisms for:

- Announcing to a client the default maximum number of returned results via the *DefaultCount* parameter returned by a *GetCapabilities* request.
- Enabling the client to request a larger (or smaller) number of results via the *count* parameter in the *DescribeEOCoverageSet* request.
- Informing the client of the total number of matching results (*numberMatched*) and the number of returned results (*numberReturned*) in the returned *EOCoverageSetDescription* document.

7.3.1. Issue: Missing Range Of Results API

What we found during our reading and implementation of the [OGC 10-140r1] *WCS Application Profile for EO* was that there does not appear to be a mechanism thorough which a client can ask for something like the next batch, or a range of results. By which we mean: If a client makes a successful *wcseo:DescribeEOCoverageSet* request and receives a *wcseo:EOCoverageSetDescription* which indicates that the *numberMatched* is quite large (say tens of thousands of coverages) and the *numberReturned* is smaller and manageable, say 200, there is no mechanism by which a client may ask for results 201- 400. A client may only request a larger number of results, not a range. This presents a real challenge for client implementations: Rather than writing their own custom XML SAX parser for the *wcseo:EOCoverageSetDescription* document, most client developers will choose to utilize a XML library with a DOM implementation. These implementations typically parse an XML document into in-memory XML objects that are then navigated through an API. The result of this is that ingesting very large XML documents can be costly, if not impossible.

Recommendation: Add to the *DescribeEOCoverageSet* request a mechanism for requesting a range of the matching results. (Change Request 405: http://ogc.standardstracker.org/show_request.cgi?id=405)

7.3.2. Issue: Search Requires Prior Knowledge Of Content.

A client must first obtain a list of *EOCoverage* identifiers, *DatasetSeries* identifiers, and/or *Stitched Mosaic* identifiers before a *wcseo:DescribeEOCoverageSet* request can be issued. Thus a client must make a *wcs:GetCapabilities* request and process the response by searching through it to identify *Coverages* and *DatasetSeries* of interest. Then the client can make a second request (*wcseo:DescribeEOCoverageSet*) to search for *EOCoverages*, *DatasetSeries*, and *Stitched Mosaics* that match the client supplied subset criteria. This is reasonable, but somewhat redundant since the *CoverageSummary* elements retrieved from *wcs:GetCapabilities* response already contain the information the client would require to evaluate a coverages match against the subset information. This pattern is more natural for the *DatasetSeries* IDs because they may represent a collection of *EOCoverages*, *DatasetSeries*, and *Stitched Mosaic* coverages each of which may or may not appear in the *wcs:GetCapabilities* response.

Recommendation: The authors feel that *DescribeEOCoverageSet* activity might be of more use to

the client if the client need only to supply the subset conditions, and not a list of identifiers. (Change Request 406: http://ogc.standardstracker.org/show_request.cgi?id=406)

7.4. Extension for WCS *GetCoverage*

While implementing the *GetCoverage* operation for higher dimensional datasets we discovered that the existing WCS-2.0 request interface provided an adequate syntax for subsetting higher dimensional data.

7.4.1. Scaling Extension [OGC 12-039]

Scaling (a.k.a. re-gridding) appears to be a natural fit for the EO-WCS model. The Scaling extension makes good sense up to the specification of the SCALEEXTENT activity. In our reading of the specification the authors failed to understand how to implement the *Scale Extent* although it may actually be simple. Examples of scale extent imply that clients provide the service with indices of an axis, like this:

```
...& SCALEEXTENT=i(10:20),j(20:30) &...
```

However, this also implies a kind of trim operation. Unfortunately, the mathematical description of the SCALEEXTENT feature is hard to understand. Perhaps it would be helpful if the WCS 2.0 Scale Extension specification had one or more examples of SCALEEXTENT that illustrate how it should be used and how it modifies the returned data.

[OGC 12-039] Requirement 15 *getCoverage-scale-to-extent-result*

```
Let c be a coverage with dimension d, grid domain  $X = \{l_1, \dots, h_1\} \times \dots \times \{l_d, \dots, h_d\}$ ,  
range set  $R = \{(v, p) \mid \text{where } p = (x_1, \dots, x_d) \text{ and } l_i \leq x_i \leq h_i \text{ for } 1 \leq i \leq d\}$ ;
```

```
let further  $s = \{(a, s_l, s_h)\}$  be a valid GetCoverage request scale interval set with  
each  
 $a_i$  being the identifier of a Grid axis in c, and  $s_{li} \leq s_{hi}$  being target domain  
intervals.
```

```
Then, the response coverage to a successful GetCoverage request scaling coverage c by  
scale  
vector s shall have dimension d, Grid domain  $X'$  of dimension d where the domain extent  
is the same as in c for all those axes not mentioned in s, and set to  $[s_{li}, \dots, s_{hi}]$   
for all axes  $a_i$  occurring in s, range set  $R' = \{(v', p') \mid p' = (x'_1, \dots, x'_d) \text{ } X' \text{ and } v' \text{ is a function of the corresponding original location } p \text{ and possibly a neighborhood of } p\}$ , where any eventual subsetting of the coverage is done before this scale  
operation.
```

Recommendation: The WCS Scale Extension specification [OGC 12-039] should be changed to include a simpler explanation or SCALEEXTENT should be dropped from the specification as the other scaling and subsetting commands appear to be more than adequate for the desired outcomes. (Change Request: http://ogc.standardstracker.org/show_request.cgi?id=407)

7.4.2. Regarding SlicePoint Subsetting Restrictions

Section 7.4.1 of [OGC 10-140r1] *WCS Application Profile for EO* states:

"The GetCoverage request is unchanged over WCS Core [OGC 09-110r4], except that for EO Coverages slicing is disallowed as it would leave the EO Metadata undefined."

In short: No SlicePoint subsetting because it breaks the EO-WCS metadata description of the object.

Given that WCS is, at its heart, a data access protocol, the authors believe that this restriction is a bad choice because it sacrifices data access in order to protect the metadata integrity of the WCS service. Additionally, in the case of servers that are returning complex data objects like NetCDF, DAP2, and DAP4 the response comes, in every case, with syntactic metadata that will **always** be correct for the returned object, regardless of any issues with EO metadata delivered in the Coverage section of the response. The NetCDF and DAP4 responses may also contain additional semantic metadata describing the returned object, and this metadata may or may not be correct due to various subsetting and scaling activities. The result is that prohibiting SlicePoint subsetting in no way protects the various metadata chains from inaccuracy. And in practice, how is a "SlicePoint" different from a "Trim" in which max=min?

The authors believe that lenient data access syntax semantics are preferable to no access, and that disallowing SlicePoint subsetting blocks access and ultimately fails to preserve metadata integrity for the various complex data formats returned by the EO-WCS service.

Recommendation: Allow SlicePoint subsetting. (Change Request 408: http://ogc.standardstracker.org/show_request.cgi?id=408)

7.4.3. Data Request API Examples

OPeNDAP request interface

Here we examine an OPeNDAP data access request for NCEP wind velocity data. This request utilizes a server side function, `grid()`, to subset the data by value in the *isobaric*, *latitude*, and *longitude* dimensions. This is coupled with an array subsetting expression to retrieve a single slice of the *time* dimension. The result is returned as a NetCDF file.

DAP2 Data Request Subsetting By Value And Array Index.

```
http://testbed-12.opendap.org:8080/opendap/testbed-12/Global_0p25deg_best_hs002.nc.nc?grid(u-component_of_wind_isobaric,"41.9<=lat<=46.3","0.0<=isobaric<=500.0","55.0<=lon<=63.6"),grid(v-component_of_wind_isobaric,"41.9<=lat<=46.3","0.0<=isobaric<=500.0","55.0<=lon<=63.6"),u-component_of_wind_isobaric[0][*][*][*],v-component_of_wind_isobaric[0][*][*][*])
```

Here is an example of a more traditional OPeNDAP request URL that produces the same result as the previous example, but achieves this through the subsetting by array index syntax:

DAP2 Data Request Subsetting By Index

```
http://testbed-12.opendap.org:8080/opendap/testbed-12/ncep/Global_0p25deg_best_hs002.nc.nc?u-component_of_wind_isobaric[0][0:1:3][87:1:97][110:1:128],v-component_of_wind_isobaric[0][0:1:3][87:1:97][110:1:128]
```

DAP4 Data Request Subsetting By Index

```
http://testbed-12.opendap.org:8080/opendap/testbed-12/ncep/Global_0p25deg_best_hs002.nc.dap.nc?dap4.ce=u-component_of_wind_isobaric[0][0:1:3][87:1:97][110:1:128];v-component_of_wind_isobaric[0][0:1:3][87:1:97][110:1:128]
```

WCS Equivalent Request Interface

In this example we subset the NCEP dataset wind velocity coverage by specifying bound values for the latitude, longitude, and isobaric dimensions. We subset the time dimension using array index subsetting to extract a slicePoint (a single time point) from the dataset. The data is returned as a NetCDF file.

WCS Data Access Request, Subsetting By Value And Array Index.

```
http://testbed-12.opendap.org:8080/WCS-2.0?service=WCS&version=2.0.1&request=GetCoverage&coverageId=eo_ncep_model_example&subset=longitude(55.0,63.6)&subset=latitude(41.9,46.3)&subset=isobaric(0.0,500.0)&subset=time(0)&format=application/x-netcdf-cf1.0
```

Executing this request returns a NetCDF file. A cursory inspection of the file using the NetCDF command line tool *ncdump* shows that the returned data object appears to be correct, at least with respect to the syntactic metadata matching the expected result:

```
[-bash: ~] ncdump -h ncep_subset.nc
netcdf ncep_subset {
dimensions:
    time = 1 ;
    isobaric = 4 ;
    lat = 11 ;
    lon = 19 ;
variables:
    double time(time) ;
        time:units = "Hour since 2016-06-17T00:00:00Z" ;
```

```

time:standard_name = "time" ;
time:long_name = "GRIB forecast or observation time" ;
time:calendar = "proleptic_gregorian" ;
time:_CoordinateAxisType = "Time" ;
float isobaric(isobaric) ;
isobaric:units = "Pa" ;
isobaric:long_name = "Isobaric surface" ;
isobaric:positive = "down" ;
isobaric:Grib_level_type = 100 ;
isobaric:_CoordinateAxisType = "Pressure" ;
isobaric:_CoordinateZisPositive = "down" ;
float lat(lat) ;
lat:units = "degrees_north" ;
lat:_CoordinateAxisType = "Lat" ;
lat:standard_name = "latitude" ;
float lon(lon) ;
lon:units = "degrees_east" ;
lon:_CoordinateAxisType = "Lon" ;
lon:standard_name = "longitude" ;
float u-component_of_wind_isobaric(time, isobaric, lat, lon) ;
u-component_of_wind_isobaric:long_name = "u-component of wind @ Isobaric
surface" ;
u-component_of_wind_isobaric:units = "m/s" ;
u-component_of_wind_isobaric:abbreviation = "UGRD" ;
u-component_of_wind_isobaric:missing_value = 0.f ;
u-component_of_wind_isobaric:grid_mapping = "LatLon_Projection" ;
u-component_of_wind_isobaric:coordinates = "reftime time isobaric lat lon lat
lon" ;
u-component_of_wind_isobaric:Grib_Variable_Id = "VAR_0-2-2_L100" ;
u-component_of_wind_isobaric:Grib2_Parameter = 0, 2, 2 ;
u-component_of_wind_isobaric:Grib2_Parameter_Category = "Momentum" ;
u-component_of_wind_isobaric:Grib2_Parameter_Name = "u-component of wind" ;
u-component_of_wind_isobaric:Grib2_Level_Type = "Isobaric surface" ;
u-component_of_wind_isobaric:Grib2_Generating_Process_Type = "Forecast" ;
float v-component_of_wind_isobaric(time, isobaric, lat, lon) ;
v-component_of_wind_isobaric:long_name = "v-component of wind @ Isobaric
surface" ;
v-component_of_wind_isobaric:units = "m/s" ;
v-component_of_wind_isobaric:abbreviation = "VGRD" ;
v-component_of_wind_isobaric:missing_value = 0.f ;
v-component_of_wind_isobaric:grid_mapping = "LatLon_Projection" ;
v-component_of_wind_isobaric:coordinates = "reftime time isobaric lat lon lat
lon" ;
v-component_of_wind_isobaric:Grib_Variable_Id = "VAR_0-2-3_L100" ;
v-component_of_wind_isobaric:Grib2_Parameter = 0, 2, 3 ;
v-component_of_wind_isobaric:Grib2_Parameter_Category = "Momentum" ;
v-component_of_wind_isobaric:Grib2_Parameter_Name = "v-component of wind" ;
v-component_of_wind_isobaric:Grib2_Level_Type = "Isobaric surface" ;
v-component_of_wind_isobaric:Grib2_Generating_Process_Type = "Forecast" ;

// global attributes:

```

```

:Originating_or_generating_Center = "US National Weather Service, National
Centres for Environmental Prediction (NCEP)" ;
:Originating_or_generating_Subcenter = "0" ;
:GRIB_table_version = "2,1" ;
:Type_of_generating_process = "Forecast" ;

:Analysis_or_forecast_generating_process_identifier_defined_by_originating_centre =
"Analysis from GFS (Global Forecast System)" ;
:Conventions = "CF-1.6" ;
:history = "Read using CDM IOSP GribCollection v3" ;
:featureType = "GRID" ;
:History = "Translated to CF-1.0 Conventions by Netcdf-Java CDM
(CFGridWriter2)\012Original Dataset =
/data/ldm/pub/native/grid/NCEP/GFS/Global_0p25deg/GFS-Global_0p25deg.ncx3; Translation
Date = 2016-06-20T21:22:16.485Z" ;
:geospatial_lat_min = 0. ;
:geospatial_lat_max = 0. ;
:geospatial_lon_min = 0. ;
:geospatial_lon_max = 0. ;
}

```

7.5. Demonstration

Here we provide a simple demonstration of the servers subsetting capabilities.

7.5.1. Index base subsetting

Using an array index based subset expression for the *time* axis we produced a visualization of the NCEP Global 0.25 deg wind data for the second time instance (index 1) in the dataset. It represents the two component of the wind: *u* component and *v* component. *u* is the zonal velocity (component of horizontal wind towards East) and *v* is the meridional velocity (component of horizontal wind toward North).

The data were retrieved in NetCDF format from the test server using the following hand built WCS request:

```

http://testbed-12.opendap.org:8080/WCS-2.0?
  service=WCS&version=2.0.1
  &request=GetCoverage
  &coverageId=eo_ncep_model_example
  &format=application/x-netcdf-cf1.0
  &subset=time(1)

```

The returned response was written to a local file and then opened with ncBrowse (graphical netCDF file browser) which was used to produce the images in Figures 2 & 3.

Global_0p25deg_best_hs002.nc.nc
u-component of wind @ Isobaric surface

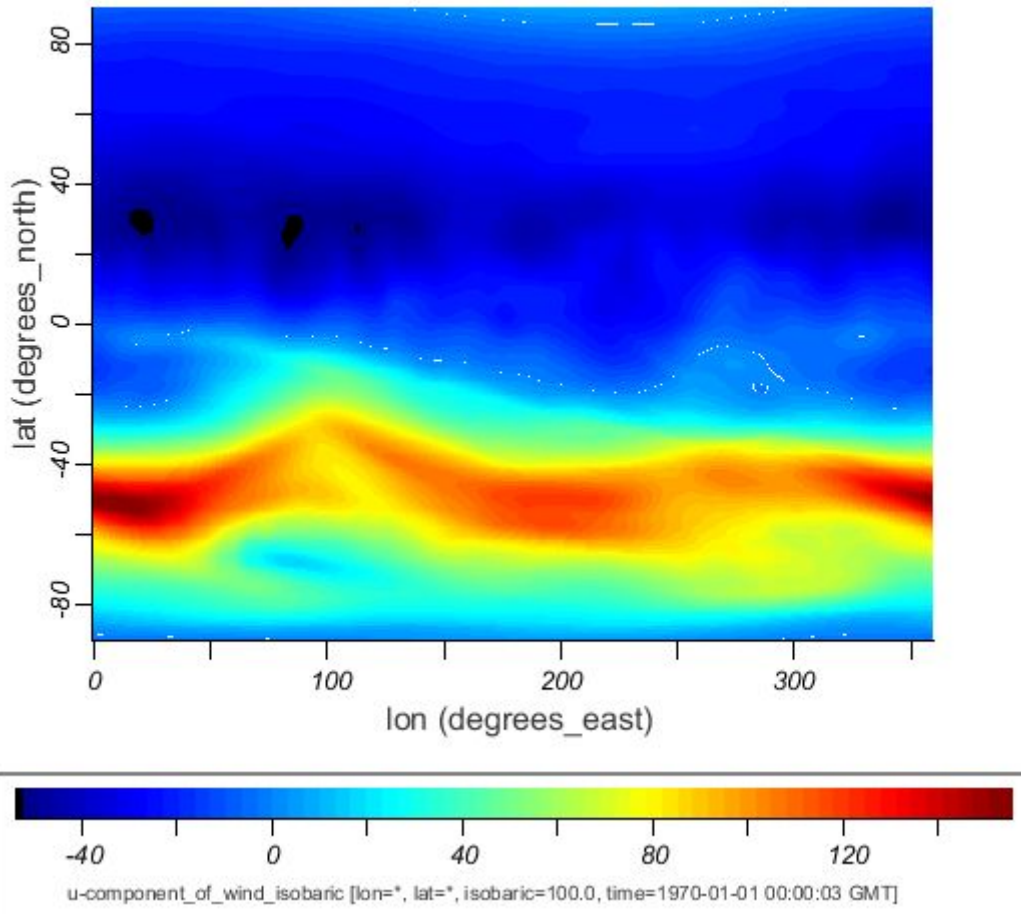


Figure 2: Global 25 Deg U-Component of Wind (single time point and no geographic constraint)

Global_0p25deg_best_hs002.nc.nc
v-component of wind @ Isobaric surface

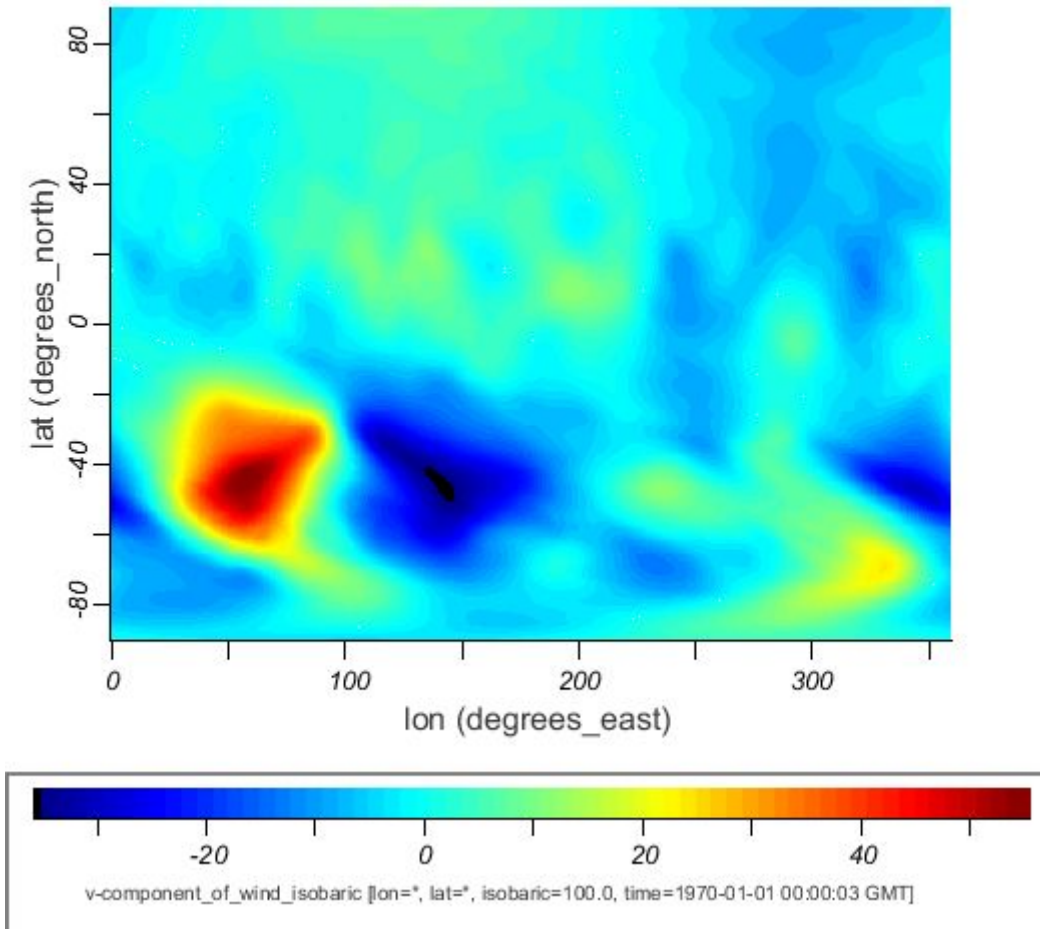


Figure 3: Global 25 Deg V-Component of Wind (single time point and no geographic constraint)

7.5.2. Value based subsetting

In order to demonstrate the value and index based subsetting capabilities we used the following hand written WCS subset request to retrieve data in NetCDF format:

WCS Data Access Request, Subsetting By Value And Array Index.

```
http://testbed-12.opendap.org:8080/WCS-2.0?  
service=WCS  
&version=2.0.1  
&request=GetCoverage  
&coverageId=eo_ncep_model_example  
&subset=longitude(55.0,63.6)  
&subset=latitude(41.9,46.3)  
&subset=isobaric(0.0,500.0)  
&subset=time(0)  
&format=application/x-netcdf
```

In this request the *longitude*, *latitude*, and *isobaric* axes are subset by value (Indicated by the floating-point values) and *time* axis is subset by array index (indicated by the integer value).

The *format* parameter requests the response as a NetCDF file.

The returned response was written to a local file and then opened with **Panoply** which was used to produce the images in Figures 4 & 5.

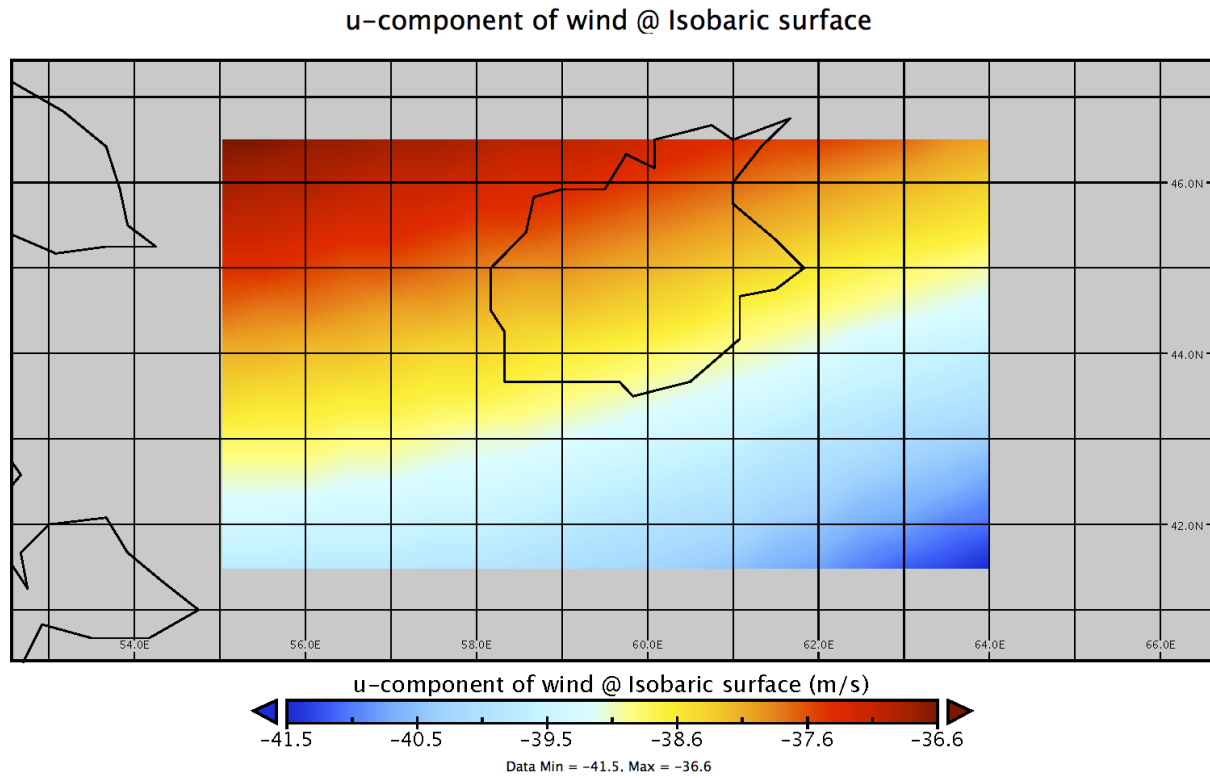


Figure 4: Global 25 Deg U-Component of Wind (single time point with geographic and isobaric constraint)

v-component of wind @ Isobaric surface

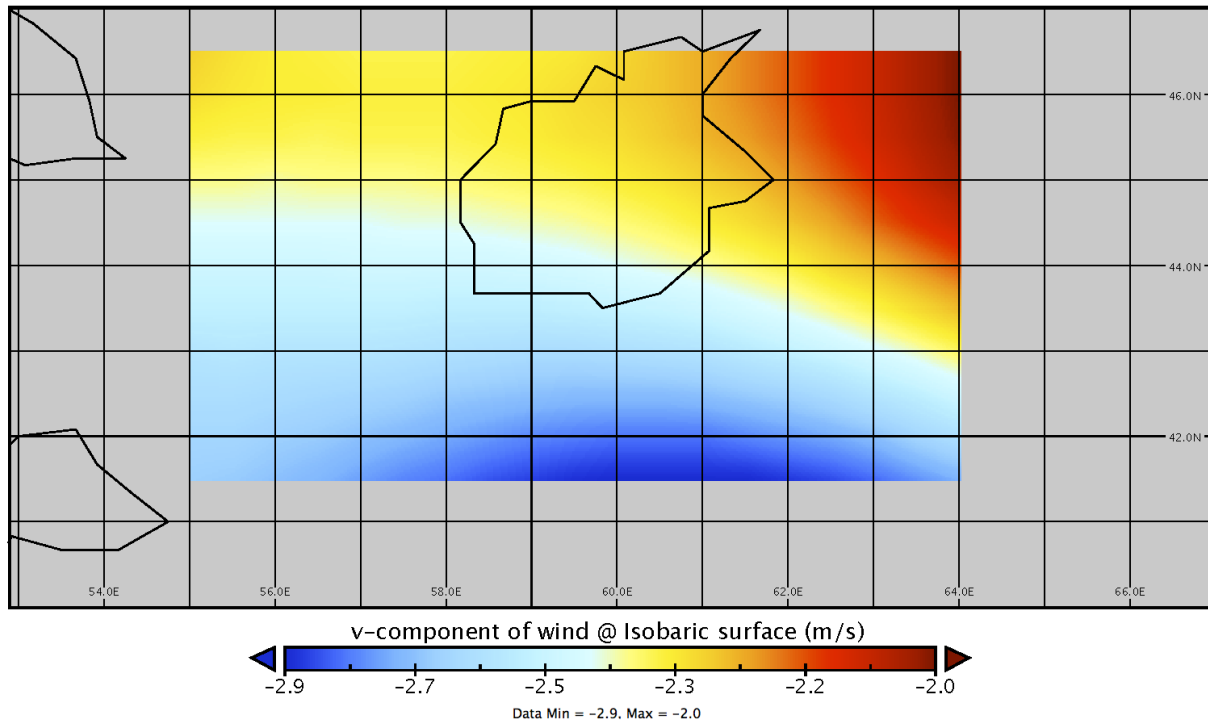


Figure 5: Global 25 Deg V-Component of Wind (single time point with geographic and isobaric constraint)

Chapter 8. EO-WCS Update to Support 3D Coverages

8.1. EO-WCS Update

The proposed EO-WCS 1.1 [OGC 10-140r2] update is edited as AsciiDoc just like this document and available online as [rendered HTML version](#) or [source code](#).

EO-WCS 1.1 drops the requirement for 2D coverages. In fact it explicitly allows coverages with more dimensions as long as they have a geographic footprint on Earth. The proposed changes can be seen in [this commit](#).

It further clarifies the usage of `rangeType` when non-NCNAME characters are present in a band identifier. Again, the proposed changes can be seen in [this commit](#).

8.2. Use Case - Fire Emissions, San Francisco

8.2.1. Data

The data is simulated (FLEXPART) dispersion of fire emissions. The fire burned one day (from 0 to 24h) on 2016-02-23 in the northeastern part of San Francisco. Three simulations for consecutive days are available and the emissions transport is simulated for three days each. The temporal resolution is 3h for 10 height levels each.

The license is CC-BY using "DLR-DFD (German Aerospace Center — German Remote Sensing Data Center)". Many thanks to DLR for providing the data.

The data of all three simulations is made available via EO-WCS. The 24 coverages of each simulation are grouped together as DatasetSeries. Please follow the procedure below for retrieving the data.

8.2.2. *GetCapabilities* Request

Start with a *GetCapabilities* request to obtain the DatasetSeries IDs of the three simulations.

Request

<http://ows.eox.at/testbed-12/eoxserver/ows?service=wcs&request=getcapabilities>

DatasetSeries Summary

```

<wcs:Contents>
  <wcs:Extension>
    <wcseo:DatasetSeriesSummary>
      <ows:WGS84BoundingBox>
        <ows:LowerCorner>0.000000 -135.000000</ows:LowerCorner>
        <ows:UpperCorner>45.000000 -90.000000</ows:UpperCorner>
      </ows:WGS84BoundingBox>

      <wcseo:DatasetSeriesId>dlr_fire_emission_dispersion_california_20160223</wcseo:Dataset
      SeriesId>
      <gml:TimePeriod
      gml:id="dlr_fire_emission_dispersion_california_20160223_timeperiod">
        <gml:beginPosition>2016-02-23T03:00:00Z</gml:beginPosition>
        <gml:endPosition>2016-02-27T00:00:00Z</gml:endPosition>
      </gml:TimePeriod>
      </wcseo:DatasetSeriesSummary>
      .
      .
    </wcs:Extension>
  </wcs:Contents>

```

8.2.3. DescribeEOCoverageSet Request

In a second step use the DatasetSeries IDs, e.g., `dlr_fire_emission_dispersion_california_20160223` in a *DescribeEOCoverageSet* request to obtain detailed descriptions of the included coverages. Potentially include a temporal subset for the time you're interested in.

Request

http://ows.eox.at/testbed-12/eoxserver/ows?service=wcs&version=2.0.1&request=describeocoverageset&eoid=dlr_fire_emission_dispersion_california_20160223

Including a temporal subset:

[http://ows.eox.at/testbed-12/eoxserver/ows?service=wcs&version=2.0.1&request=describeocoverageset&eoid=dlr_fire_emission_dispersion_california_20160223&subset=phenomenonTime\("2016-02-23T03:00:00Z","2016-02-23T03:00:00Z"\)](http://ows.eox.at/testbed-12/eoxserver/ows?service=wcs&version=2.0.1&request=describeocoverageset&eoid=dlr_fire_emission_dispersion_california_20160223&subset=phenomenonTime(\)

Coverage Description

The omitted parts are provided separately below.

```

<wcs:CoverageDescription gml:id="dlr_fire_emission_dispersion_california_20160223_1">
  <gml:boundedBy>
    <gml:Envelope axisLabels="lat long height"
srsName="http://www.opengis.net/def/crs/EPSSG/0/4979" uomLabels="deg deg m"
srsDimension="3">
      <gml:lowerCorner>-0.00000001 -135.00000001 0</gml:lowerCorner>
      <gml:upperCorner>45.00000000 -89.99999999 1800</gml:upperCorner>
    </gml:Envelope></gml:boundedBy>

  <wcs:CoverageId>dlr_fire_emission_dispersion_california_20160223_1</wcs:CoverageId>
  <gmlcov:metadata>
    <gmlcov:Extension>
      <wcseo:EOMetadata>
        <eop:EarthObservation
gml:id="eop_dlr_fire_emission_dispersion_california_20160223_1">
          <om:phenomenonTime>
            .
          </om:phenomenonTime>
        </eop:EarthObservation>
      </wcseo:EOMetadata>
    </gmlcov:Extension>
  </gmlcov:metadata>
  <gml:domainSet>
    .
  </gml:domainSet>
  <gmlcov:rangeType>
    <swe:DataRecord>
      .
    </swe:DataRecord>
  </gmlcov:rangeType>
  <wcs:ServiceParameters>
    <wcs:CoverageSubtype>RectifiedDataset</wcs:CoverageSubtype>
    <wcs:nativeFormat>image/tiff</wcs:nativeFormat>
  </wcs:ServiceParameters>
</wcs:CoverageDescription>
</wcs:CoverageDescriptions>

```

Time Reference

```

<om:phenomenonTime>
  <gml:TimePeriod
gml:id="phen_time_dlr_fire_emission_dispersion_california_20160223_1">
  <gml:beginPosition>2016-02-23T03:00:00Z</gml:beginPosition>
  <gml:endPosition>2016-02-23T03:00:00Z</gml:endPosition>
  </gml:TimePeriod>
</om:phenomenonTime>
<om:resultTime>
  <gml:TimeInstant
gml:id="res_time_dlr_fire_emission_dispersion_california_20160223_1">
  <gml:timePosition>2016-02-23T03:00:00Z</gml:timePosition>
  </gml:TimeInstant>
</om:resultTime>

```

Domain Reference

```

<gml:domainSet>
  <gml:RectifiedGrid
gml:id="dlr_fire_emission_dispersion_california_20160223_1_grid" dimension="3">
  <gml:limits>
    <gml:GridEnvelope>
      <gml:low>0 0 0</gml:low>
      <gml:high>359 359 9</gml:high>
    </gml:GridEnvelope>
  </gml:limits>
  <gml:axisLabels>lat long height</gml:axisLabels>
  <gml:origin>
    <gml:Point srsName="http://www.opengis.net/def/crs/EPSSG/0/4979"
gml:id="dlr_fire_emission_dispersion_california_20160223_1_grid_origin">
      <gml:pos>45.00000000 -135.00000000 0</gml:pos>
    </gml:Point>
  </gml:origin>
  <gml:offsetVector
srsName="http://www.opengis.net/def/crs/EPSSG/0/4979">0.00000000 0.12500000
0</gml:offsetVector>
    <gml:offsetVector srsName="http://www.opengis.net/def/crs/EPSSG/0/4979">-
0.12500000 0.00000000 0</gml:offsetVector>
    <gml:offsetVector srsName="http://www.opengis.net/def/crs/EPSSG/0/4979">0 0
200</gml:offsetVector>
  </gml:RectifiedGrid>
</gml:domainSet>

```

Range Reference

```

<swe:DataRecord>
  <swe:field name="fire_emission_b1">
    <swe:Quantity definition="">
      <swe:description>Fire_emission_Band_1</swe:description>
      <swe:nilValues>
        <swe:nilValues>
          <swe:nilValue
reason="http://www.opengis.net/def/nil/OGC/0/inapplicable">0</swe:nilValue>
          </swe:nilValues>
        </swe:nilValues>
      <swe:uom code="nanograms/m3"/>
      <swe:constraint>
        <swe:AllowedValues>
          <swe:interval>-3.40282e+38 3.40282e+38</swe:interval>
          <swe:significantFigures>38</swe:significantFigures>
        </swe:AllowedValues>
      </swe:constraint>
    </swe:Quantity>
  </swe:field>
  <swe:field name="fire_emission_b2">
    <swe:Quantity definition="">
      <swe:description>Fire_emission_Band_2</swe:description>
      <swe:nilValues>
        <swe:nilValues>
          <swe:nilValue
reason="http://www.opengis.net/def/nil/OGC/0/inapplicable">0</swe:nilValue>
          </swe:nilValues>
        </swe:nilValues>
      <swe:uom code="nanograms/m3"/>
      <swe:constraint>
        <swe:AllowedValues>
          <swe:interval>-3.40282e+38 3.40282e+38</swe:interval>
          <swe:significantFigures>38</swe:significantFigures>
        </swe:AllowedValues>
      </swe:constraint>
    </swe:Quantity>
  </swe:field>
  .
  .
  .
  .
</swe:DataRecord>

```

8.2.4. *GetCoverage* Request

Finally use the detailed information to retrieve the actual data via *GetCoverage* requests. Note that the provided service offers the data encoded as GeoTIFFs for individual timestamps including the 10 height layers as bands. The GeoTIFF standard has somehow been misused by using the bands for the third dimension. However, as shown by the demonstration described below this approach is

feasible even including on-the-fly rendering in a Web browser.

Request

http://ows.eox.at/testbed-12/eoxserver/ows?service=wcs&version=2.0.1&request=getcoverage&coverageid=dlr_fire_emission_dispersion_california_20160223_1

Response

A GeoTIFF file with 10 Float32 bands, one for each height. The Float32 values represent the fire emission in nanograms/m³.

8.2.5. Demonstration

The demonstration of the visualization of 3D data in the browser making extensive use of WebGL as shown in the figure below is available at <http://ows.eox.at/testbed-12/eox/>.

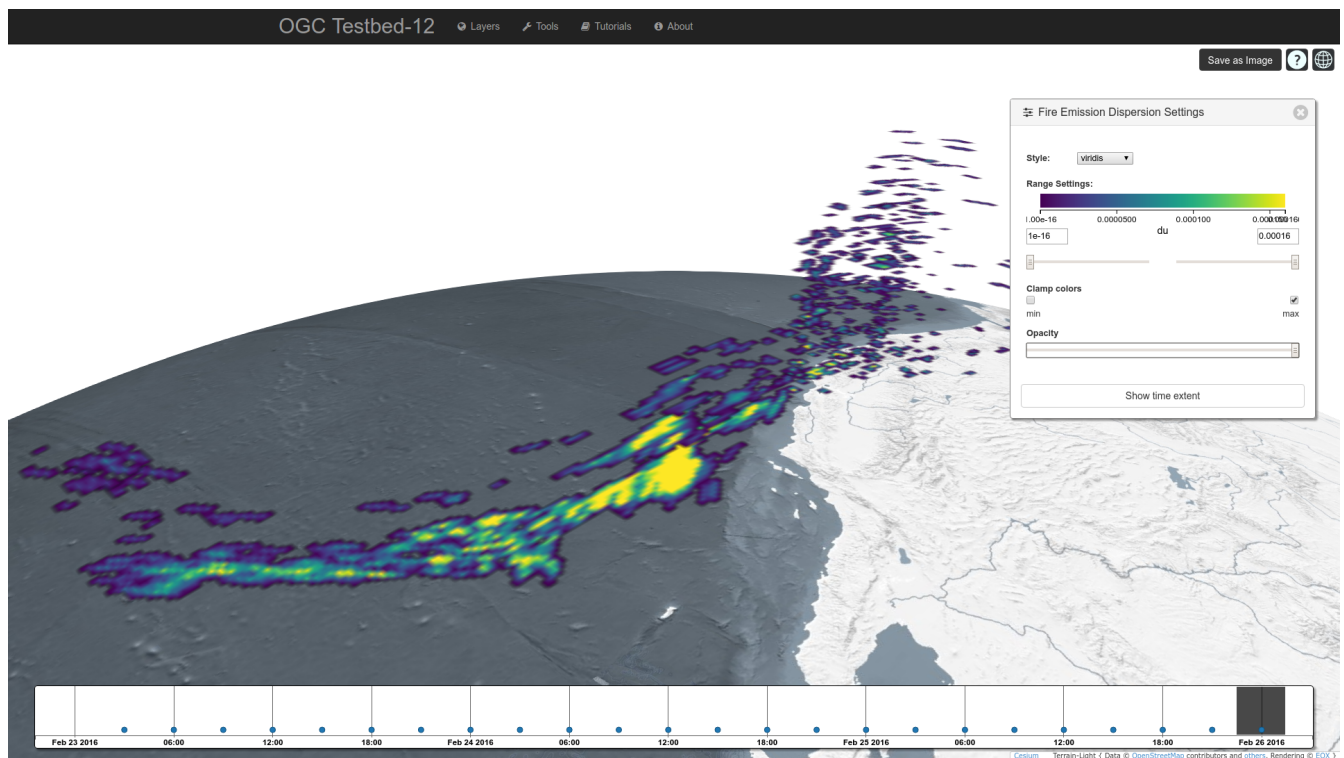


Figure 6: FLEXPART simulation of dispersion of fire emissions

Chapter 9. WCS PYXIS Client Testing

Discussion

This chapter will try to explore some of the knowledge gathered during the client testing performed by PYXIS and highlight future potential recommendations/changes for improvement.

9.1. Additional metadata information on WCS GetCapabilities

The WCS *getCapabilities* web-api is designed to give clients an overview of available datasets. However, *CoverageSummary* and *DatasetSeriesSummary* provide an incomplete information set if the client is to display a meaningful representation of the coverage (compared to WFS/WCS metadata like name/description/tags). Moreover, in this Testbed we used 3d/4d coverages, which require a client to perform *DescribeCoverage* in order to understand if the coverage has more domains. This can become an issue when a client need to perform *DescribeCoverage* for each coverage just to display the list of available datasets and their domains.

The *DescribeEOCoverageSet* returns the complete metadata and grid information for all related coverages, which is a single request that allow the client to get all information available about this set.

Moving forward, it will be helpful for the client to know if a coverage is 2D or 3D, has a single field or a more complex rangeSet from the *GetCapabilities* response so it can limit the number of *DescribeCoverage* requests needed just to construct a list available coverages (and what they represent) on the server.

9.2. WCS 2.0.1 request simplified by using only native grid coordinates

The WCS standard allows the client to utilize a single subsetting syntax: *subset=axis(low,high)*. Previous versions of WCS had different methods to deal with domain coordinate subsets and time domain subsets. Treating all axis equally will reduce code complexity on client side.

9.3. WCS 2.0.1 additional metadata for displaying meaningful native gird coordinates.

EO-WCS can represent time as a collection of 2D coverages, and the EO Metadata contains the time information to display for the user. However, 3d data (elevation) or 4d data (elevation and time) can only be represented as 3D grids or 4D grids.

Here is an example of a 4D coverage:

```

<gml:GridEnvelope>
  <gml:low>0 0 0 0</gml:low>
  <gml:high>111 30 360 719</gml:high>
</gml:GridEnvelope>
<gml:axisLabels>time isobaric latitude longitude</gml:axisLabels>

```

The client can show the following information:

1. The given data is 4D
2. Time and isobaric have an index range of (0...111) and (0..30)
3. Using the *gml:Envelope* - we can also display the spatial resolution of the data.

However, the user can't easily determine the values of time[7] or isobaric[12]. We can add some display information of the variable such as for isobaric(12) = "45000 f" (isobaric unit) and time(0) = "2016-06-17T00:00:00Z", perhaps similar to the WMS <domain>.

9.4. Detecting lat/lon of axes in WCS

WCS (OGC in general) generally recognize latitude, longitude, and time as special coordinates. Yet it attempts to treat them in a generic manner using general coordinate domain representations. Perhaps the OGC should commit to dedicated special representations whose semantics are well defined.

Sample example:

```

<RectifiedGrid>
  <TimeAxis start="0" delta="3.0" units="Hours since 04/01/2016T00:00:00Z" />
  <LatitudeAxis start="90.0" count="720" delta="-0.25" />
  <LongitudeAxis start="20" count="1600" delta="0.25" />
</RectifiedGrid >

```

Currently when a client accesses a WCS server it is required to manually parse an XML attribute value (axisLabels) to understand which axis is latitude or longitude. The current client is doing this by comparing the axisLabel to known strings like "latitude", "y", "lat" etc. This could be improved by automatic detection of lat/lon axes, and same could be done to time axes as well.

9.5. Non clear treatment of XY and lat/lon axes

The order of latitude and longitude axes, and the meaning of those fields keep changing between different versions and implementations of WCS. Even the three examples used by OPeNDAP encountered this issue.

The inundation model data (newSF_1in100.nc) is lon x lat:

```
Float64 depth[time = 108][longitude = 4800][latitude = 4800];
```


The NCEP Model data is lat x lon:

```
Float32 v-component_of_wind_isobaric[time = 112][isobaric = 31][lat = 361][lon = 720];
```

And the MODIS L3 data is also lat x lon:

```
Float32 l3m_data[time = 1][lat = 4320][lon = 8640];
```

Therefore, the current PYXIS client develop a custom GDAL driver to perform "WCS handshake" with the server to verify the assumptions and implementation details of the server.

Here is a brief overview on the handshake process:

- Parse *DescribeCoverage* and generate a *geoTransform*.
- Check if axis dx/dy are flipped. Example:

```
<gml:RectifiedGrid gml:id="dlr_fire_emission_dispersion_california_20160223_10_grid"
dimension="2">
  <gml:limits>
    <gml:GridEnvelope>
      <gml:low>0 0</gml:low>
      <gml:high>359 359</gml:high>
    </gml:GridEnvelope>
  </gml:limits>
  <gml:axisLabels>lat long</gml:axisLabels>
  <gml:origin>
    <gml:Point srsName="http://www.opengis.net/def/crs/EPSSG/0/4326"
gml:id="dlr_fire_emission_dispersion_california_20160223_10_grid_origin">
      <gml:pos>45.00000000 -135.00000000</gml:pos>
    </gml:Point>
  </gml:origin>
  <gml:offsetVector srsName="http://www.opengis.net/def/crs/EPSSG/0/4326">0.00000000
0.12500000</gml:offsetVector>
  <gml:offsetVector srsName="http://www.opengis.net/def/crs/EPSSG/0/4326">-0.12500000
0.00000000</gml:offsetVector>
</gml:RectifiedGrid>
```

In this example the *offsetVector* shows that the grid is in lon/lat order.

- Request lat(0,3) and lon(0,2) from the server. This request is used to validate the GeoTransform and XY order. Requesting a 4x3 pixel image allow us to detect the axis swap.
 - We detect if we receive an image that is 3x4 and mark this server as swaps XY
 - Most dataset return GeoTransform information with the response so we validate the GeoTransform result. This step can also detect lat/lon swapping.
- Request lat(0,7) and lon(0,5) with scale down by 0.5 which is expected to get back 4x3 pixels

image. we can now verify if scaling works on this server. Additionally some servers have different XY and LatLon swap which can detect as well.

Chapter 10. Future Recommendations

10.1. GRIB Encoding

General Regularly-distributed Information in Binary form (GRIB) as defined by WMO (World Meteorological Organization) is a binary encoding of gridded data which defines the international gridded data exchange standard for all national meteorological organization. The WCS2.0 could be extended offering GRIB encoded coverages for numerous standard meteorological visualization packages, like ADAGUC, Diana, MetView, Ninjo, Synergie, and Visual Weather, along with standard GIS packages like ArcGIS, Gaia, and QGIS. Successful implementation will be to correctly produce visualization of various coverage requests.

10.2. Coverage Collections Extension

Support coverage collection to the various standard GIS packages as well as meteorological visualization packages mentioned in the above section. Success of the implementation will be judged based on accurate production of the visualization of the various requested coverage collections by the displayed packages.

10.3. MetOcean Profile

Support MetOcean data profiles such as vertical slices, trajectories, ‘curtains’, and ‘Swiss cheese’ as encoded coverages to various GIS standard packages as well as meteorological visualization packages. These metadata coverages need to be validate with standard catalogues such as CSW, OPeNDAP, Thredds. Successful implementation will be based on correctness of the package display and visualization from the various profile requests and the interoperability of the output packages.

10.4. Others

- Review and validate the results achieved in the proposed EO-WCS extensions to broader context in an OGC Testbed, mainly including CITE testing.
- Improvement and harmonization of rangeType descriptions of Coverage data and values.

Appendix A: Revision History

Table 2. Revision History

Date	Release	Editor	Primary clauses modified	Descriptions
April 04, 2016	R. Shrestha	0.0	all	initial version, outline
April 12, 2016	R. Shrestha	0.1	all	updated content
June 28, 2016	R. Shrestha	1.0	all	Preliminary Draft
August 22, 2016	R. Shrestha	1.1	Requirements, References	Updated Draft
September 30, 2016	R. Shrestha	2.0	All	Draft Submitted for Review
October 24 , 2016	N. Potter, J.Gallagher	2.0	Contribution on requirement section of the <i>clause-requirement.adoc</i> and implementation work on <i>Clause-requirement-OPeNDAP.adoc</i>	Merged into the main repo
October 24 , 2016	S. Meissl	2.0	Contribution on requirement section of the <i>clause-requirement.adoc</i> and implementation work on <i>Clause-requirement-EO-WCS.adoc</i>	Merged into the main repo
October 25, 2016	R. Shrestha	2.0	Compile contributions from OPeNDAP and EOX, minor updates on text and formatting of the entire document	Draft Submitted for internal Review

Date	Release	Editor	Primary clauses modified	Descriptions
October 31, 2016	R. Shrestha	3.0	Compile contributions from OPeNDAP and PYXIS, Added client testing discussion chapter plus some minor editing	ER submitted in pending document
November 11, 2016	R. Shrestha	3.0	Compile contributions from OPeNDAP addressing feedback on the ER	ER submitted in pending document

Appendix B: Bibliography

- [1] Baart F, de Boer G, de Haas W, Donchyts G, Philippart M, van Koningsveld M, Plieger M.: A Comparison between WCS and OPeNDAP for Making Model Results and Data Products Available through the Internet. Transactions in GIS, 16(2), pp.249-265 (2012).
- [2] OGC 12-095. "OWS-9 Innovation - Coverages: Coverage Access (OPeNDAP) Study." June 18, 2013
- [3] Unidata, Accessed on 06/28/16 at: <http://www.unidata.ucar.edu/software/netcdf/>
- [4] OPeNDAP, Accessed on 06/28/16 at: <http://opendap.org/about>
- [5] EARTHDATA, Accessed on 06/28/16 at: <https://earthdata.nasa.gov/standards/data-access-protocol-2>