Open Geospatial Consortium Inc.

Date: 2011-August 27

Reference number of this document: OGC 11-010

Version: **2.0**

Category: OpenGIS® Discussion Paper

Editors: Ben Domenico and Stefano Nativi

WCS 2.0 Interface Standard – NetCDF Encoding Format Extension

Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore 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, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Document type:	OpenGIS® Discussion Paper
Document subtype:	Extension
Document stage:	Draft proposed version 2.0
Document language:	English

This page is intentionally left blank.

Contents

Page

Figures

Page

Tables

Page

i. Preface

This Discussion Paper describes a draft standard: an extension of the "WCS 2.0 Interface Standard – Core" specification [OGC 09-110r3].

This extension is an evolution of the Discussion Paper "Web Coverage Service (WCS) 1.1.2 extension for CF-netCDF 3.0 encoding" [OGC 09-018r02].

This specification document refers to the following specifications:

- OGC "CF-netCDF extension specifications" [OGC 11-???] which extends the "NetCDF Core" specification [OGC 10-090].
- OGC "NetCDF Binary Encoding Extension Standard: NetCDF Classic and 64-bit Offset Format" [OGC 10-092].

Suggested additions, changes, and comments on this draft standard are welcome and encouraged. Such suggestions may be submitted by email message or by making suggested changes in an edited copy of this document.

ii. Document terms and definitions

This document uses the standard terms defined in Subclause 5.3 of [OGC 06-121r9], which is based on the ISO/IEC Directives, Part 2. Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

iii. Submitting organizations

The following organizations have submitted this Specification to the Open GeoSpatial Consortium, Inc.:

- The University Corporation for Atmospheric Research (UCAR)
- The Italian National Research Council (CNR)
- The PIN –University of Florence

iv. Document Contributor Contact Points

All questions regarding this document should be directed to the editors:

Name	Organization
Ben Domenico	UCAR/Unidata
Stefano Nativi	CNR and PIN –Univ. of Florence

v. Revision history

Date	Release	Editor	Primary clauses modified	Description
2010-08-09	0.1.0	Stefano Nativi	All	Created
2010-09-12	0.1.1	Ben Domenico	Comments	Updated
2011-02-19	1.0.0	Stefano Nativi	All –added Multi-pointCoverage data	Updated
2011-08-27	1.0.0	Stefano Nativi	CF ver. 1.6	Updated

vi. Changes to the OGC Abstract Specification

The OpenGIS® Abstract Specification does not require any changes to accommodate the technical contents of this document.

vii. Future work

Improvements in this document are desirable to support changes and additions to CF-netCDF encoding. However, it is important that WCS and CF-netCDF remain "loosely coupled" in the sense that each can change and evolve without having to rewrite the other each time.

In particular, this extension standard encoding profile is limited to multi-point, and regular and warped grids; however, irregular grids are important in the CF-netCDF community and work is underway to expand the CF-netCDF to encompass other coverages types. CF conventions version 1.6 introduced ragged arrays datasets. They can be encoded in WCS using multi-point geometries.

This specification is written for netCDF classic version (aka version 3), but the netCDF version 4 is now being released. Currently, the plan is to submit a separate extension standard for CF-netCDF version 4 as it becomes more heavily used in the community.

Foreword

This document is an extension of the Web Coverage Service (WCS) version 2.0 *core* interface [OGC 09-110r3] for a specific encoding format: the CF-netCDF encoding.

This specification deals with: multi-dimensional gridded data and multi-dimensional multi-point data.

This specification is based on the WCS CF-netCDF classic encoding extension introduced in the OGC Discussion Paper "Web Coverage Service (WCS) 1.1.2 extension for CF-netCDF 3.0 encoding" [OGC 09-018r02]. The extension specified in the discussion paper is still valid for WCS 1.1.2 implementations.

This extension refers to the OGC netCDF standard specifications: "CF-netCDF Data Model extension" specification [OGC 11-??] and "NetCDF Binary Encoding Extension Standard: NetCDF Classic and 64-bit Offset Format" [OGC 10-092].

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 Inc. 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.

Introduction

The Web Coverage Service (WCS) supports electronic retrieval of geospatial data as "coverages" – that is, digital geospatial information representing space and time varying phenomena.

A WCS provides access to coverage data in forms that are useful for client-side rendering, as input into scientific models, and for other clients.

Every implementation of a WCS shall adhere to the WCS 2.0 *Core* interface specification. In fact, this standard defines only basic requirements; indeed, a few extensions are required in order to completely specify a WCS for implementation.

The WCS *core* standard does not prescribe support for any particular coverage encoding format, leaving other standards to define extensions of the *core* for specific encodings. For the transfer of coverage-valued results from server to client, at least one coverage format encoding extension is required to be implemented by both.

This document specifies an extension of the WCS *core* standard for the CF-netCDF encoding format.

CF-netCDF encoding format is netCDF conforming to the Climate and Forecast (CF) conventions (CF-netCDF). It consists of either a binary, XML, or ASCII format which encodes the CF-netCDF data model.

This specification is based on the netCDF (network Common Data Form) ver. 3.0 file format (referred as *classic* file format) using the CF (Climate and Forecast) conventions ver. 1.6.

This specification deals with multi-dimensional gridded data and multi-dimensional multi-point data.

Explanations and best practices for users and implementers of OGC WCS 2.0 are provided in [1].

Explanation and best practices for users and implementers of OGC netCDF-CF are provided in [4].

Web Coverage Service (WCS) — CF-netCDF encoding extension

1 Scope

This document specifies the CF-netCDF encoding format for WCS 2.0.

This WCS *core* extension allows clients to evaluate, request and use data encoded in CF-netCDF format from a WCS server. In fact, this specification introduces a set of requirements which a WCS implementation must fulfil to transfer coverage-valued results from server to client using the CF-netCDF encoding format.

This standard specifies the CF-netCDF data model mapping onto the WCS data model. In addition, this standard specifies the possible binary and XML-based encoding formats returned by a coverage request expressed through the WCS interface.

This specification deals with multi-dimensional discrete coverage data and multi-dimensional multi-point data.

2 Conformance

Standardization target are WCS 2.0 implementations (currently: servers).

This document establishes a single requirements class, *CF-netCDF*, of http://www.opengis.net/spec/WCS_coverage-encoding_netcdf/req/CF-netCDF/ with a single pertaining conformance class, *CF-netCDF*, with URI http://www.opengis.net/spec/WCS_coverage-encoding_netcdf/conf/CF-netCDF. Requirements and conformance test URIs defined in this document are relative to http://www.opengis.net/spec/WCS_coverage-encoding_netcdf/.

Annex A lists the conformance tests which shall be exercised on any software artifact claiming to implement an OGC WCS 2.0 extension for CF-netCDF 3.0 (or *classic*) encoding.

3 Normative references

This *OGC Web Coverage Service (WCS) — CF-netCDF encoding extension* specification consists of the present document. The complete specification is identified by OGC URI http://www.opengis.net/spec/WCS_coverage-encoding_netcdf/, the document has OGC URI http://www.opengis.net/doc/WCS_coverage-encoding_netcdf/IS/CF-netCDF/.

The complete specification is available for download from http://www.opengeo-spatial.org/standards/WCS_coverage-encoding_netcdf .

The following normative documents contain provisions that, through reference in this text, constitute provisions of this specification. 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 10-090, *NetCDF Core version 1.0*

Conformance classes used:

- core

OGC 10-092, *NetCDF Binary Encoding Extension Standard: NetCDF Classic and 64-bit Offset Format version 1.0*

Conformance classes used:

- netCDF classic
- netCDF 64-bit

OGC 11-???, *CF-netCDF Data Model extension specification*, version 2.0.1

Conformance classes used:

- CF-netCDF core
- CF-netCDF Discrete Sampling
- CF-netCDF Mapping onto ISO 19123

OGC 07-036, *Geography Markup Language (GML) Encoding Standard*, version 3.2.1

Conformance classes used:

- GML writing

OGC 09-146r1, *OGC® GML Application Schema for Coverages*, version 1.0

Conformance classes used:

- gml-coverage

OGC 10-151, *OGC® Web Coverage Service 2.0 Interface Standard – Index Subsetting Extension*, version 1.0

Conformance classes used:

- index-subsetting

OGC 09-110r3, *OGC® WCS 2.0 Interface Standard -Core*, version 2.0.0

Conformance classes used:

- core

4 Terms and definitions

For the purposes of this document, the terms and definitions given in the above references apply. In addition, the following terms and definitions apply.

4.1 Coverage

Feature that acts as a function to return values from its range for any direct position within its spatiotemporal domain [OGC 07-111]

4.2 Data Model

A description of the organization of data in a manner that reflects an information structure [ISO/IEC 11179-1 Specification and standardization of data elements – Part 1: Framework]

NOTE: netCDF literature reports the following definition for data model: a way of thinking about scientific data by applying a data model theory. It is an abstraction that describes how datasets are represented and used. In computer terms, a data model can be thought of as equivalent to an abstract object model in Object Oriented Programming in that an abstract data model describes data objects and what methods can be used on them.

4.3 Feature

abstraction of real world phenomena [5].
NOTE A feature may occur as a type or an instance.

4.4 Grid

Network composed of two or more sets of curves in which the members of each set intersect the members of the other sets in a algorithmic way [5]

NOTE: the curves partition a space into grid cells.

4.5 Multi-point coverage

A discrete coverage which is characterized by a finite domain consisting of points. Generally, the domain is a set of irregularly distributed points.

A set of hydrographic soundings is an example of a discrete point coverage.

4.6 NetCDF

NetCDF is a standard for data on complex grids –curvilinear in XY; sigma and density-related in Z; climatological and artificial calendars in T; and heading towards "tile mosaics" and 5D forecast ensembles in the near future.

4.7 OPeNDAP

For purposes of this document, OPeNDAP is an approach that allows applications to access remote, time-aggregated collections of netCDF-CF files (virtual datasets – often terabyte sized) through the unaltered netCDF API –as if they were local netCDF files.

5 Conventions

5.1 Namespace prefix conventions

The following namespaces are used in this document. The prefix abbreviations used constitute conventions used here, but are **not** normative. The namespaces to which the prefixes refer are normative, however.

— Namespace mappings

Prefix	Namespace URI	Description
GML	http://www.opengis.net/gml/3.2	GML 3.2.1
GMLCOV	http://www.opengis.net/gmlcov/1.0	GML Application Schema for Coverages 1.0
ncml	http://unidata.ucar.edu/ncml	ncML schema
WCS	http://www.opengis.net/wcs/2.0	WCS 2.0
xsd	http://www.w3.org/2001/XMLSchema	XML Schema

5.2 CF-netCDF data model

This document does not contain inline a formal specification of the CF-netCDF data model. In fact, it is available as an OGC specification [OGC 11-???]. This specification formalizes the CF-netCDF data model mapping onto the ISO 19123 coverage geometry model.

For reader's convenience, the CF-netCDF data model is shown in Figure 1, Figure 2, and Figure 3. An example of CF-netCDF dataset is reported in the Annex B.

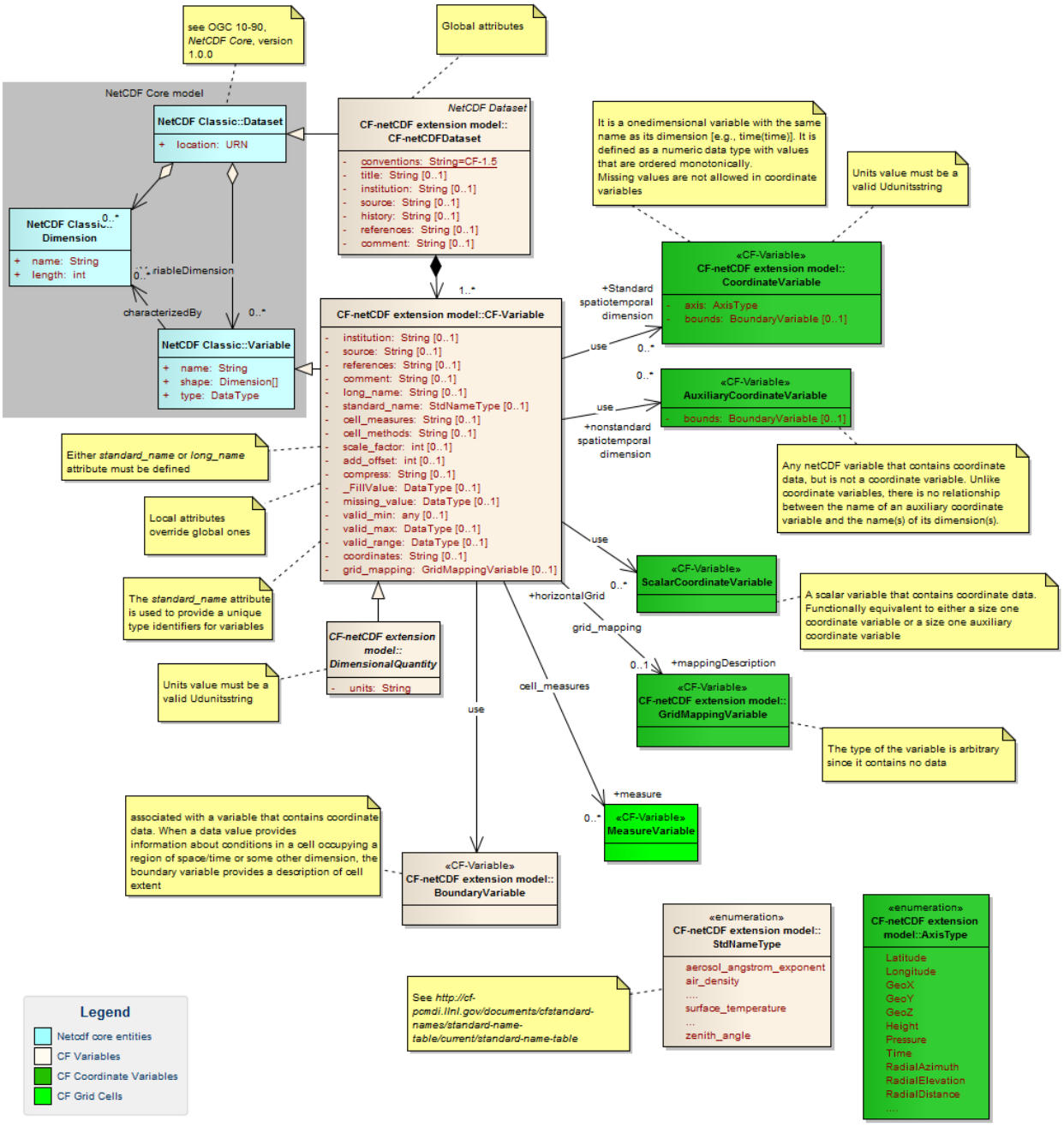

Figure 1 - CF-netCDF data model: Dataset and Variable elements

Figure 2 - CF-netCDF data model: Coordinate Variable and Coordinate System elements

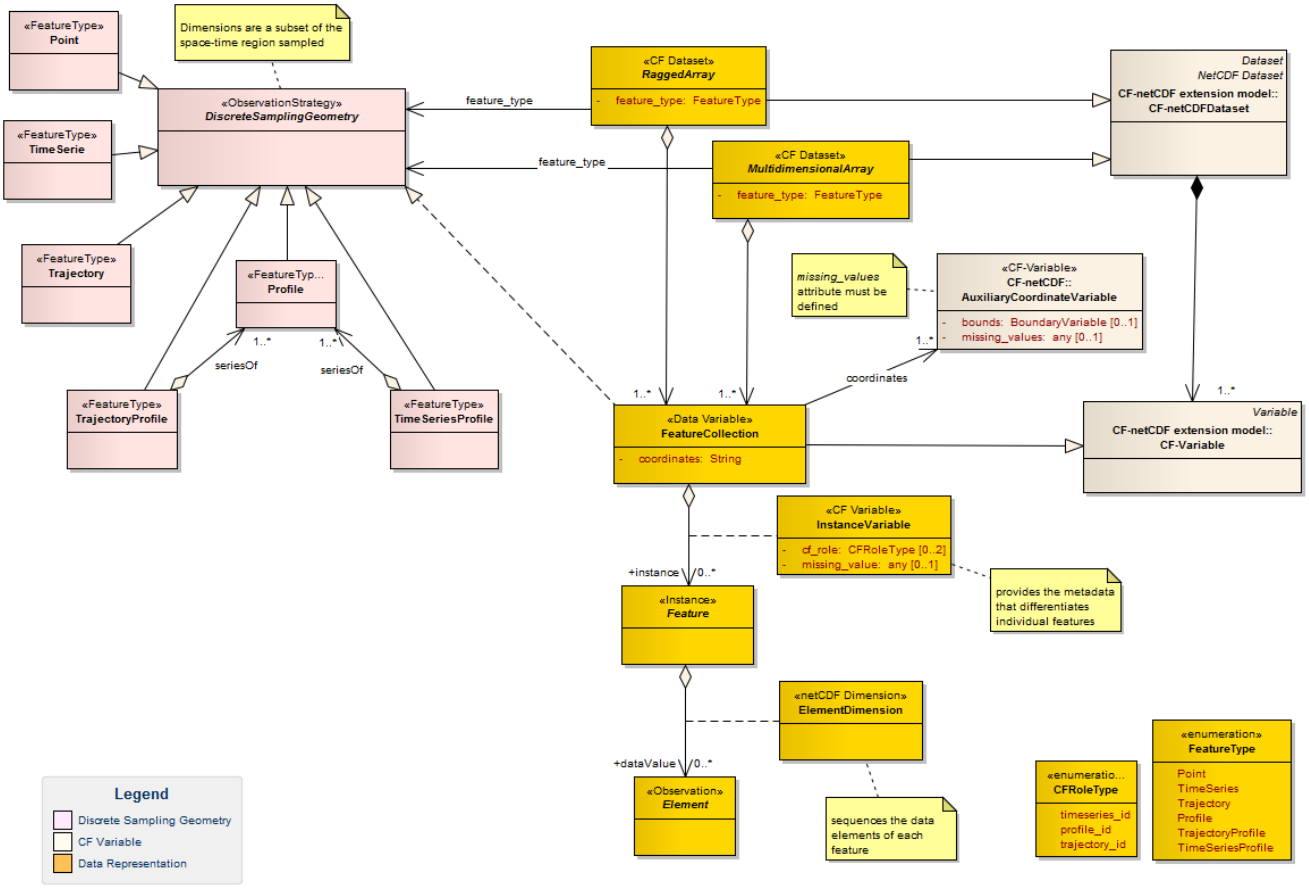

Figure 3 - CF-netCDF data model: Discrete Sampling Geometry elements

5.3 OPeNDAP

OPeNDAP (Open-source Project for a Network Data Access Protocol)[1] provides software which makes local data accessible to remote locations regardless of local storage format. OPeNDAP software is freely available.

NetCDF Community applications commonly make use of the OPeNDAP approach to access remote, time-aggregated collections of netCDF-CF files (virtual datasets – often terabyte sized) through the unaltered netCDF API –as if they were local netCDF files.

For the scope of this document, a simplified view of OPeNDAP is that it is a transparent mechanism by which an application can use netCDF API calls on a remote file. Thus, for any netCDF subset that may be derived from a WCS server, there may be an OPeNDAP URL that is an indirect reference to that same subset.

Essentially, the netCDF file is just a de-referencing of this URL. Any application program that can utilize a netCDF file can (in principal) utilize the URL equivalently.

[1] http://opendap.org/

NetCDF and CF Communities commonly use OPeNDAP to exchange netCDF datasets [12], [13]. The OPeNDAP support (and inclusion in this work) is important for the effectiveness of this specification.

6 CF-NetCDF dataset mapping to WCS 2.0 data model

WCS 2.0 data model introduces the concept of `WCS:OfferedCoverage` which is composed of: a coverage data model for WCS (i.e. `GMLCOV:Coverage`) and specific service metadata `WCS:ServiceParameters` [1] –see Figure 4.

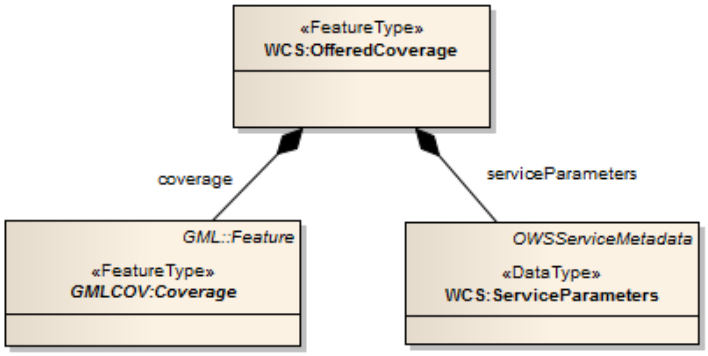

Figure 4. `OfferedCoverage` UML class diagram

6.1 WCS 2.0 coverage data model

`GMLCOV:Coverage` object is defined in [6]. For the reader's convenience the data model is depicted in Figure 5 and Figure 6.

The related `GML:DomainSet` and `GML:RangeSet` entities are defined in [7]; while, `SWE Common:DataRecord` is defined in [8].

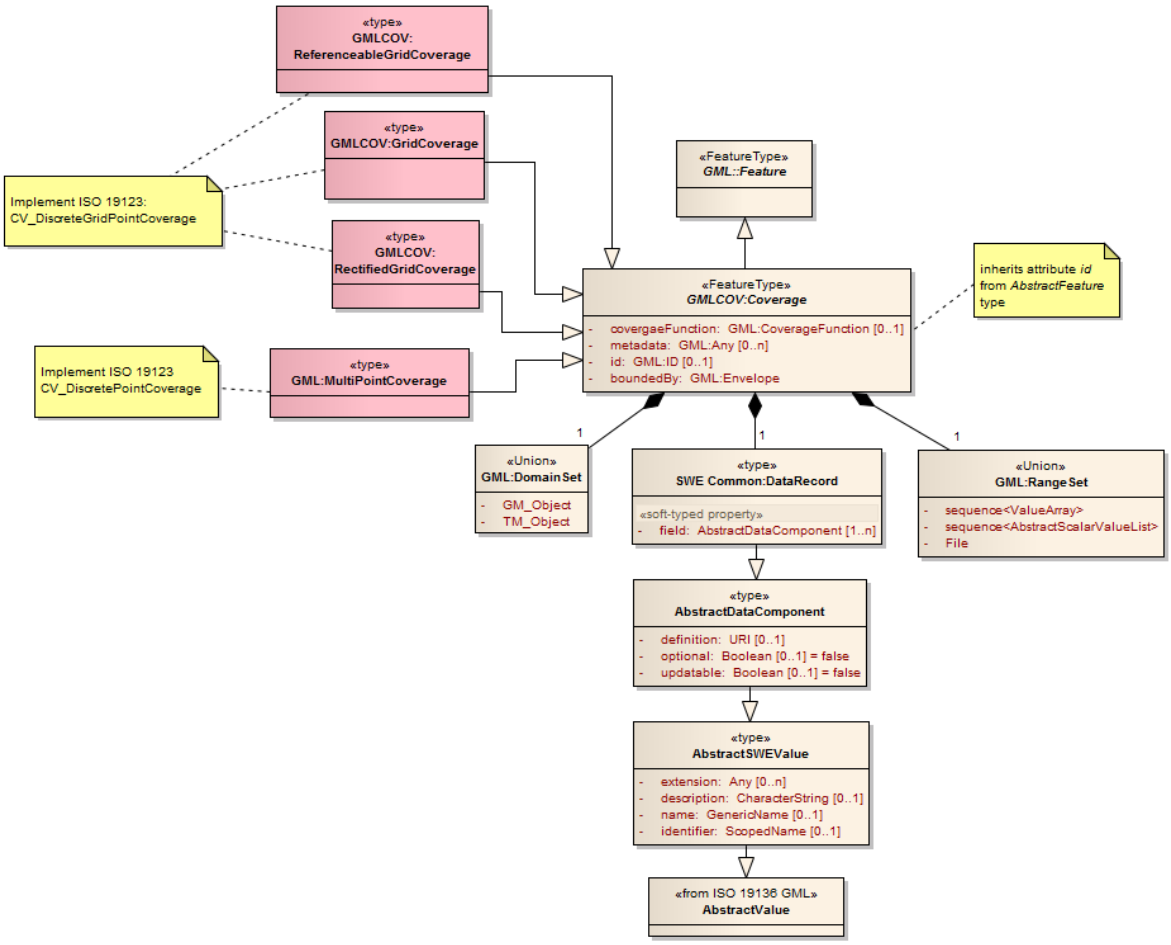

Figure 5. WCS 2.0 `Coverage` structure

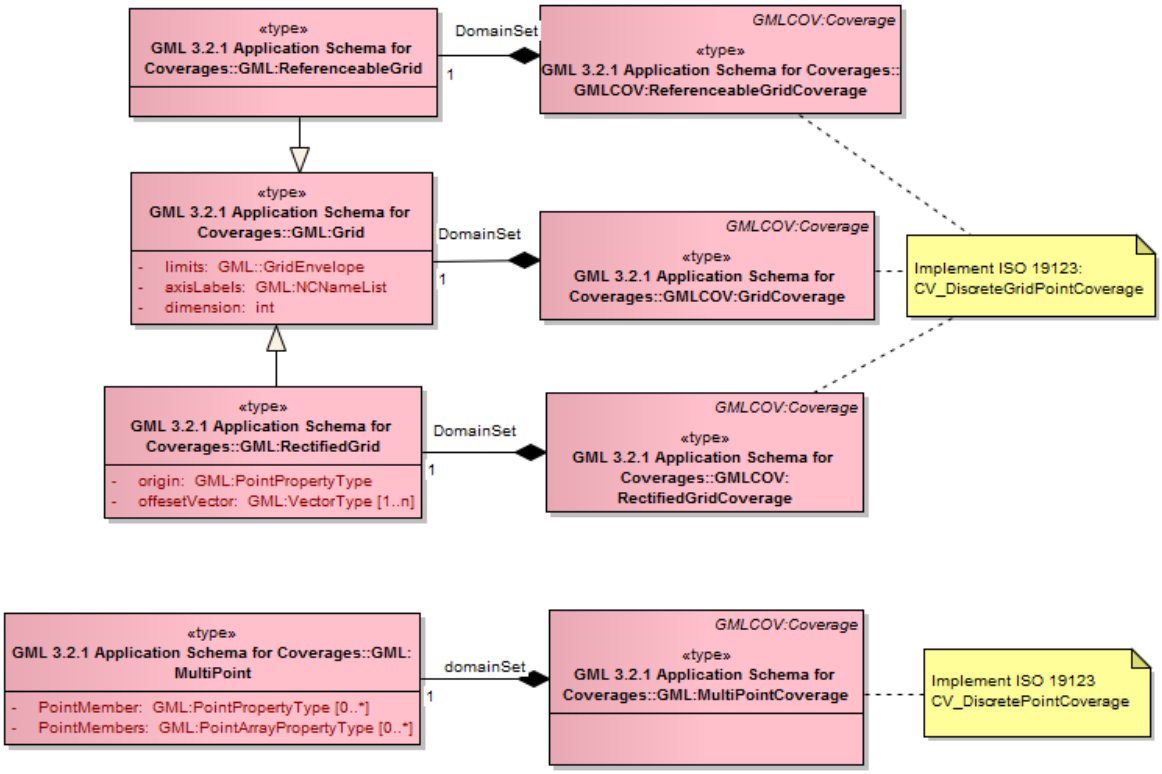

Figure 6. `GML:DomainSet` implementations

6.2 CF-netCDF dataset mapping to WCS 2.0 Coverage types

Since CF-netCDF dataset realizes the ISO19123:CV_DiscreteCoverage type [4], a WCS 2.0 server manages those datasets as either (see Figure 5): `GMLCOV:GridCoverage` type, `GMLCOV:RectifiedGridCoverage` type, `GMLCOV:ReferenceableGridCoverage` type, or `GMLCOV:MultiPointCoverage` –i.e. `ISO 19123:CV_DiscretePointCoverage`.

6.2.1 Mapping rules

Considering the mapping between the `GMLCOV:Coverage` object and property names to the corresponding class names and their attributes in ISO 19123 (see Table D.8 of [7] and the constraints introduced by [6] and [1]), and the mapping between the CF-netCDF dataset object and property names to the corresponding class names and their attributes in ISO 19123 (see Chapter 7 of OGC 11-??? [4]), Table 1 defines the mapping between the CF-netCDF dataset object and property names to the corresponding `GMLCOV:Coverage` object and property names. Figure 7 and Figure 8 depict these mapping rules.

Table 1. Mapping rules between CF-netCDF dataset and `GMLCOV:Coverage`

CF-netCDF entity	ISO Grid Coverage entity	Mapping Cardinality	GMLCOV:Coverage entity	Rule #
CF_Dataset and sub-types (e.g. *MultidimensionalArray* and *RaggedArray* entities)	*CV_DiscreteCoverage* *(CV_DiscreteGridPointCoverage* or *CV_DiscretePointCoverage)*	1 to 0..n	GML:Coverage implemented as: GMLCOV:GridCoverage| GMLCOV:RectifiedGridCoverage | GMLCOV:ReferenceableGridCoverage | GMLCOV:MultiPointCoverage	1
CF_Variable(s) and sub-types related to a *CoordinateSystem*	*CV_CoverageFunction*	1 to 0..1	coverageFunction, domainSet, rangeType, rangeSet implemented as, respectively: GML:CoverageFunction GML:Grid | GML:RectifiedGrid | GML:ReferenceableGrid | GML:MultiPoint SWE:DataRecord GML:RangeSet (see Table 2)	--
CoordinateSystem	*CS_CRS*	1 to 1	boundedBy implemented as GML:Envelope or the subtype GML:EnvelopeWithTimePeriod (GML:Envelope attributes: srsName and axisLabels must be instantiated) (see Table 3)	--
FeatureCollection	*CV_CoverageFunction*	1 to 0..1	coverageFunction, domainSet, rangeType, rangeSet implemented as, respectively: GML:CoverageFunction GML:Grid | GML:RectifiedGrid | GML:ReferenceableGrid | GML:MultiPoint SWE:DataRecord GML:RangeSet (see Table 4)	--

Requirement 1 /req/WCS2.0/CF-netCDF/OfferedCoverageTypes: CF-netCDF datasets, its subtypes and its realizations CF-netCDF RaggedArray and CF-netCDF

MultidimensionalArray **shall** realize one of the following `GMLCOV:Coverage` subtypes: `GMLCOV:GridCoverage`, `GMLCOV:RectifiedGridCoverage`, `GMLCOV:ReferenceableGridCoverage`, `GMLCOV:MultiPointCoverage`.

Additionally, the `GML:DomainSet` is implemented as one of the following subtypes: `GML:Grid`, `GML:RectifiedGrid`, `GML:ReferenceableGrid`, `GML:MultiPoint` (see Figure 6).

Table 2. Mapping rules between CF_Variable and `GMLCOV:Coverage` elements

CF-netCDF entity	ISO Coverage entity	Mapping Cardinality	`GMLCOV` entity	`GML 3.2.1` entity	rule #
CF_CoordinateVariable [a] (Valid subtypes are: LatitudeCoordinateLongitudeCoordinate, VerticalCoordinate, TimeCoordinate) or Spatial/ temporal *CF_AuxiliaryCoordinateVariable* [b]	`ISO19123:CV_DiscreteGridPointCoverage.CV_GridValueMatrix.CV_GridPoint.gridCoord`	1 to 1	`GMLCOV:GridCoverage.domainSet | GMLCOV:RectifiedGridCoverage.domainSet | GMLCOV:ReferenceableGridCoverage.domainSet`	`GML:DomainSet` realized as: `GML:Grid` or its subtypes: `GML:RectifiedGrid | GML:ReferenceableGrid`	2
	`ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.geometry.GM_Point`	1 to 1	`GMLCOV:MultiPointCoverage.domainSet`	`GML:DomainSet` realized as: `GML:MultiPoint`	2
	`ISO19123:CV_DiscreteGridPointCoverage.CV_GridValueMatrix.sequencingRule`	1 to 0..1	`GMLCOV:Coverage.coverageFunction`	`GML:CoverageFunction.GridFunction`	3
	`ISO19123:CV_DiscreteGridPointCoverage.CV_GridValueMatrix.startSequence`				
	`ISO19123:CV_DiscretePointCoverage.`*`CoverageFunction`* [e]	1 to 0..1	`GMLCOV:Coverage.coverageFunction`	`GML:CoverageFunction.CoverageMappingRule`	3
data *CF_Variable* [c] or non Spatial/temporal *CF_AuxiliaryCoordinateVariable* [e]	`ISO19123:CV_DiscreteGridPointCoverage.CV_GridValuesMatrix.values`	1 to 1	`GMLCOV: Coverage.rangeSet`	`GML:RangeSet`	4
	`ISO19123:CV_DiscretePointCoverage. CV_PointValuePair.value` [f]	1 to 1	`GMLCOV: Coverage.rangeSet`	`GML:RangeSet`	4
data *CF_Variable* [d] or non spatial/temporal *CF_AuxiliaryCoordinateVariable* [e]	`ISO19123:CV_DiscreteGridPointCoverage.RangeType.AttributesType entry (i.e. AttributeName, TypeName)`	1 to 1	`GMLCOV: Coverage.rangeType`	`SWE:DataRecord`	5
	`CV_DiscretePointCoverage.RangeType.AttributesType entry (i.e. AttributeName, TypeName)`	1 to 1	`GMLCOV: Coverage.rangeType`	`SWE:DataRecord`	5

[a] The set of spatial/temporal coordinate variables (shared by a group of CF_Variables) maps to the coverage domain, whose geometry is represented by a single grid (`ISO19123:CV_Grid`) or point value pairs (`ISO19123:CV_PointValuePair`).

[b] The set of spatial/temporal auxiliary coordinate variables (shared by a group of CF_Variables) maps to the coverage domain, whose geometry is represented by a single grid (`ISO19123:CV_Grid`) or point value pairs (`ISO19123:CV_PointValuePair`). The CF_Variables must define a `coordinates` attribute identifying the auxiliary coordinate variables. This is the case implemented by `CF_FeatureCollection` variables.

[c] The set of CF_Variables comprising the group that shares a common Coordinate System (i.e. set of spatial/temporal CF_CoordinateVariables or CF_AuxiliaryCoordinateVariables). The CF_Variable data values generate the grid value matrix record values: `CV_GridValueMatrix.values.record` entry (i.e. `AttributeName, Any`).
The *range* of each `CV_DiscreteGridPointCoverage` is a list of records with an attribute for every related CF-netCDF variable and for every CF_CoordinateVariable –of the shared CoordinateSystem- that is not allowed in a coverage CRS (i.e. parametric dimension axes). Thus, the CF_Variable properties (i.e. *name* and *type*) realize the `CV_DiscreteGridPointCoverage.RangeType.AttributesType` entry (i.e. `AttributeName, TypeName`).

[d] The *range* of each `CV_DiscreteGridPointCoverage` is a list of records with an attribute for every related CF-netCDF variable and for every CF_AuxiliaryCoordinateVariable that is not allowed in a coverage domain/CRS (i.e. parametric dimension axes).

[e] Parametric dimensions; examples are: *pressure*, *salinity*, *density*.
These coordinate variables do not behave like a discrete coverage *domain* axis; on the contrary, they behave like a discrete coverage *range* axis.

[f] In a `gml:MultiPointCoverage` the mapping from the domain to the range is straightforward.

- For `gml:DataBlock` encodings, the points of the `gml:MultiPoint` are mapped in document order to the tuples of the data block.
- For `gml:CompositeValue` encodings, the points of the `gml:MultiPoint` are mapped to the members of the composite value in document order.
- For `gml:File` encodings, the points of the `gml:MultiPoint` are mapped to the records of the file in sequential order.

CF_RaggedArray representations of a CF_Dataset should use such encodings

Requirement 2 /req/WCS2.0/CF-netCDF/OfferedCoverage.domainSet#1: The set of spatial/temporal CF_CoordinateVariable (comprised of a common coordinate system) that characterize a CF-netCDF dataset, **shall** realize a `GML:Coverage.domainSet` property, implemented as: `GML:MultiPoint`, `GML:Grid` object or its subtypes: `GML:RectifiedGrid`, `GML:ReferenceableGrid`.

Requirement 3 /req/WCS2.0/CF-netCDF/OfferedCoverage.domainSet#2: if the spatial/temporal domain is defined by auxiliary variables, the set of spatial/temporal CF_AuxiliaryCoordinateVariable that characterize a CF-netCDF Variable, **shall** realize a `GML:Coverage.domainSet` property, implemented as: `GML:MultiPoint`, `GML:Grid` object or its subtypes: `GML:RectifiedGrid`, `GML:ReferenceableGrid`.

Requirement 4 /req/WCS2.0/CF-netCDF/OfferedCoverage.coverageFunction#1: if the mapping from the domain to the range of the coverage is not characterized by the default values for `GML:Coverage` (see [7]), the set of spatial/temporal CF_CoordinateVariable (comprised of a common coordinate system) that characterizes a CF-netCDF dataset **shall** realize a `GML:Coverage.coverageFunction` property, implemented as either a

GML:CoverageFunction.GridFunction or a GML:CoverageFunction.CoverageMappingRule object.

Requirement 5 /req/WCS2.0/CF-netCDF/OfferedCoverage.coverageFunction#2: if the spatial/temporal domain is defined by auxiliary variables, the set of spatial/temporal CF_AuxiliaryCoordinateVariable that characterizes one or more CF-netCDF Variables **shall** realize a GML:Coverage.coverageFunction property, implemented as either a GML:CoverageFunction.GridFunction or a GML:CoverageFunction.CoverageMappingRule object.

Requirement 6 /req/WCS2.0/CF-netCDF/OfferedCoverage.rangeType#1: The set of CF-netCDF CF_Variable that share a common spatial/temporal coordinate system, **shall** realize a GML:Coverage.rangeType property, implemented as a SWE:DataRecord object.

Requirement 7 /req/WCS2.0/CF-netCDF/OfferedCoverage.rangeSet#1: The set of CF-netCDF CF_Variable that share a common spatial/temporal coordinate system, **shall** realize a GML:Coverage.rangeSet property, implemented as a GML:RangeSet object.

Requirement 8 /req/WCS2.0/CF-netCDF/OfferedCoverage.parametric-rangeType#2: the set of CF_AuxiliaryCoordinateVariable, which are not space or time dimension, **shall** contribute to realize a GML:Coverage.rangeType property, implemented as a SWE:DataRecord object.

Requirement 9 /req/WCS2.0/CF-netCDF/OfferedCoverage.parametric-rangeSet#2: The the set of CF_AuxiliaryCoordinateVariable, which are not space or time dimension, **shall** contribute to realize a GML:Coverage.rangeSet property, implemented as a GML:RangeSet object.

Table 3. Mapping rules between CF-netCDF CoordinateSystem and `GMLCOV:Coverage` elements

CF-netCDF entity	ISO Coverage entity	Mapping Cardinality	`GML 3.2.1` entity	Rule #
Spatial/temporal *Coordinate System* [a]	`CV_DiscreteGridPointCoverage.SC_CRS.SC_CoordinateSystem | CV_DiscretePointCoverage.SC_CRS.SC_CoordinateSystem`	1 to 1	`GML:Envelope` or the subtype `GML:EnvelopeWithTimePeriod`	6

CF_CoordinateVariable [b]	`CV_DiscreteGridPointCoverage.SC_CRS.SC_CoordinateSystem.SC_CoordinateSystemAxis`	1 to 1	`GML:Envelope.axisLabels`
			
			or
			
			`GML:EnvelopeWithTimePeriod.axisLabels`
				7

or spatial/temporal *CF_AuxiliaryCoordinateVariable* [d]	`CV_DiscretePointCoverage.SC_CRS.SC_CoordinateSystem.SC_CoordinateSystemAxis`	1 to 1		

GridMappingVariable	`CV_DiscreteGridPointCoverage.SC_CRS.SC_CoordinateSystem.type |`
	
	`CV_DiscretePointCoverage.SC_CRS.SC_CoordinateSystem.type`
		1 to 0..1	the subype of `GML_AbstractCRS` pointed by the `GML:Envelope.srsName` attribute[c]	8

[a] Parametric coordinate systems are allowed in CF-netCDF but not in CRS[2]. A coordinate system is of type parametric if a physical or material property is used as a dimension [21]; valuable examples are pressure in meteorology and density in oceanography. It is possible that a CoordinateSystem entity does not contain any axes allowed in coverage CRS (i.e. only parametric dimension axes).
Only spatial and temporal coordinates in a CF-netCDF CoordinateSystem become part of a coverage CRS, whereas parametric dimension axes are mapped to compound range set components.

[b] A CF-netCDF spatial/temporal CoordinateVariable shared by a group of CF-Variables. Valid subtypes are: LatitudeCoordinate LongitudeCoordinate, VerticalCoordinate, TimeCoordinate, CoordinateSystem.CoordinateAxis (with axis type attribute equal to: *Lat*, *Lon*, *Height*, *GeoX*, *GeoY*, *GeoZ*, *RadialAzimuth*, *RadialElevation*, *RadialDistance*, *time).*

[c] Wherever a GML object is associated with a Coordinate Reference System, this is implemented by an attribute (`srsName`) pointing to a `GML:AbstractCRS` element [7].

[d] The set of spatial/temporal auxiliary coordinate variables (shared by a group of CF_Variables) maps to the coverage domain, whose geometry is represented by a single grid (`ISO19123:CV_Grid`) or point value pairs (`ISO19123:CV_PointValuePair`).

Requirement 10/req/WCS2.0/CF-netCDF/OfferedCoverage.boundedBy: The spatial/temporal CoordinateSystem, which is comprised of either spatial and temporal

[2] Future extension to ISO 19111 (see ISO/CD19111-2) may permit parametric CRS, that would accommodate the pressure axis.

CF_CoordinateVariable objects or spatial and temporal CF_AuxiliaryCoordinateVariable objects, **shall** realize a `GML:Coverage.boundedBy` property, implemented as a `GML:Envelope` object or its subtype `GML:EnvelopeWithTimePeriod`.

Additionally, the attribute `GML:Envelope.srsName` **shall** have a value.

Requirement 11/req/WCS2.0/CF-netCDF/OfferedCoverage.boundedBy.axisLabels#1: The set of spatial/temporal CF_CoordinateVariable comprised of a CoordinateSystem object, **shall** realize a `GML:Envelope.axisLabels` object or its subtype `GML:EnvelopeWithTimePeriod.axisLabels`.

Requirement 12/req/WCS2.0/CF-netCDF/OfferedCoverage.boundedBy.axisLabels#2: The set of spatial/temporal CF_Auxiliary CoordinateVariable, comprised of a CoordinateSystem object related to CF_Variables, **shall** realize a `GML:Envelope.axisLabels` object or its subtype `GML:EnvelopeWithTimePeriod.axisLabels`.

Requirement 13/req/WCS2.0/CF-netCDF/OfferedCoverage.boundedBy.CRS: The CF-netCDF GridMappingVariable, **shall** determine the subype of `GML_AbstractCRS` pointed by the `GML:Envelope.srsName` attribute.

Table 4 - Mapping rules between CF-netCDF DiscreteSamplingGeometry and `GMLCOV:Coverage` elements

CF-netCDF entity	ISO Coverage entity	Mapp. Cardin	`GMLCOV` entity	`GML 3.2.1` entity	Rule #

Spatial/temporal *CF_InstanceVariable*[a]	`CV_DiscreteGridPointCoverage.SC_CRS.SC_CoordinateSystem.SC_CoordinateSystemAxis`
	or
	`CV_DiscretePointCoverage.SC_CRS.SC_CoordinateSystem.SC_CoordinateSystemAxis`
		1 to 1	--	`GML:Envelope.axisLabels`
				or
				`GML:EnvelopeWithTimePeriod.axisLabels`
					9

	`ISO19123:CV_DiscreteGridPointCoverage.CV_GridValueMatrix.CV_GridPoint.gridCoord`	1 to 1	`GMLCOV:GridCoverage.domainSet` | `GMLCOV:RectifiedGridCoverage.domainSet` | `GMLCOV:ReferenceableGridCoverage.domainSet`	`GML:DomainSet` realized as: `GML:Grid` or its subtypes: `GML:RectifiedGrid` | `GML:ReferenceableGrid`	10
	`ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.geometry.GM_Point`	1 to 1	`GMLCOV:MultiPointCoverage.domainSet`	`GML:DomainSet` realized as: `GML:MultiPoint`	10
	`ISO19123:CV_DiscreteGridPointCoverage.CV_GridValueMatrix.sequencingRule`	1 to 0..1	`GMLCOV:Coverage.coverageFunction`	`GML:CoverageFunction.GridFunction`	11
	`ISO19123:CV_DiscreteGridPointCoverage.CV_GridValueMatrix.startSequence`	1 to 0..1		`GML:CoverageFunction.CoverageMappingRule`	11
	`ISO19123:CV_DiscretePointCoverage.`*`CoverageFunction`*[e]				
CF_MultidimensionalArray.InstanceDimension[b]	`ISO19123:CV_DiscreteGridPointCoverage.CV_GridValuesMatrix.values`	1 to 0..1	`GMLCOV:Coverage.rangeSet`	`GML:RangeSet`	12
	`ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.`	1 to 0..1			

	value[f]				
	ISO19123:CV_DiscreteGridPointCoverage.RangeType.AttributesType entry (i.e. AttributeName, TypeName)	1 to 0..1	GMLCOV: Coverage.rangeType	SWE:DataRecord	13
	CV_DiscretePointCoverage.RangeType.AttributesType entry (i.e. AttributeName, TypeName)	1 to 0..1			
CF_RaggedArray.SampleDimension and *CF_IndexVariable*[c] or *CF_CountVariable*[c]	ISO19123:CV_DiscreteGridPointCoverage.CV_GridValuesMatrix.values	1 to 0..1	GMLCOV: Coverage.rangeSet	GML:RangeSet	14
	ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.value [f]	1 to 0..1			
	ISO19123:CV_DiscreteGridPointCoverage.RangeType.AttributesType entry (i.e. AttributeName, TypeName)	1 to 0..1	GMLCOV: Coverage.rangeType	SWE:DataRecord	15
	CV_DiscretePointCoverage.RangeType.AttributesType entry (i.e. AttributeName, TypeName)	1 to 0..1			
CF_Point FeatureCollection domain[d]	ISO19123:CV_DiscreteCoverage.domainExtent and CV_DiscreteCoverage.SC_CRS.SC_CoordinateSystem.SC_CoordinateSystemAxis	1 to 1	GMLCOV: Coverage.domainSet	GML:EnvelopeWithTimePeriod.axisLabels = "x y t" | "lat lon t" | "lon lat t"	16
CF_TimeSerie FeatureCollection domain[e]	ISO19123:CV_DiscreteCoverage.domainExtent	1 to 1	GMLCOV: Coverage.domainSet	GML:EnvelopeWithTimePeriod.axisLabels = "x y t" | "lat lon t" | "lon lat t"	16

CF_Trajectory FeatureCollection domain [f]	ISO19123:CV_DiscreteCoverage.domainExtent	1 to 1	GMLCOV:Coverage.domainSet	GML:EnvelopeWithTimePeriod.axisLabels = "x y t" | "lat lon t" | "lon lat t"	16
CF_Profile FeatureCollection domain [g]	ISO19123:CV_DiscreteCoverage.domainExtent	1 to 1	GMLCOV:Coverage.domainSet	GML:EnvelopeWithTimePeriod.axisLabels = "x y z t" | "lat lon z t" | "lon lat z t"	16
CF_TimeSeriesProfile FeatureCollection domain [h]	ISO19123:CV_DiscreteCoverage.domainExtent	1 to 1	GMLCOV:Coverage.domainSet	GML:EnvelopeWithTimePeriod.axisLabels = "x y z t" | "lat lon z t" | "lon lat z t"	16
CF_TrajectoryProfile FeatureCollection domain [i]	ISO19123:CV_DiscreteCoverage.domainExtent	1 to 1	GMLCOV:Coverage.domainSet	GML:EnvelopeWithTimePeriod.axisLabels = "x y z t" | "lat lon z t" | "lon lat z t"	16

[a] spatial/temporal CF_AuxiliaryCooridnateVariable which has as only dimension the "instance" dimension; it is used to differentiate the Feature instances comprised of a Feature Collection -e.g. X(station), lat(profile), etc.
This mapping was already formalized by Requirement 3, Requirement 5, and Requirement 12.

[b] CF_MultidimensionalArray instance dimension(s) may be used to generate an instance dimensionality for the coverage range-set.

[c] CF_RaggedArray sample dimension(s) and variables used to index the observation elements (i.e. either CF_IndexVariable or CF_CountVariable) may be used to generate an instance dimensionality for the coverage range-set.

[d] Point Feature Collection has the following mandatory space-time coordinates x(i) y(i) t(i); i=collection instance.

[e] TimeSerie Feature Collection has the following mandatory space-time coordinates x(i) y(i) t(i,o); i=collection instance; o=observation.

[f] Trajectory Feature Collection has the following mandatory space-time coordinates x(i,o) y(i,o) t(i,o); i=collection instance; o=observation.

[g] Profile Feature Collection has the following mandatory space-time coordinates x(i) y(i) z(i,o) t(i); i=collection instance; o=observation.

[h] TimeSerieProfile Feature Collection has the following mandatory space-time coordinates x(i) y(i) z(i,p,o) t(i,p); i=collection instance; p=profile; o=observation.

[i] TrajectoryProfile Feature Collection has the following mandatory space-time coordinates x(i,p) y(i,p) z(i,p,o) t(i,p); i=collection instance; p=profile; o=observation.

Requirement 14/req/WCS2.0/CF-netCDF/RangeInstanceDimension#1: each CF_MultidimensionalArray.InstanceDimension object **shall** contribute to realize a `GML:Coverage.rangeType` property, implemented as a `SWE:DataRecord` object.

Requirement 15/req/WCS2.0/CF-netCDF/ RangeInstanceDimension#2: each CF_MultidimensionalArray.InstanceDimension object **shall** contribute to realize a `GML:Coverage.rangeSet` property, implemented as a `GML:RangeSet` object.

Requirement 16/req/WCS2.0/CF-netCDF/RangeInstanceDimension#3: each CF_RaggedArray.SampleDimension object along with associated CF_IndexVariable or CF_CountVariable **shall** contribute to realize a `GML:Coverage.rangeType` property, implemented as a `SWE:DataRecord` object.

Requirement 17/req/WCS2.0/CF-netCDF/ RangeInstanceDimension#4: each CF_RaggedArray.SampleDimension object along with associated CF_IndexVariable or CF_CountVariable **shall** contribute to realize a `GML:Coverage.rangeSet` property, implemented as a `GML:RangeSet` object.

Requirement 18/req/WCS2.0/CF-netCDF/PointFeatureCollection.axisLabels: each CF_Point FeatureCollection **shall** realize a `GML:EnvelopeWithTimePeriod.axisLabels` with value equal to `"x y t" | "lat lon t" | "lon lat t"`.

Requirement 19/req/WCS2.0/CF-netCDF/TrajectoryFeatureCollection.axisLabels: each CF_Trajectory FeatureCollection **shall** realize a `GML:EnvelopeWithTimePeriod.axisLabels` with value equal to `"x y t" | "lat lon t" | "lon lat t"`.

Requirement 20/req/WCS2.0/CF-netCDF/ProfileFeatureCollection.axisLabels: each CF_Profile FeatureCollection **shall** realize a `GML:EnvelopeWithTimePeriod.axisLabels` with value equal to `"x y z t" | "lat lon z t" | "lon lat z t"`.

Requirement 21/req/WCS2.0/CF-netCDF/TimeSeriesProfileFeatureCollection.axisLabels: each CF_TimeSeriesProfile FeatureCollection **shall** realize a `GML:EnvelopeWithTimePeriod.axisLabels` with value equal to `"x y z t" | "lat lon z t" | "lon lat z t"`.

Requirement 22/req/WCS2.0/CF-netCDF/TrajectoryProfileFeatureCollection.axisLabels: each CF_TrajectoryProfile FeatureCollection **shall** realize a `GML:EnvelopeWithTimePeriod.axisLabels` with value equal to `"x y z t" | "lat lon z t" | "lon lat z t"`.

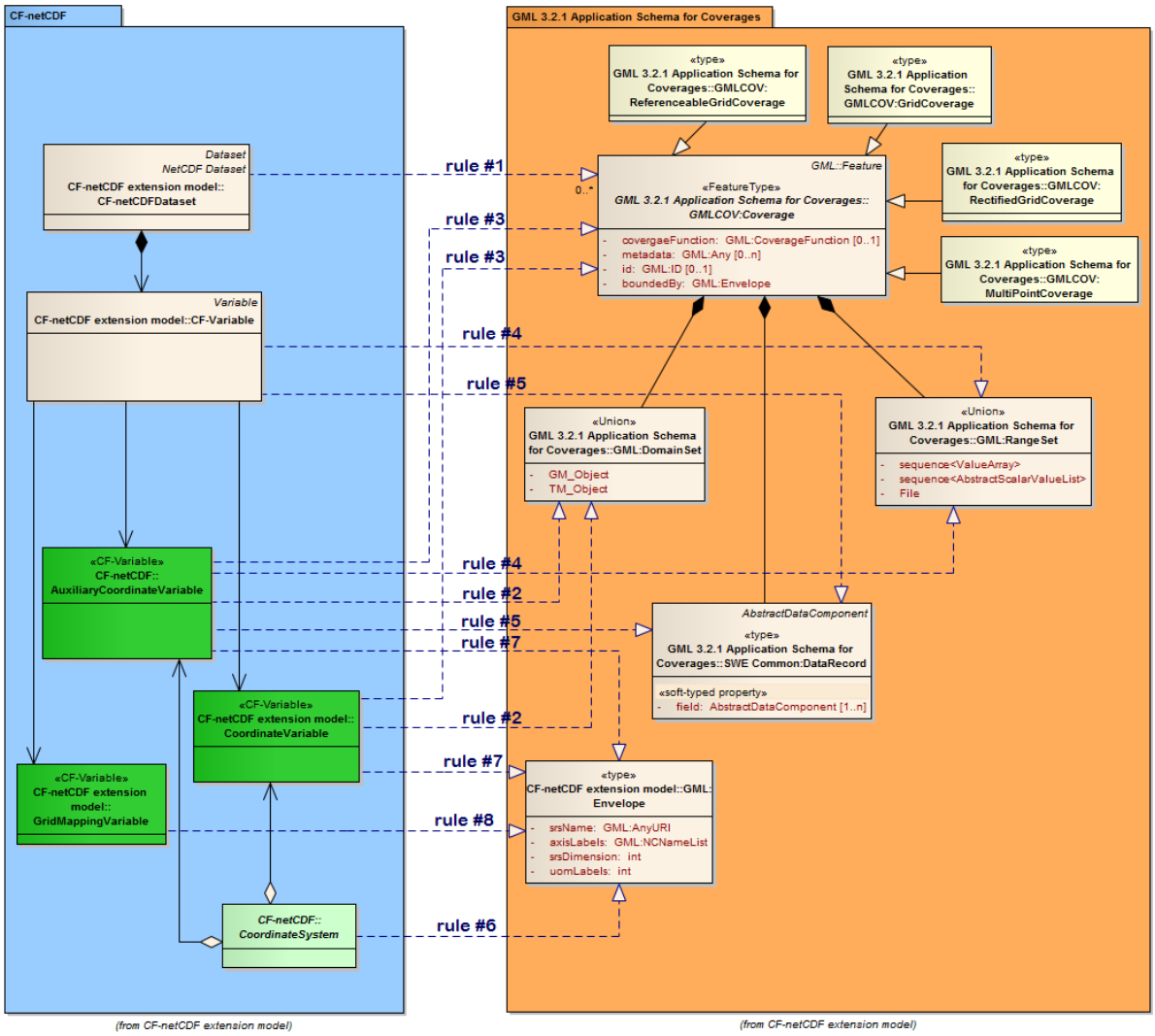

Figure 7. CF-netCDF dataset mapping to GMLCOV data model

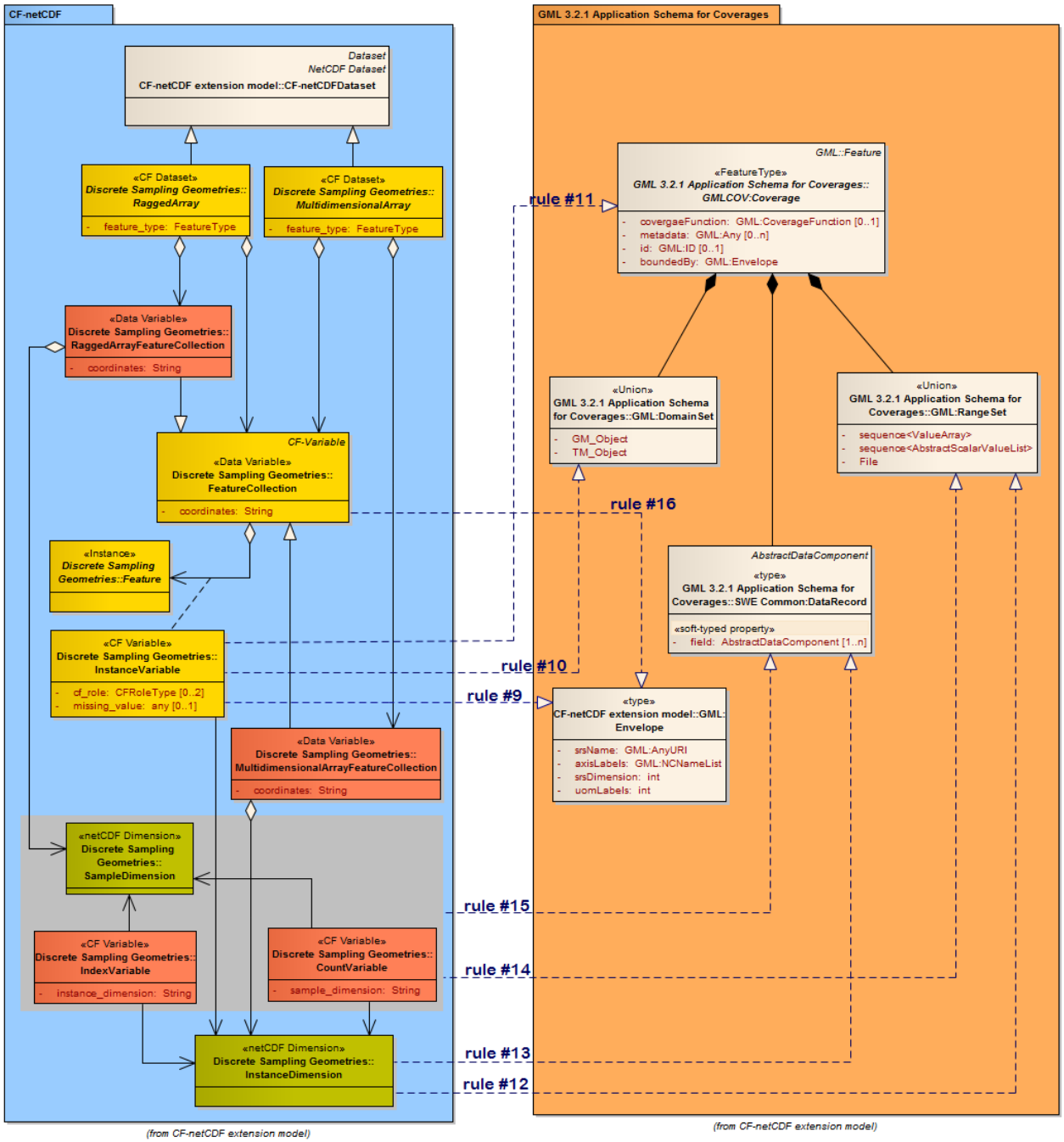

Figure 8. CF-netCDF discrete sampling geometries mapping to GMLCOV data model

6.3 WCS:ServiceParameters.coverageSubtype qualification

Requirement 23/req/WCS2.0/CF-netCDF/coverageSubtype-content: The `coverageSubtype` in the `ServiceParameters` of an `OfferedCoverage`, **shall** have one of the following values: “CF-netCDF:DiscreteGridPointCoverage”, “CF-netCDF:DiscreteGridPointCoverage.RectifiedGrid”, “CF-netCDF:DiscreteGridPointCoverage.ReferenceableGrid”, “CF-netCDF:DiscreteMultiPointCoverage” .

6.4 WCS:ServiceMetadata extension

Requirement 24/req/WCS2.0/CF-netCDF/coverageSubtype-content: the WCS implementation **shall** support the extension for indexed subsetting, which has URI identifier `http://www.opengis.net/spec/WCS_service-model_index-subsetting/1.0` .

7 WCS service model

7.1 GetCapabilities operation

Requirement 25/req/WCS2.0/CF-netCDF/extension-identifier:
A WCS service implementing this extension **shall** include the following URI in the `Profile` element of the `ServiceDescription` in a *GetCapabilities* response:
http://www.opengis.net/spec/WCS_coverage-encoding_netcdf/req/CF-netCDF

7.2 *DescribeCoverage* operation

7.2.1 *DescribeCoverage* response extension

The response to a successful DescribeCoverage request contains a list of coverage metadata, one for each coverage identifier passed in the request.

Requirement 26/req/WCS2.0/CF-netCDF/describeCoverage-response-extension:
The response to a successful *DescribeCoverage* request **shall** consist of a `CoverageDescriptions` as described in Figure 9.Figure 9, and Table 5.

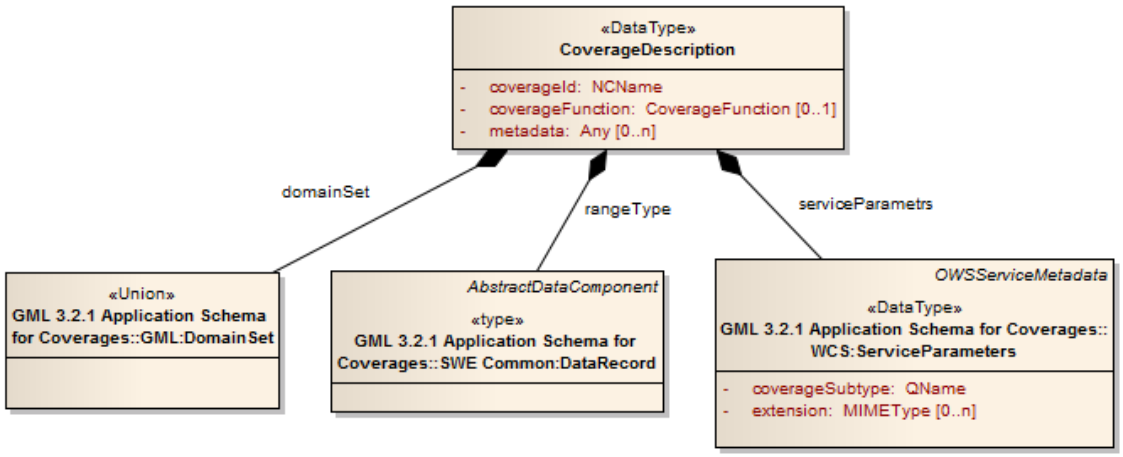

Figure 9. UML class diagram of `CoverageDescription` response extension

Table 5. `ServiceParameters` extended components

Name	Definition	Data type	Multiplicity
coverage-Subtype	Indicator of the type of coverage	QName[a]	one (mandatory)
extension	File encoding provided by the server for Coverage instances to be returned[b]	MIMEType	zero or more (optional)

a. see clause 6.3

b. see clause 8

Requirement 27/req/ WCS2.0/CF-netCDF/describeCoverage-response-extension-values:
If present, any `extension` parameter value of `ServiceParameter` object in a *DescribeCoverage* response **shall** be equal to one of the following MIME types: "*application/CF-netCDF*", "*application/ncML+xml*", "*application/gml+xml*", "*application/x-OPeNDAP-URL*".

Requirement 28/req/ WCS2.0/CF-netCDF/describeCoverage-response-extension-defaultValue:
If no `extension` parameter value of `ServiceParameter` object is present in a *DescribeCoverage* response, the WCS server **shall** be intended to return a CF-netCDF binary encoding, as specified by OGC 10-092 [9].

NOTE: Thus, the default coverage encoding format is the `netCDF binary` specification.

For each selected coverage, the rest of the *DescribeCoverage* response returns the information contained by the `WCS:OfferedCoverage` structure minus the coverage `rangeSet`. Thus, for the response see clause 6 and Table 1, Table 2, and Table 3.

7.3 *GetCoverage* operation

7.3.1 *GetCoverage* request extension

Requirement 29/req/ WCS2.0/CF-netCDF/getCoverage-request-extension:
A *GetCoverage* request **shall** consist of a structure as defined in Figure 10, and Table 6.

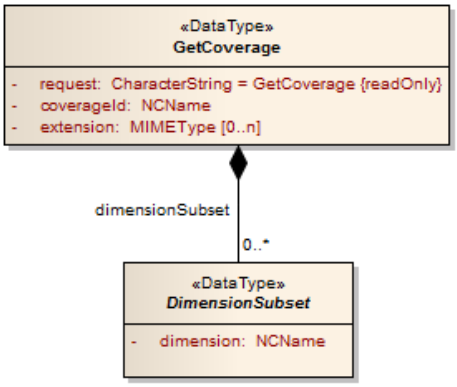

Figure 10. UML class diagram of `GetCoverage` request extension

Table 6. `GetCoverage` operation extended request

Name	Definition	Data type	Multiplicity
`service`	Service identifier	`String`, fixed to "WCS"	one (mandatory)
`version`	WCS service version indicator	`String`, fixed to a pattern of three dot-separated decimal digits	one (mandatory)
`extension`	File encoding requested for the Coverage instance to be returned[a]	`MIMEType`	zero or more (optional)
`coverageId`	Identifier of coverage evaluated	`NCName`	One (mandatory)
`dimension-Subset`	Subsetting specifications, one per subsetting dimension	`DimensionSubset`	Zero or more (optional)

[a]. see clause 8

Requirement 30 /req/ WCS2.0/CF-netCDF/getCoverage-request-extension-values:
If present, any `extension` parameter value in a *GetCoverage* request **shall** be equal to one of the following MIME type: "*application/CF-netCDF*", "*application/ncML+xml*", "*application/gml+xml*", "*application/x-OPeNDAP-URL*".

Requirement 31 /req/ WCS2.0/CF-netCDF/getCoverage-request-extension-defaultValue:
If no extension parameter value is present in a *GetCoverage* request, the WCS server **shall** be intended to return a CF-netCDF binary encoding, as specified by OGC 10-092 [9].

NOTE: Thus, the dafault coverage encoding format is the `CF-netCDF binary` specification.

7.3.2 GetCoverage response

The response to a successful *GetCoverage* request is a concrete sub-type of the `GMLCOV:Coverage` class, as showed in Figure 5.

NOTE: the GML encoding is solely used for the purpose of defining the response semantics. A server may deliver a result coverage in some other format –as defined by this WCS format extension. However, it is required that result coverages, regardless of what the encoding chosen is, have a contents which is consistent with the GML specification [1].

7.4 Coverage encoding formats

Requirement 32/req/WCS2.0/CF-netCDF/format-extension: for transmission of *GetCoverage* responses, a WCS service implementating this extension **shall** support the present encoding format extension, which has URI identifier http://www.opengis.net/spec/WCS_coverage-encoding_netcdf/req/CF-netCDF .

7.5 Extensions

This WCS implementation shall support the extension for indexed subsetting (see Requirement 24) as well as the mandatory WCS-core `DimensionSubset` cases [1]. Thus, clients can subset coverages in two independent ways, either by indicating coordinates in the CRS named in the `srsName` attribute of the coverage's `gml:Envelope` or by using grid coordinates as specified in the coverage's `gml:Grid`.

7.6 Protocol binding

The present extension does not require any specific binding. However, the following bindings are suggested:

HTTP/POST with XML (profile conformance URI: http://www.opengis.net/spec/WCS_protocol-binding_post-xml/1.0/).

HTTP/GET with KVP (profile conformance URI: http://www.opengis.net/spec/WCS_protocol-binding_get-kvp/1.0/).

SOAP with XML (profile conformance URI: http://www.opengis.net/spec/WCS_protocol-binding_soap/1.0/).

8 CF-netCDF data encodings returned by a successful *GetCoverage* response

The schema depicted in Figure 11 introduces the possible CF-netCDF data encodings used to return requested Coverages.

Requirement 33/req/ WCS2.0/CF-netCDF/coverageEncoding-structure:
The CF-netCDF data encoding structure used to return requested `GMLCOV:Coverages` **shall** adhere to Figure 11.

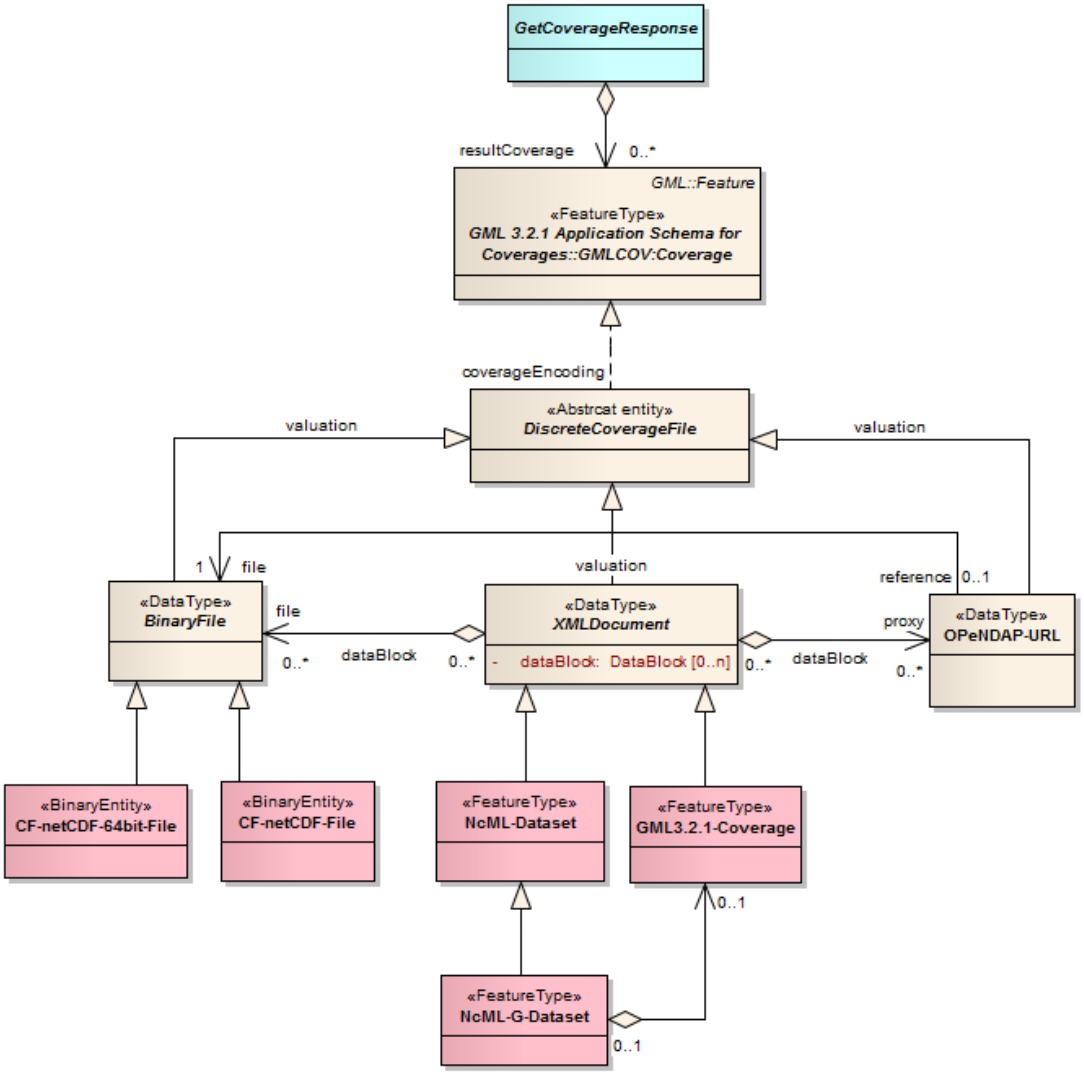

Figure 11. Supported CF-netCDF Coverage encodings

8.1 GridCoverageFile class

The response to a successful *GetCoverage* request is a concrete sub-type of `GMLCOV:Coverage` which is encoded as a *GridCoverageFile*, an abstract entity which has three valuation subclasses: *BinaryFile*, *XMLDocument*, and *OPenDAP-URL*.

NOTE: a CF-netCDF *BinaryFile* may be accessed directly or can be referenced by: (a) an *XMLDocument* object (e.g. an ncML dataset); (b) an *OPeNDAP-URL* object.

8.2 BinaryFile class

BinaryFile class has two concrete sub-types: *CF-netCDF-File* and *CF-netCDF-64bit-File*. They represent a netCDF ver. 3.0 binary file, complying with the CF ver. 1.1 conventions. These binary encodings are based on the "OGC NetCDF Binary Encoding Extension Standard: NetCDF Classic and 64-bit Offset Format" [9].

MIME type string used to refer to CF-netCDF binary encoding is:

application/CF-netCDF

Note: the only MIME type in use presently is "application/x-netcdf" and it is not officially registered. Thus the MIME types discussed below are new and application for registration will be made to IANA.

8.3 XMLDataset class

XMLDataset class has three concrete sub-types: *NcML-Dataset*, *GML-3.2.1-Coverage*, and *NcML-G-Dataset*.

8.3.1 NcML-Dataset

NcML-Dataset represents a ncML ver. 1.0 document [10]. A ncML document provides the XML encoding of one or more CF-netCDF files. For performances sake, a ncML document generally encodes only the CF-netCDF coverage metadata leaving the data values in the binary format and providing pointers (i.e. references) to those.

A ncML document may point to data values: (a) directly –i.e. referencing a CF-netCDF binary file; (b) including an OPeNDAP URL –which in turn reference a CF-netCDF file.

MIME type string used to refer to ncML-G dataset encoding is:

application/ncML+xml

This xml media type is still unregistered.

8.3.2 GML-3.2.1-Coverage

GML-3.2.1-Coverage represents the GML-based coverage encoding as specified by the WCS 2.0 Core standard [1]. Such a document is generated applying the requirements contained in clause 6 and summarized in Table 1, Table 2, and Table 3.

MIME type string used to refer to GML coverage encoding is [OGC 09-144r1]:

application/gml+xml

8.3.3 NCML-G-Dataset

NcML-G-Dataset represents a ncML-G ver. 1.0 document [10]. A ncML-G document extends a ncML-Dataset by including a GML Coverage element generated from the ncML dataset itself. In fact, this language was conceived to explicitly encode the two data models mapping. This may be very useful to fully support both GIS and netCDF applications.

MIME type string used to refer to ncML-G dataset encoding is:

application/ncML+xml

This xml media type is still unregistered.

8.4 OPeNDAP-URL class

This class realizes a well-adopted online reference to a CF-netCDF file. In fact, OPeNDAP [11] is a relatively simple approach (i.e. a Web protocol) that allows applications to access remote, time-aggregated collections of netCDF-CF files (virtual datasets –often terabyte sized) through the unaltered netCDF API –as if they were local netCDF files.

An OPeNDAP-URL instance may also be contained by any concrete *XMLDataset* object to reference a CF-netCDF binary file.

OPenDAP URL may be encoded using one of the following options: i) a specific MIME type for an OPeNDAP endpoint –analogous to *application/x-jpip-xml*; ii) an HTML document containing the OPeNDAP URL; iii) a simple text –semantically, this is a very dull solution, and is deprecated.

WCS server implementing this specification must use the following MIME type string to refer to OPeNDAP URL entity:

application/x-opendap-url

NOTE:

Result coverages, regardless of what the encoding chosen is, have a contents which is consistent with the `GMLCOV:Coverage` specification –see clause 6 and Table 1, Table 2, and Table 3.

Requirement 34/req/WCS2.0/CF-netCDF/coverages-encoding:
For a WCS service implementing this extension, the contents of the response to a successful *GetCoverage* request **shall** be encoded in one of the following coverage encoding formats: *CF-netCDF-File, CF-netCDF-64bit-File, ncML-Dataset, GML-3.2.1-Coverage, ncML-G-Dataset, OPeNDAP-URL*.

Requirement 35/req/WCS2.0/CF-netCDF/binaryFile-MIME-type:
WCS server implementing this specification **shall** use the following MIME type string to refer to CF-netCDF binary encoding: "*application/CF-netCDF*".

Requirement 36/req/WCS2.0/CF-netCDF/ncML-Dataset-MIME-type:
WCS server implementing this specification **shall** use the following MIME type string to refer to ncML-Dataset encoding: "*application/ncML+xml*".

Requirement 37/req/WCS2.0/CF-netCDF/OPenDAP-URL-reference:
WCS server implementing this specification **shall** use the following MIME type string to refer to OPeNDAP URL: "*application/x-opendap-url"*

Requirement 38/req/WCS2.0/CF-netCDF/OPeNDAP-URL-encoding:
In a GetCoverage successful response, a coverage returned as an OPenDAP URL **shall** be encoded as an HTML document containing the OPeNDAP URL.

8.5 *GetCoverage* response: *ncML-Dataset* document structure

The netCDF community makes use of ncML to encode complex netCDF data structures. In fact, it is possible to use ncML elements to define a virtual netCDF dataset which consists of one or more netCDF data files.

In a GetCoverage response returning a ncML document, the discrete coverage values are encoded and returned according to one or more of the following options:

(i) XML encoded values conforming to the ncML schema: this ncML section encodes the content of a CF-netCDF binary file.
(ii) CF-NetCDF binary files (i.e. *BinaryFile* objects) attached to an ncML document.
(iii) OPenDAP URL encoded in the ncML schema: actually, this ncML section references the content of a CF-netCDF binary file.

In any case, the coverage metadata section is encoded in the ncML document as elements conforming to the ncML schema.

For instance, Figure 12 depicts a complex *ncML-Dataset* (and optionally *ncML-G-Dataset*) structure transferred as a successful WCS *GetCoverage* response.

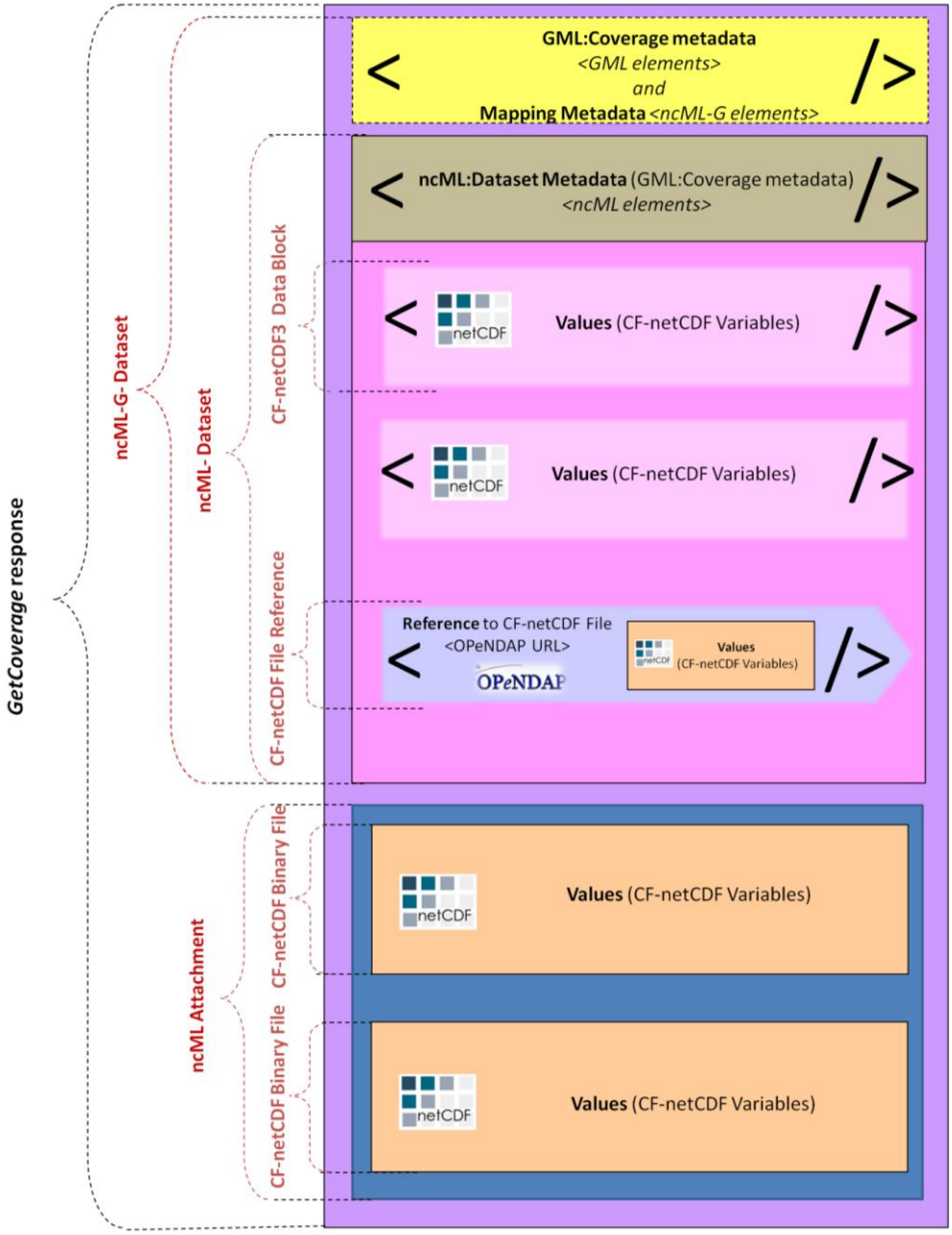

Figure 12. An example of *GetCoverage* response returning a complex *ncML-Dataset*

This type of encoding allows to return very complex data packages. As to the considered example, the returned coverage (i.e. dataset) values are encoded as: (a) two ncML *DataBlocks* elements (encoding values defined in a couple of CF-netCDF binary files); (b) an *OPeNDAP-URL* reference to the values defined in a third CF-netCDF file; (c) binary attachments –i.e. the data values defined in a fourth and fifth CF-netCDF binary file enclosed as ncML binary attachments.

For *ncML-Dataset*, coverage metadata are encoded as ncML elements. Besides, the *ncML-G-Dataset* extension adds a related grid coverage description using the GML elements (see Figure 12); thus, the metadata elements mapping is explicitly reported.

9 WCS GetCoverage response message: Multipart data encoding

Multipart data encoding is a technique for transferring multiple contents inside a single HTTP Response. In the context of WCS *GetCoverage* response it can be adopted whenever a binary content must be transferred.

Different strategies should be adopted for the different possible protocol bindings and encoding formats. Table 7 shows the four possibilities:

Table 7. *Multipart possible strategies*

Case	Request	Target	Possible Responses	Specification
#3	SOAP	ncML-Dataset	SOAP with ncML Attachment	SOAP Messages with Attachments 1.2 / MTOM-XOP
#4	HTTP (GET or POST)	ncML-Dataset	MIME Multipart with ncML parts	MIME multipart/related

The WCS *GetCoverage* response can be encoded in a Multipart message. The encoding strategy is different depending on the use of SOAP or HTTP binding.

9.1 Case #3 and #4: SOAP or HTTP Response returning ncML data

In case of ncML data, the response can refer to a local or remote document using the Manifest. Local resources are always attached to the message, while remote resources can be attached as a copy to avoid multiple access. The HTTP or SOAP response must be encoded along with its attachments in a MIME multipart/related message using the rules described in B.6.2 and B.6.1, respectively. At least one multipart section will contain XML data instead of binary data.

An ncML dataset include either coverage data values or a pointer to them –i.e. a pointer to one or more netCDF binary files.

Therefore, returned ncML documents may directly encode coverage data values or point to one or more netCDF binary files. Example B.6.3 shows a response returning an ncML document which points to a netCDF binary file, which is included in the MIME multipart/related message. Both data are remote resources.

Examples B.6.4.1 and B.6.4.2 describe responses returning ncML documents which encode coverage data values.

Example B.6.3: SOAP Response returning an ncML document referring a netCDF binary file included in the message.

```
MIME-Version: 1.0

Content-Type: multipart/related; boundary="----=_NextPart_000_0000_93251752.3C5526C0"


------=_NextPart_000_0000_93251752.3C5526C0

Content-Type: application/xml; charset="UTF-8"


<?xml version="1.0" encoding="UTF-8"?>

<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope">

  <soap:Header/>

  <soap:Body>

    <Coverages xmlns="http://www.opengis.net/wcs/1.1/ows" xmlns:ows="http://www.opengis.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:schemaLocation="http://www.opengis.net/wcs/1.1/ows ../owsCoverages.xsd
http://www.opengis.net/ows ../../../ows/1.0.0/ows19115subset.xsd">

      <Coverage>

        <ows:Title>Example</ows:Title>

        <ows:Abstract>Example of coverage</ows:Abstract>

        <Identifier>Example</Identifier>

        <Reference href="cid:bfa8e8ac@example.net" />

        <Reference href="http://example.net/data/CoverageDataValues.nc" />

        <Reference href="http://example.net/data/Dataset.ncml" />

      </Coverage>

    </Coverages>

  </soap:Body>

</soap:Envelope>

------=_NextPart_000_0000_93251752.3C5526C0

Content-Type: application/xml

Content-ID: <bfa8e8ac@example.net>


...(metadata content)...

------=_NextPart_000_0000_93251752.3C5526C0

Content-Type: application/CF-netCDF
```

Content-Transfer-Encoding: base64

Content-ID: <c3499c27@example.net>

Content-Location: http://example.net/data/CoverageDataValues.nc……

...(netCDF binary data content, pointed by the ncML dataset)...

------=_NextPart_000_0000_93251752.3C5526C0

Content-Type: application/ncML+xml

Content-Transfer-Encoding: base64

Content-ID: <1324bc12@example.net>

Content-Location: http://example.net/data/Dataset.ncml

…..

...(ncML dataset content)...

------=_NextPart_000_0000_93251752.3C5526C0--

NcML documents may contain coverage data values encoded as binary sections in base64 encoding. They can be left enclosed in the ncML document or, for efficiency purposes, they can be extracted and encoded in a format other than base64. Such improvement can be obtained serializing the ncML values element using nested multipart messages according to the MTOM/XOP specifications.

The following two encoding examples report multipart sections for ncML with binary data included and extracted using XOP, respectively.

9.1.1 Binary data included

Example 9.1.1: Response excerpt showing a ncML dataset containing coverage data values encoded as a binary section in base64 encoding.

multipart section containing a ncML dataset with binary data included (section outlined by a pale blue background)

------=_NextPart_000_0000_93251752.3C5526C0

Content-Type: application/CF-netCDF

Content-Transfer-Encoding: base64

Content-ID: <1324bc12@example.net>

Content-Location: http://example.net/data/data2.ncml

<?xml version="1.0" encoding="UTF-8"?>

<netcdf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns="http://www.unidata.ucar.edu/namespaces/netcdf/ncml-2.2">

```
<dimension name="longitude" length="96"/>
<dimension name="bounds_axis" length="2"/>
<dimension name="latitude" length="73"/>
<dimension name="air_pressure" length="15"/>
<dimension name="time" length="140"/>
<attribute name="Conventions" type="string" value="CF-1.1"/>
<attribute name="source" type="string" value="Data from model run ABC123"/>
<variable name="longitude" shape="longitude" type="double">
   <attribute name="standard_name" type="string" value="longitude"/>
   <attribute name="units" type="string" value="degrees_east"/>
   <attribute name="bounds" type="string" value="bound_longitude"/>
   <attribute name="axis" type="string" value="X"/>
   <values increment="3.75" npts="96" start="0.0"/>
</variable>
<variable name="bound_longitude" shape="longitude bounds_axis" type="double"> </variable>
<variable name="latitude" shape="latitude" type="double">
   <attribute name="standard_name" type="string" value="latitude"/>
   <attribute name="units" type="string" value="degrees_north"/>
   <attribute name="bounds" type="string" value="bound_latitude"/>
   <attribute name="axis" type="string" value="Y"/>
   <values increment="-2.5" npts="73" start="90.0"/>
</variable>
<variable name="bound_latitude" shape="latitude bounds_axis" type="double"> </variable>
<variable name="air_pressure" shape="air_pressure" type="float">
   <attribute name="standard_name" type="string" value="air_pressure"/>
   <attribute name="units" type="string" value="hPa"/>
   <attribute name="axis" type="string" value="Z"/>
   <attribute name="positive" type="string" value="down"/>
   <values increment="70.71429" npts="15" start="10.0"/>
</variable>
<variable name="time" shape="time" type="double">
   <attribute name="calendar" type="string" value="360_day"/>
```

```
      <attribute name="standard_name" type="string" value="time"/>
      <attribute name="units" type="string" value="days since 2289-1-1"/>
      <attribute name="bounds" type="string" value="bound_time"/>
      <attribute name="axis" type="string" value="T"/>
      <values increment="360.0" npts="140" start="510.0"/>
   </variable>
   <variable name="bound_time" shape="time bounds_axis" type="double"> </variable>
   <variable name="air_temperature" shape="time air_pressure latitude longitude" type="float">
      <attribute name="standard_name" type="string" value="air_temperature"/>
      <attribute name="units" type="string" value="Kir_temperature"/>
      <attribute name="long_name" type="string" value="temperature on pressure level"/>
      <attribute name="cell_methods" type="string" value="longitude: latitude: mean time: mean (interval:
4 h)"/>
      <attribute name="_FillValue" type="float" value="-1.073742e+09"/>
      <values xm:xmlmime="binary/octet-
stream">Li4uKGJhc2U2NCBlbmNvZGVkIGRhdGEpLi4u...(base64 encoded
data)...BlbmNvZGVkIGRhdGEpLi4u==</values>
   </variable>
</netcdf>
------=_NextPart_000_0000_93251752.3C5526C0--
```

9.1.2 Binary extracted using XOP

Example 9.1.2: HTTP multipart/related response showing a ncML dataset containing coverage data values extracted and encoded in a format other than base64 by serializing the <values> element using nested multipart messages according to the MTOM/XOP specifications.

multipart section containing a ncML dataset with binary data extracted using XOP (section outlined by a pale blue background)

```
------=_NextPart_000_0000_93251752.3C5526C0
MIME-Version: 1.0
Content-Type: Multipart/Related;boundary=MIME_boundary;  type="application/xop+xml";
   start="<12dea45c@example.net >";
   startinfo="application/ncML+xml";
```

```
--MIME_boundary
Content-Type: application/xop+xml; charset=UTF-8;
   type="application/ncML+xml; action=\"ProcessData\""
Content-ID: <12dea45c@example.net>
Content-Location: http://example.net/data/data2.ncml

<?xml version="1.0" encoding="UTF-8"?>
<netcdf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://www.unidata.ucar.edu/namespaces/netcdf/ncml-2.2">
   <dimension name="longitude" length="96"/>
   <dimension name="bounds_axis" length="2"/>
   <dimension name="latitude" length="73"/>
   <dimension name="air_pressure" length="15"/>
   <dimension name="time" length="140"/>
   <attribute name="Conventions" type="string" value="CF-1.1"/>
   <attribute name="source" type="string" value="Data from model run ABC123"/>
   <variable name="longitude" shape="longitude" type="double">
      <attribute name="standard_name" type="string" value="longitude"/>
      <attribute name="units" type="string" value="degrees_east"/>
      <attribute name="bounds" type="string" value="bound_longitude"/>
      <attribute name="axis" type="string" value="X"/>
      <values increment="3.75" npts="96" start="0.0"/>
   </variable>
   <variable name="bound_longitude" shape="longitude bounds_axis" type="double"> </variable>
   <variable name="latitude" shape="latitude" type="double">
      <attribute name="standard_name" type="string" value="latitude"/>
      <attribute name="units" type="string" value="degrees_north"/>
      <attribute name="bounds" type="string" value="bound_latitude"/>
      <attribute name="axis" type="string" value="Y"/>
      <values increment="-2.5" npts="73" start="90.0"/>
   </variable>
   <variable name="bound_latitude" shape="latitude bounds_axis" type="double"> </variable>
```

```
    <variable name="air_pressure" shape="air_pressure" type="float">
        <attribute name="standard_name" type="string" value="air_pressure"/>
        <attribute name="units" type="string" value="hPa"/>
        <attribute name="axis" type="string" value="Z"/>
        <attribute name="positive" type="string" value="down"/>
        <values increment="70.71429" npts="15" start="10.0"/>
    </variable>
    <variable name="time" shape="time" type="double">
        <attribute name="calendar" type="string" value="360_day"/>
        <attribute name="standard_name" type="string" value="time"/>
        <attribute name="units" type="string" value="days since 2289-1-1"/>
        <attribute name="bounds" type="string" value="bound_time"/>
        <attribute name="axis" type="string" value="T"/>
        <values increment="360.0" npts="140" start="510.0"/>
    </variable>
    <variable name="bound_time" shape="time bounds_axis" type="double"> </variable>
    <variable name="air_temperature" shape="time air_pressure latitude longitude" type="float">
        <attribute name="standard_name" type="string" value="air_temperature"/>
        <attribute name="units" type="string" value="Kir_temperature"/>
        <attribute name="long_name" type="string" value="temperature on pressure level"/>
        <attribute name="cell_methods" type="string" value="longitude: latitude: mean time: mean (interval:
4 h)"/>
        <attribute name="_FillValue" type="float" value="-1.073742e+09"/>
        <values xm:xmlmime="binary/octet-stream"><xop:Include
    xmlns:xop='http://www.w3.org/2004/08/xop/include'
    href='cid:bd43gh2y@example.net'/>
        </values>
    </variable>
</netcdf>
--MIME_boundary
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <bd43gh2y@example.net>
```

```
….
...(binary encoded data)...
--MIME_boundary--
------=_NextPart_000_0000_93251752.3C5526C0--
```

9.2 Content-ID generation

According to [RFC 2045] "Content-ID values must be generated to be world-unique". In WCS responses it is suggested that Content-ID has the following format:

Content-ID : = Local "@" Domain

where Local is a locally unique identifier and Domain is an Internet domain administered by the coverages provider. The locally unique identifier is obscure. This means that no semantic is required to be associated to the local name. For example it could be generated with a hashing function from coverages metadata (such as in the examples above).

Req ???

Actually, ncML (that is the netCDF XML encoding) seems to be particularly suitable for being dispatched with MTOM/XOP.

10 Bibliography

[1] OGC 09-153, *WCS 2.0 Overview: Core and Extensions*, version 1.0.0

[2] OGC 10-091, CF-netCDF Primer: CF-netCDF Core and Extensions, version 1.0

[3] OGC 10-90, *NetCDF Core*, version 1.0.0

[4] OGC 10-???, *CF-netCDF Data Model extension specification*, version 2.0.1, August 2011.

[5] ISO/TC 211, *IS 19123:2005 Geographic information — Schema for coverage geometry and functions*, ISO/IS 19123:2005

[6] OGC 09-146r1, *OGC® GML Application Schema – Coverages,* version 1.0.0.

[7] OGC 07-036, *OGC® Geography Markup Language (GML) Encoding Standard*, version 3.2.1.

[8] OGC 08-094, *OGC® SWE Common Data Model Encoding Standard*, version 2.0.0.

[9] OGC 10-092, *NetCDF Binary Encoding Extension Standard: NetCDF Classic and 64-bit Offset Format*, version 1.0.

[10] S. Nativi, J. Caron, E. Davis and B. Domenico, "Design and implementation of netCDF Markup Language (NcML) and Its GML-based extension (NcML-GML)", Computers & Geosciences Journal, Volume 31, Issue 9, November 2005, Pages 1104-1118, Elsevier Publication.
http://linkinghub.elsevier.com/retrieve/pii/S0098300405001019
http://www.unidata.ucar.edu/software/netcdf/ncml/v2.2/AnnotatedSchema.html

[11] NASA Community Standard, ESDS-RFC-004: *The Data Access Protocol – DAP* 2.0 available at: http://www.esdswg.org/spg/rfc/ese-rfc-004

[12] NASA ESDS–RFC-021 Technical Working Group Final Report.

[13] NASA ESDS-RFC-021v0.02, CF Metadata Conventions, April 2010.

Annex A
(normative)

Abstract test suite

A WCS CF-netCDF extension implementation must satisfy the following system characteristics to be conformant with this specification.

A.1 Conformance Test Class: CF-netCDF

The OGC URI identifier of this conformance class is: http://www.opengis.net/spec/WCS_coverage-encoding_netcdf/conf/CF-netCDF.

Tests identifiers below are relative to http://www.opengis.net/spec/WCS_coverage-encoding_netcdf/.

A.2 CF-netCDF Dataset structure

Test id: **/conf/CF-netCDF/structural-adherence**

Test Purpose: **Requirement Errore. L'origine riferimento non è stata trovata.**

Test method: TBD

Annex B
(Informative)
Examples

The following paragraphs collect the examples used and explained throughout the document.

B.1. Example: netCDF ver. 3 with CF1.1 convention dataset

```
netcdf air_temperature {

dimensions:

        longitude = 96 ;

        bounds_axis = 2 ;

        latitude = 73 ;

        air_pressure = 15 ;

        time = 140 ;

variables:

        double longitude(longitude) ;

                longitude:standard_name = "longitude" ;

                longitude:units = "degrees_east" ;

                longitude:bounds = "bound_longitude" ;

                longitude:axis = "X" ;

                double bound_longitude(longitude, bounds_axis) ;

        double latitude(latitude) ;

                latitude:standard_name = "latitude" ;

                latitude:units = "degrees_north" ;

                latitude:bounds = "bound_latitude" ;

                latitude:axis = "Y" ;

                double bound_latitude(latitude, bounds_axis) ;

        float air_pressure(air_pressure) ;

                air_pressure:standard_name = "air_pressure" ;

                air_pressure:units = "hPa" ;

                air_pressure:axis = "Z" ;

                air_pressure:positive = "down" ;

        double time(time) ;

                time:calendar = "360_day" ;
```

```
            time:standard_name = "time" ;
            time:units = "days since 2289-1-1" ;
            time:bounds = "bound_time" ;
            time:axis = "T" ;
            double bound_time(time, bounds_axis) ;
      float air_temperature(time, air_pressure, latitude, longitude) ;
            air_temperature:standard_name = "air_temperature" ;
            air_temperature:units = "K" ;
            air_temperature:long_name = "temperature on pressure levels" ;
            air_temperature:cell_methods = "longitude: latitude: mean time: mean (interval: 4 h)" ;
            air_temperature:_FillValue = -1.073742e+09f ;

// global attributes:
            :Conventions = "CF-1.1" ;
            :source = "Data from model run ABC123" ;
data:

 longitude = 0, 3.75, 7.5, 11.25, 15, 18.75, 22.5, 26.25, 30, 33.75, 37.5,
    41.25, 45, 48.75, 52.5, 56.25, 60, 63.75, 67.5, 71.25, 75, 78.75, 82.5,
    86.25, 90, 93.75, 97.5, 101.25, 105, 108.75, 112.5, 116.25, 120, 123.75,
    127.5, 131.25, 135, 138.75, 142.5, 146.25, 150, 153.75, 157.5, 161.25,
    165, 168.75, 172.5, 176.25, 180, 183.75, 187.5, 191.25, 195, 198.75,
    202.5, 206.25, 210, 213.75, 217.5, 221.25, 225, 228.75, 232.5, 236.25,
    240, 243.75, 247.5, 251.25, 255, 258.75, 262.5, 266.25, 270, 273.75,
    277.5, 281.25, 285, 288.75, 292.5, 296.25, 300, 303.75, 307.5, 311.25,
    315, 318.75, 322.5, 326.25, 330, 333.75, 337.5, 341.25, 345, 348.75,
    352.5, 356.25 ;
```

B.2. Example for an ncML dataset

```
<?xml version="1.0" encoding="UTF-8"?>
<netcdf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://www.unidata.ucar.edu/namespaces/netcdf/ncml-2.2">
   <dimension name="longitude" length="96"/>
   <dimension name="bounds_axis" length="2"/>
```

```
   <dimension name="latitude" length="73"/>
   <dimension name="air_pressure" length="15"/>
   <dimension name="time" length="140"/>
   <attribute name="Conventions" type="string" value="CF-1.1"/>
   <attribute name="source" type="string" value="Data from model run ABC123"/>
   <variable name="longitude" shape="longitude" type="double">
      <attribute name="standard_name" type="string" value="longitude"/>
      <attribute name="units" type="string" value="degrees_east"/>
      <attribute name="bounds" type="string" value="bound_longitude"/>
      <attribute name="axis" type="string" value="X"/>
      <values increment="3.75" npts="96" start="0.0"/>
   </variable>
   <variable name="bound_longitude" shape="longitude bounds_axis" type="double"> </variable>
   <variable name="latitude" shape="latitude" type="double">
      <attribute name="standard_name" type="string" value="latitude"/>
      <attribute name="units" type="string" value="degrees_north"/>
      <attribute name="bounds" type="string" value="bound_latitude"/>
      <attribute name="axis" type="string" value="Y"/>
      <values increment="-2.5" npts="73" start="90.0"/>
   </variable>
   <variable name="bound_latitude" shape="latitude bounds_axis" type="double"> </variable>
   <variable name="air_pressure" shape="air_pressure" type="float">
      <attribute name="standard_name" type="string" value="air_pressure"/>
      <attribute name="units" type="string" value="hPa"/>
      <attribute name="axis" type="string" value="Z"/>
      <attribute name="positive" type="string" value="down"/>
      <values increment="70.71429" npts="15" start="10.0"/>
   </variable>
   <variable name="time" shape="time" type="double">
      <attribute name="calendar" type="string" value="360_day"/>
      <attribute name="standard_name" type="string" value="time"/>
      <attribute name="units" type="string" value="days since 2289-1-1"/>
      <attribute name="bounds" type="string" value="bound_time"/>
      <attribute name="axis" type="string" value="T"/>
      <values increment="360.0" npts="140" start="510.0"/>
   </variable>
   <variable name="bound_time" shape="time bounds_axis" type="double"> </variable>
   <variable name="air_temperature" shape="time air_pressure latitude longitude" type="float">
      <attribute name="standard_name" type="string" value="air_temperature"/>
      <attribute name="units" type="string" value="Kir_temperature"/>
      <attribute name="long_name" type="string" value="temperature on pressure level"/>
      <attribute name="cell_methods" type="string" value="longitude: latitude: mean time: mean (interval:
4 h)"/>
      <attribute name="_FillValue" type="float" value="-1.073742e+09"/>
   </variable>
</netcdf>
```

B.3. GetCoverage response encoding examples

This paragraph detail examples of GetCoverage response encodings based on SOAP and Multipart-related technology. -

B.3.1. SOAP Response returning an ncML document referring a netCDF binary file included in the message

```
MIME-Version: 1.0

Content-Type: multipart/related; boundary="----=_NextPart_000_0000_93251752.3C5526C0"


------=_NextPart_000_0000_93251752.3C5526C0

Content-Type: application/xml; charset="UTF-8"


<?xml version="1.0" encoding="UTF-8"?>

<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope">

  <soap:Header/>

  <soap:Body>

    <Coverages xmlns="http://www.opengis.net/wcs/1.1/ows" xmlns:ows="http://www.opengis.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:schemaLocation="http://www.opengis.net/wcs/1.1/ows ../owsCoverages.xsd
http://www.opengis.net/ows ../../../ows/1.0.0/ows19115subset.xsd">

      <Coverage>

        <ows:Title>Example</ows:Title>

        <ows:Abstract>Example of coverage</ows:Abstract>

        <Identifier>Example</Identifier>

        <Reference href="cid:bfa8e8ac@example.net" />

        <Reference href="http://example.net/data/CoverageDataValues.nc" />

        <Reference href="http://example.net/data/Dataset.ncml" />

      </Coverage>

    </Coverages>

  </soap:Body>

</soap:Envelope>

------=_NextPart_000_0000_93251752.3C5526C0

Content-Type: application/xml

Content-ID: <bfa8e8ac@example.net>


...(metadata content)...

------=_NextPart_000_0000_93251752.3C5526C0

Content-Type: application/CF-netCDF
```

```
Content-Transfer-Encoding: base64
Content-ID: <c3499c27@example.net>
Content-Location: http://example.net/data/CoverageDataValues.nc……
...(netCDF binary data content, pointed by the ncML dataset)...
------=_NextPart_000_0000_93251752.3C5526C0
Content-Type: application/ ncML+xml
Content-Transfer-Encoding: base64
Content-ID: <1324bc12@example.net>
Content-Location: http://example.net/data/Dataset.ncml
…..
...( ncML dataset content)...
------=_NextPart_000_0000_93251752.3C5526C0--
```

B.3.2. **C.3.4 Multipart section containing ncML with binary data included**

```
------=_NextPart_000_0000_93251752.3C5526C0
Content-Type: application/CF-netCDF
Content-Transfer-Encoding: base64
Content-ID: <1324bc12@example.net>
Content-Location: http://example.net/data/data2.ncml

<?xml version="1.0" encoding="UTF-8"?>
<netcdf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://www.unidata.ucar.edu/namespaces/netcdf/ncml-2.2">
   <dimension name="longitude" length="96"/>
   <dimension name="bounds_axis" length="2"/>
   <dimension name="latitude" length="73"/>
   <dimension name="air_pressure" length="15"/>
   <dimension name="time" length="140"/>
   <attribute name="Conventions" type="string" value="CF-1.1"/>
   <attribute name="source" type="string" value="Data from model run ABC123"/>
   <variable name="longitude" shape="longitude" type="double">
```

```
    <attribute name="standard_name" type="string" value="longitude"/>
    <attribute name="units" type="string" value="degrees_east"/>
    <attribute name="bounds" type="string" value="bound_longitude"/>
    <attribute name="axis" type="string" value="X"/>
    <values increment="3.75" npts="96" start="0.0"/>
</variable>
<variable name="bound_longitude" shape="longitude bounds_axis" type="double"> </variable>
<variable name="latitude" shape="latitude" type="double">
    <attribute name="standard_name" type="string" value="latitude"/>
    <attribute name="units" type="string" value="degrees_north"/>
    <attribute name="bounds" type="string" value="bound_latitude"/>
    <attribute name="axis" type="string" value="Y"/>
    <values increment="-2.5" npts="73" start="90.0"/>
</variable>
<variable name="bound_latitude" shape="latitude bounds_axis" type="double"> </variable>
<variable name="air_pressure" shape="air_pressure" type="float">
    <attribute name="standard_name" type="string" value="air_pressure"/>
    <attribute name="units" type="string" value="hPa"/>
    <attribute name="axis" type="string" value="Z"/>
    <attribute name="positive" type="string" value="down"/>
    <values increment="70.71429" npts="15" start="10.0"/>
</variable>
<variable name="time" shape="time" type="double">
    <attribute name="calendar" type="string" value="360_day"/>
    <attribute name="standard_name" type="string" value="time"/>
    <attribute name="units" type="string" value="days since 2289-1-1"/>
    <attribute name="bounds" type="string" value="bound_time"/>
    <attribute name="axis" type="string" value="T"/>
    <values increment="360.0" npts="140" start="510.0"/>
</variable>
<variable name="bound_time" shape="time bounds_axis" type="double"> </variable>
<variable name="air_temperature" shape="time air_pressure latitude longitude" type="float">
```

```
        <attribute name="standard_name" type="string" value="air_temperature"/>

        <attribute name="units" type="string" value="Kir_temperature"/>

        <attribute name="long_name" type="string" value="temperature on pressure level"/>

        <attribute name="cell_methods" type="string" value="longitude: latitude: mean time: mean (interval:
4 h)"/>

        <attribute name="_FillValue" type="float" value="-1.073742e+09"/>

        <values xm:xmlmime="binary/octet-
stream">Li4uKGJhc2U2NCBlbmNvZGVkIGRhdGEpLi4u...(base64 encoded
data)...BlbmNvZGVkIGRhdGEpLi4u==</values>

    </variable>

</netcdf>

------=_NextPart_000_0000_93251752.3C5526C0--
```

B.3.3. C.3.5 Multipart section containing ncML with binary data extracted using XOP

```
------=_NextPart_000_0000_93251752.3C5526C0

MIME-Version: 1.0

Content-Type: Multipart/Related;boundary=MIME_boundary;  type="application/xop+xml";

    start="<12dea45c@example.net >";

    startinfo="application/ncML+xml";


--MIME_boundary

Content-Type: application/xop+xml; charset=UTF-8;

    type="application/ncML+xml; action=\"ProcessData\""

Content-ID: <12dea45c@example.net>

Content-Location: http://example.net/data/data2.ncml


<?xml version="1.0" encoding="UTF-8"?>

<netcdf xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xmlns="http://www.unidata.ucar.edu/namespaces/netcdf/ncml-2.2">

    <dimension name="longitude" length="96"/>

    <dimension name="bounds_axis" length="2"/>

    <dimension name="latitude" length="73"/>

    <dimension name="air_pressure" length="15"/>
```

```
<dimension name="time" length="140"/>
<attribute name="Conventions" type="string" value="CF-1.1"/>
<attribute name="source" type="string" value="Data from model run ABC123"/>
<variable name="longitude" shape="longitude" type="double">
   <attribute name="standard_name" type="string" value="longitude"/>
   <attribute name="units" type="string" value="degrees_east"/>
   <attribute name="bounds" type="string" value="bound_longitude"/>
   <attribute name="axis" type="string" value="X"/>
   <values increment="3.75" npts="96" start="0.0"/>
</variable>
<variable name="bound_longitude" shape="longitude bounds_axis" type="double"> </variable>
<variable name="latitude" shape="latitude" type="double">
   <attribute name="standard_name" type="string" value="latitude"/>
   <attribute name="units" type="string" value="degrees_north"/>
   <attribute name="bounds" type="string" value="bound_latitude"/>
   <attribute name="axis" type="string" value="Y"/>
   <values increment="-2.5" npts="73" start="90.0"/>
</variable>
<variable name="bound_latitude" shape="latitude bounds_axis" type="double"> </variable>
<variable name="air_pressure" shape="air_pressure" type="float">
   <attribute name="standard_name" type="string" value="air_pressure"/>
   <attribute name="units" type="string" value="hPa"/>
   <attribute name="axis" type="string" value="Z"/>
   <attribute name="positive" type="string" value="down"/>
   <values increment="70.71429" npts="15" start="10.0"/>
</variable>
<variable name="time" shape="time" type="double">
   <attribute name="calendar" type="string" value="360_day"/>
   <attribute name="standard_name" type="string" value="time"/>
   <attribute name="units" type="string" value="days since 2289-1-1"/>
   <attribute name="bounds" type="string" value="bound_time"/>
   <attribute name="axis" type="string" value="T"/>
```

```
        <values increment="360.0" npts="140" start="510.0"/>
    </variable>
    <variable name="bound_time" shape="time bounds_axis" type="double"> </variable>
    <variable name="air_temperature" shape="time air_pressure latitude longitude" type="float">
        <attribute name="standard_name" type="string" value="air_temperature"/>
        <attribute name="units" type="string" value="Kir_temperature"/>
        <attribute name="long_name" type="string" value="temperature on pressure level"/>
        <attribute name="cell_methods" type="string" value="longitude: latitude: mean time: mean (interval:
4 h)"/>
        <attribute name="_FillValue" type="float" value="-1.073742e+09"/>
        <values xm:xmlmime="binary/octet-stream"><xop:Include
    xmlns:xop='http://www.w3.org/2004/08/xop/include'
    href='cid:bd43gh2y@example.net'/>
        </values>
    </variable>
</netcdf>
--MIME_boundary
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <bd43gh2y@example.net>

...(binary encoded data)...
--MIME_boundary--
------=_NextPart_000_0000_93251752.3C5526C0--
```