Open Geospatial Consortium Inc.

Date: 2011-Agu 18

Reference number of this document: OGC 10-??

Version: **2.0.1**

Category: OpenGIS® Discussion Paper

Editors: Ben Domenico and Stefano Nativi

CF-netCDF Data Model extension specification

To obtain additional rights of use, visit http://www.opengeospatial.org/legal/.

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 2.0.1
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 "netCDF core" Data Model specification: OGC 10-090 [1].

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 National Research Council of Italy (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

Copyright © 2010 Open Geospatial Consortium

v. Revision history

Date	Release	Editor	Primary clauses modified	Description
2010-08-09	1.0.0	Stefano Nativi and Ben Domenico	All	Created
2010-09-10	1.1.0	Stefano Nativi	All	Extended to Multi-point coverages
2011-05-25	2.0	Stefano Nativi	CF conventions	Modified the CF convention data model and related sections to be compliant with version 1.5
2011-08-09	2.0.1	Stefano Nativi	CF conventions	Modified the CF convention data model and related sections to be compliant with version 1.6

Copyright © 2011 Open Geospatial Consortium

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 conventions and netCDF binary and XML encodings. However, it is important that CF and 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, including irregular gridded datasets. The current plan is to include these augmentations in subsequent versions of this standard extension.

This specification is written for netCDF version 3, but 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.

Copyright © 2011 Open Geospatial Consortium

Foreword

This document is an extension of the “netCDF core” Data Model specification: OGC 10-090 [1].

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

The data model specified by this document can be encoded using the “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.

Copyright © 2010 Open Geospatial Consortium

Introduction

The OGC netCDF encoding supports electronic encoding of geospatial data, that is, digital geospatial information representing space and time-varying phenomena.

NetCDF (network Common Data Form) is a data model for array-oriented scientific data, a freely distributed collection of access libraries implementing support for that data model, and a machine-independent format. Together, the interfaces, libraries, and format support the creation, access, and sharing of multi-dimensional scientific data.

This document specifies the CF-netCDF data model extension; every CF-netCDF dataset shall adhere to this specification.

CF-netCDF encoding format is netCDF conforming to the Climate and Forecast (CF) conventions (i.e. CF-netCDF). This specification is based on the netCDF (network Common Data Form) ver. 3.0 file format using the CF (Climate and Forecast) conventions ver. 1.6

For the scope of this specification, CF-netCDF data model deals with multi-dimensional discrete coverage data.

Copyright © 2011 Open Geospatial Consortium

CF-netCDF data model extension

1 Scope

This document specifies the CF-netCDF data model extension.

This standard specifies the CF-netCDF data model mapping onto the ISO 19123 coverage schema.

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

2 Conformance

Standardization target are CF-netCDF implementations (currently encodings).

This document establishes three requirements classes:

1. *CF-netCDF core*, of http://www.opengis.net/spec/netCDF_data-model/req/CF-netCDF-1.6-core
2. *CF-netCDF Discrete Sampling*, of http://www.opengis.net/spec/netCDF_data-model/req/CF-netCDF-*1.6-DiscreteSampling*
3. *CF-netCDF Mapping onto ISO 19123*, of http://www.opengis.net/spec/netCDF_data-model/req/CF-netCDF-*1.6-ISOMapping*

and three corresponding pertaining conformance class, *CF-netCDF*, with URIs:

1. http://www.opengis.net/spec/ netCDF_data-model/conf/CF-netCDF-1.6-core.
2. http://www.opengis.net/spec/ netCDF_data-model/conf/CF-netCDF-*1.6-DiscreteSampling*
3. http://www.opengis.net/spec/ netCDF_data-model/conf/CF-netCDF-*1.6-ISOMapping*

Requirements and conformance test URIs defined in this document are relative to http://www.opengis.net/spec/netCDF_data-model/.

Annex A lists the conformance tests which shall be exercised on any software artifact claiming to implement an OGC CF-netCDF extension for netCDF.

3 Normative references

This *OGC CF-netCDF Data Model extension* specification consists of the present document. The complete specification is identified by OGC URI http://www.opengis.net/spec/netCDF_data-model/, the document has OGC URI http://www.opengis.net/spec/netCDF_data-model/IS/CF-netCDF/.

The complete specification is available for download from http://www.opengis.net/spec/netCDF_data-model/ .

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

For this specification, there is one external normative document contain provisions that are quoted verbatim in this text and hence constitute provisions of this specification:

NASA ESDS-RFC-021v0.02, CF Metadata Conventions, April 2010. Available at:

http://www.esdswg.org/spg/rfc/esds-rfc-021/ESDS-RFC-021-v0.01.pdf

An overview of this specification is provided by [18].

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 CDL syntax

The ASCII format used to describe the contents of a netCDF file is called CDL (network Common Data form Language). This format represents arrays using the indexing conventions of the C programming language, i.e., index values start at 0, and in multidimensional arrays, when indexing over the elements of the array, it is the last declared dimension that is the fastest varying in terms of file storage order. The netCDF utilities *ncdump* and *ncgen* use this format.

4.2 Coverage

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

4.2.1 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.

4.2.2 **Feature**

abstraction of real world phenomena [4].

NOTE A feature may occur as a type or an instance.

4.2.3 **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 [4].

NOTE: the curves partition a space into grid cells.

4.2.4 **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.

4.2.5 **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.

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
xsd	http://www.w3.org/2001/XMLSchema	XML Schema
gml	http://www.opengis.net/gml/3.2	GML 3.2.1
ncml	http://unidata.ucar.edu/ncml	ncML schema

6 NetCDF Data Model extension

A formal specification of the netCDF data model is provided by [1]; while, [2] provides a general description of the netCDF technology.

For reader's convenience, the netCDF core data model is shown in Figure 1.

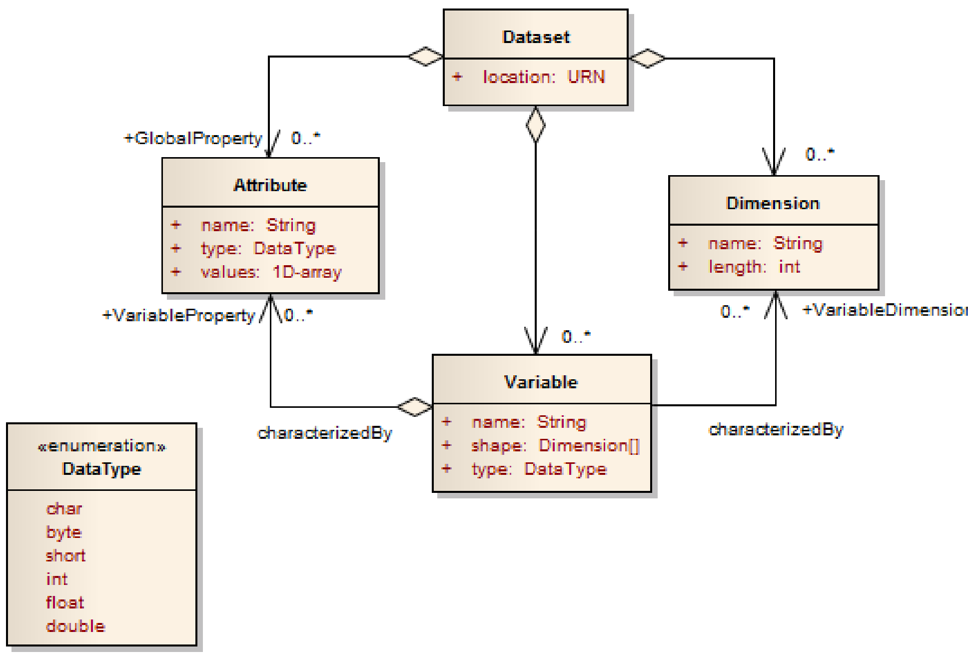

Figure 1 - NetCDF core data model [1]

6.1 CF Conventions

The purpose of the CF conventions is to require conforming datasets to contain sufficient metadata that they are self-describing in the sense that each variable in the file has an associated description of what it represents, including physical units if appropriate, and that each value can be located in space (relative to earth-based coordinates) and time.

This enables users of data from different sources to decide which quantities are comparable, and facilitates building applications with powerful extraction, regridding, and display capabilities. This specification considers version 1.6 of the CF conventions. The full CF Conventions reference documentation is provided in [8] and [16]. For this specification, provisions were given by [18].

6.1.1 **Introduced Requirements**

Most of the introduced requirements are taken from the "CF Conformance Requirements and Recommendations" published by the Program for Climate Model Diagnosis & Intercomparison (PCMDI) [10], from the "NetCDF Climate and Forecast (CF) Metadata

Conventions, Version 1.5" [8], and from "NetCDF Climate and Forecast (CF) Metadata Conventions, Version 1.6 -Chapter 9" [16].

For interoperability sake, the following requirement section specifies a CF 1.6 profile by restricting some of the original specifications –i.e. by transforming most of the recommendations into mandatory requirements.

7 CF-netCDF Data Model

7.1 Extension packages

As depicted in Figure 2, the following extension packages are modeled by this specification:

- General conventions;
- Variables and Standard attributes conventions;
- Dimensions and Dimensional Variables conventions;
- Coordinate Variables and Coordinate Types conventions;
- Coordinate Systems convention;
- Discrete Sampling Geometries.

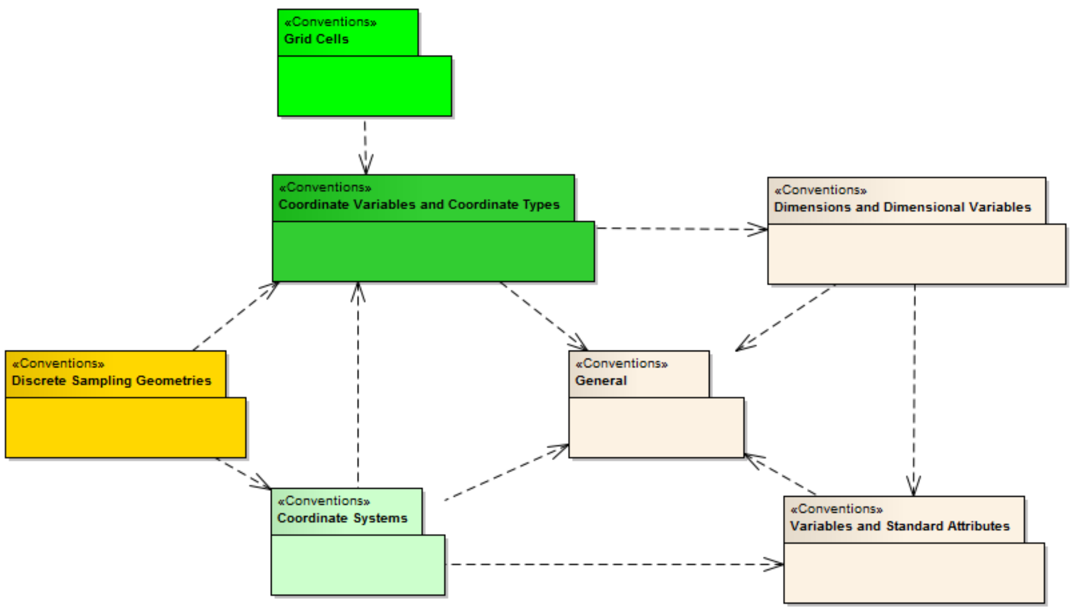

Figure 2 - CF-netCDF data model extension packages

7.2 Extension Data Model

For the scope of this specification, the CF-netCDF data model is shown in **Figure 3** and **Figure 4**.

Copyright © 2011 Open Geospatial Consortium

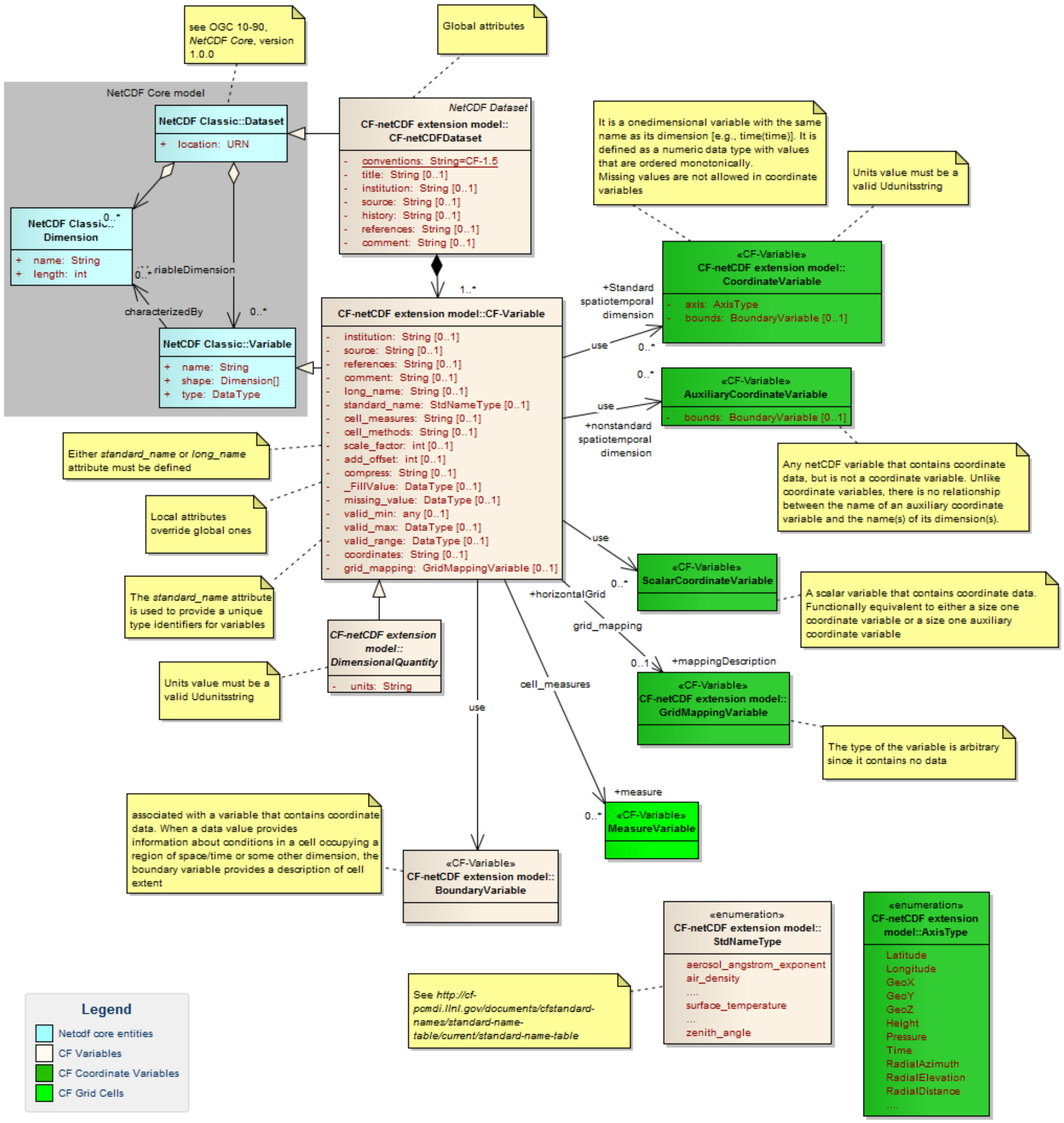

Figure 3 - CF-netCDF data model: CF Variable and Standard Attributes

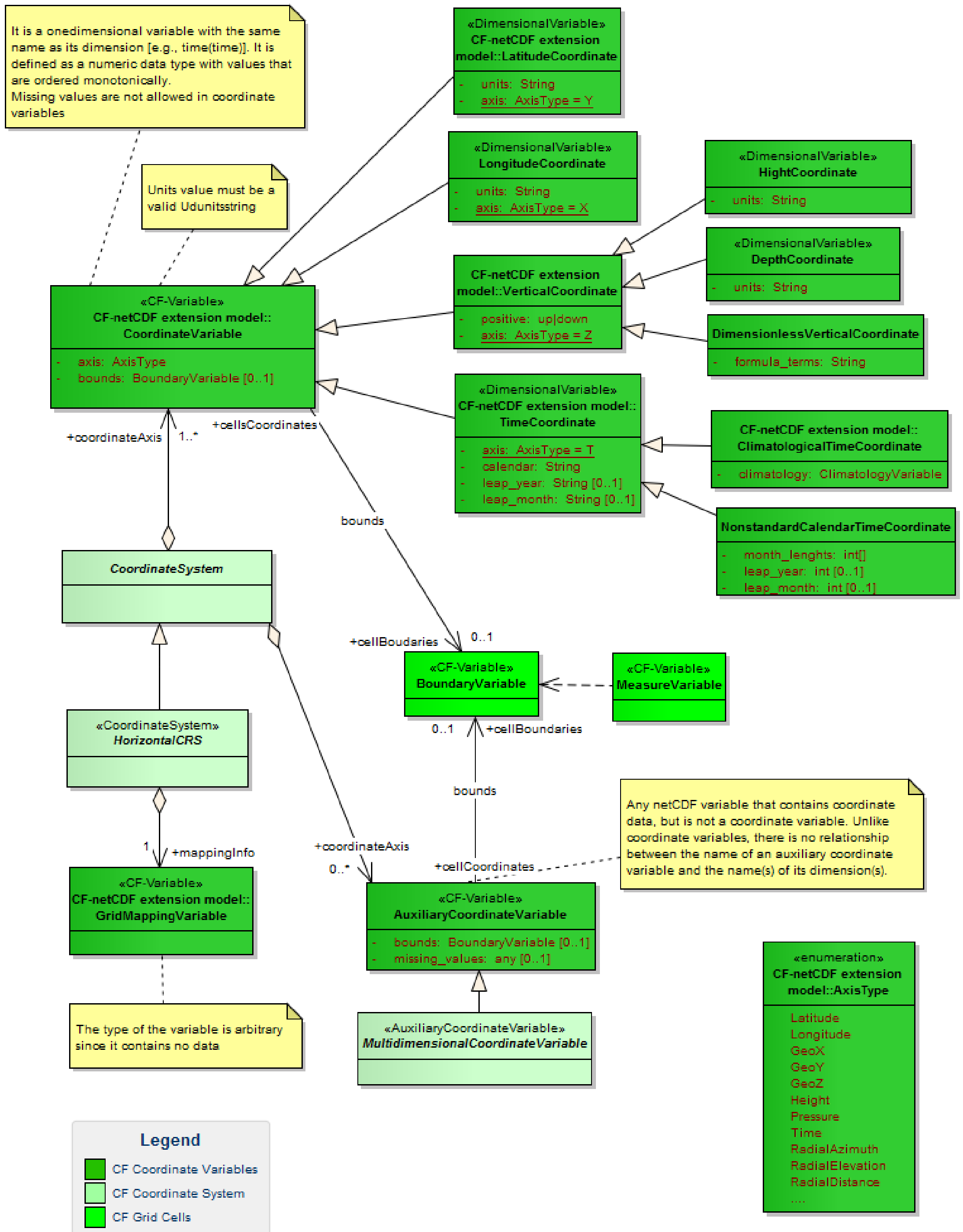

Figure 4 - CF-netCDF data model: CF Coordinate Variables, Coordinate Types, Coordinate Systems, and Grid Cells

Requirement 1 /req/CF-netCDF/structural-adherence: Any data instantiating a concrete `CF-netCDF dataset` **shall** conform with the UML diagrams in **Figure 3** and **Figure 4**Figure 1.

7.3 General Conventions

To identify that the file uses the CF convention, the `Conventions` global attribute shall be given the string value of "CF-1.6"

Requirement 2 ./req/CF-netCDF/Conventions: Any `CF-netCDF Dataset` that uses the CF convention **shall** define the global attribute `Conventions` to the string value "CF-1.6".

`Variable`, `dimension`, and `attribute` names should begin with a letter and be composed of letters, digits, and underscores.

Requirement 3 /req/CF-netCDF/NamingConventions: `CF-netCDF Variable`, `Dimension` and `Attribute` names **shall** begin with a letter and be composed of letters, digits, and underscores.

Names commencing with underscore ('_') are reserved for use by the netCDF library. Most generic applications that process netCDF datasets assume standard attribute conventions.

Requirement 4 /req/CF-netCDF/Reserved Attribute Names: `CF-netCDF Attribute names` commencing with underscore ('_') **shall** be reserved for use by the netCDF library.

Requirement 5 /req/CF-netCDF/Standard Attribute Names: The list of `CF-netCDF Attribute names` reported in the Appendix A of [8] **shall** be considered standard names and therefore reserved.

7.4 CF-netCDF Variables and Standard Attributes

Each variable in a netCDF file has an associated description which is provided by the attributes `units`, `long_name`, and `standard_name`. The `units`, and `long_name` attributes are defined in the netCDF User Guide document [7] and the `standard_name` attribute is defined in the CF conventions document [8].

Example of units, standard and long name values (CDL syntax):

```
netcdf temperature {
  dimensions:
    lat = 45;
    lon = 57;

  variables:
```

```
    double lat(lat);
      lat:standard_name="latitude";
      lat:long_name="latitude";
      lat:units="degrees_north";
    double lon(lon);
      lon:standard_name="longitude";
      lon:long_name="longitude";
      lon:units="degrees_east";

    double temperature(lon, lat);
      temperature:standard_name="air_temperature";
      temperature:long_name="temperature";
      temperature:units="K";
}
```

7.4.1 CF Standard names attribute

A fundamental requirement for exchange of scientific data is the ability to describe precisely the physical quantities being represented. The `standard_name` attribute is the name used to identify the physical quantity. It is used to provide unique identifiers for variables. *CF standard names* conventions describe what the numbers in a netCDF dataset represent; e.g., temperature, pressure, wind speed, salinity, radiance, reflectivity. Besides, this attribute can be used to identify variables that contain coordinate data.

7.4.2 Long name attribute

The `long_name` attribute is defined to contain a long descriptive name which may, for example, be used for labeling plots. For backwards compatibility with COARDS this attribute is optional. If a variable has no `long_name` attribute then an application may use, as a default the `standard_name` or the variable name itself [8].

Requirement 6 /req/CF-netCDF/StandardName#1: Any `CF-netCDF Variable` **shall** define either a `standard_name` attribute or a `long_name` attribute.

Exception, this is not mandatory only for `Bounday Variable`

Requirement 7 /req/CF-netCDF/StandardNames#2: A `standard name` **shall** contain no whitespace and **shall** be case sensitive.

In addition, the set of permissible standard names is contained in the CF standard name table published at: http://cf-pcmdi.llnl.gov/documents/cf-standard-names/standard-name-table/18/cf-standard-name-table.xml

7.5 CF-netCDF Dimensions and Dimensional Variables

A variable may have any number of `Dimensions`, including zero, and the dimensions must all have different names. The dimensions of the variable define the axes of the quantity it contains.

If any or all of the dimensions of a variable have the interpretations of "date or time" (`T`), "height or depth" (`Z`), "latitude" (`Y`), or "longitude" (`X`) then those dimensions must appear in the relative order `T`, then `Z`, then `Y`, then `X`. All other dimensions can be placed to the left of the spatiotemporal dimensions.

Requirement 8 /req/CF-netCDF/Dimensions: For any `CF-netCDF Variable,` its `Dimension` names **shall** have different names.

Requirement 9 /req/CF-netCDF/DimensionsShape: For a given spatial-temporal `CF-netCDF Variable,` its spatial-temporal `Dimensions` order **shall** appear in the relative order `T`, then `Z`, then `Y`, then `X`.

In addition, any other dimension **shall** be placed to the left of the spatiotemporal dimensions.

7.5.1 Units attribute

CF `units` conventions describe what are the *units of measure* for the numbers in a netCDF dataset.

The `units` attribute is required for all variables that represent dimensional quantities.

The value of the `units` attribute must be a string that can be recognized by Unidata's *Udunits* software package [9].

Requirement 10 /req/CF-netCDF/Units: Any dimensional `CF-netCDF Variable` (i.e. `CF-netCDF Variable` that represents dimensional quantity) **shall** define a `units` attribute.

Exception, this is not mandatory only for `Bounday Variable`

In addition, the `units` value may be physically equivalent (not necessarily identical) to the canonical units for the Variable.

Requirement 11 /req/CF-netCDF/UnitsValue: any `units` attribute value **shall** be a string that can be recognized by UNIDATA"s Udunits package [9].

In addition, exceptions are the units *level*, *layer*, and *sigma_level*.

Requirement 12 /req/CF-netCDF/Units#3: The `units` of a `CF-Variable` **shall** be consistent with the *units* given in the *standard name table* (published at:). http://cf-pcmdi.llnl.gov/documents/cf-standard-names/standard-name-table/18/cf-standard-name-table.xml
In addition, the units must also be consistent with a specified `cell_methods` attribute, if one is present.

7.6 CF-NetCDF Coordinate Variables and Coordinate Types

A variable's spatiotemporal dimensions are used to locate data values in time and space. This is accomplished by associating these dimensions with the relevant set of latitude, longitude, vertical, and time coordinates [8].

Variables with a single dimension whose names match the name of their dimension are called "coordinate variables". By convention coordinate variables define the physical coordinate for that dimension (see the netCDF Users Guide [7]).

For instance, the following dataset description shows a variable (i.e. `xwind`) that is defined on a set of spatiotemporal dimensions: latitude, longitude, vertical, and time dimension; each axis is identified by a coordinate variable. `xwind(n,k,j,i)` is associated with the coordinate values `lon(i)`, `lat(j)`, `pres(k)`, and `time(n)`.

```
dimensions:
  lat = 18 ;
  lon = 36 ;
  pres = 15 ;
  time = 4 ;
variables:
  float xwind(time,pres,lat,lon) ;
      xwind:long_name = "zonal wind" ;
      xwind:units = "m/s" ;
  float lon(lon) ;
      lon:long_name = "longitude" ;
      lon:units = "degrees_east" ;
  float lat(lat) ;
      lat:long_name = "latitude" ;
      lat:units = "degrees_north" ;
  float pres(pres) ;
      pres:long_name = "pressure" ;
      pres:units = "hPa" ;
  double time(time) ;
      time:long_name = "time" ;
      time:units = "days since 1990-1-1 0:0:0" ;
```

There are two methods used to identify variables that contain coordinate data. The first is to use the `Coordinate Variable` types, the second is the use of `Auxiliary Coordinate Variables`.

Copyright © 2011 Open Geospatial Consortium

The use of `Coordinate Variables` is required for all dimensions that correspond to one dimensional space or time coordinates. All of a variable's dimensions that are latitude, longitude, vertical, or time dimensions must have corresponding coordinate variables, i.e., one-dimensional variables with the same name as the dimension.

`Coordinate Variable` is defined as a numeric data type with values that are ordered monotonically. Missing values are not allowed in coordinate variables.

Requirement 13 /req/CF-netCDF/CoordinateData: Any `Variable` containing coordinate data **shall** be: a `Coordinate Variable |` an `Auxiliary Coordinate Variable`

Requirement 14 /req/CF-netCDF/CoordinateData/CoordinateVariable: A `Coordinate Variable` **shall** be defined for each `Dimension` that correspond to one dimensional space or time coordinates.

Requirement 15 /req/CF-netCDF/CoordinateVariable#1: any `Coordinate Variable` **shall** have a single `Dimension` whose name matches the the `Variable name`.

Requirement 16 /req/CF-netCDF/CoordinateVariable#2: `Coordinate Variable` values **shall** be ordered monotonically.

Requirement 17 /req/CF-netCDF/CoordinateVariable/AxisAttribute#1: for any `Coordinate Variable` the attribute `axis` **shall** be given one of the values `X`, `Y`, `Z` or `T`.

In addition these values stand for a longitude, latitude, vertical, or time axis respectively for the CF-netCDF cordinate types: `LongitudeCoordinate, LatitudeCoordinate, VerticalCoordinate`, `TimeCoordinate`.

Requirement 18 /req/CF-netCDF/ CoordinateVariable/AxisAttribute#2: The values `X` and `Y` for the `axis` attribute **shall** be used to identify horizontal coordinate variables.

Copyright © 2010 Open Geospatial Consortium

Requirement 19 /req/CF-netCDF/CoordinateVariable/Missing_valuesAttribute: `Coordinate Variable` shall not define an attribute `missing_values`.

The CF convention gives special constraints to latitude, longitude, vertical, and time coordinates (i.e. `LatitudeCoordinate`, `LongitudeCoordinate`, `VerticalCoordinate`, `TimeCoordinate`).

7.6.1 Latitude Coordinate

The recommended unit of latitude is `degrees_north`. Also acceptable are `degree_north`, `degree_N`, `degrees_N`, `degreeN`, and `degreesN`.

Requirement 20 **/req/CF-netCDF/LatitudeCoordinate**: For any `Latitude Coordinate` the `units` attribute values **shall** be: `degrees_north | degree_north | degree_N`, | `degrees_N` | `degreeN` | `degreesN`.

In addition, Coordinates of latitude with respect to a rotated pole should be given units of `degrees`, not `degrees_north` or equivalents,

Example of Latitude axis

```
float lat(lat) ;
   lat:long_name = "latitude" ;
   lat:units = "degrees_north" ;
   lat:standard_name = "latitude" ;
```

7.6.2 Longitude Coordinate

The recommended unit of longitude is `degrees_east`. Also acceptable are `degree_east, degree_E, degrees_E, degreeE,` and `degreesE`.

Requirement 21 **/req/CF-netCDF/LongitudeCoordinate**: For any `Longitude Coordinate` the `units` attribute values **shall** be: `degrees_east | degree_east | degree_E`, | `degrees_E` | `degreeE` | `degreesE`.

In addition, Coordinates of longitude with respect to a rotated pole should be given units of `degrees`, not `degrees_east` or equivalents,

Example of `Latitude` axis

```
float lon(lon) ;

   lon:long_name = "longitude" ;
```

```
lon:units = "degrees_east" ;
lon:standard_name = "longitude" ;
```

7.6.3 Vertical (Height, Depth, Dimensional) Coordinate

The direction of positive (i.e., the direction in which the coordinate values are increasing), whether up or down, cannot in all cases be inferred from the units. For this reason the attribute `positive` is required. The `positive` attribute may have the value `up` or `down` (case insensitive). This attribute may be applied to either coordinate variables or auxillary coordinate variables that contain vertical coordinate data.

Requirement 22 /req/CF-netCDF/VerticalCoordinatePositiveAttribute: Any `Vertical Coordinate` **shall** define the `positive` attribute whose values **shall** be: `up | down`.

The acceptable units for vertical (depth or height) coordinate variables are:
units of pressure as listed in the file `udunits.dat`. For vertical axes the most commonly used of these include include `bar`, `millibar`, `decibar`, `atmosphere (atm)`, `pascal (Pa)`, and `hPa`.
• units of length as listed in the file udunits.dat. For vertical axes the most commonly used of these include `meter (metre, m)`, and `kilometer (km)`.
• other units listed in the file udunits.dat that may under certain circumstances reference vertical position such as units of density or temperature.

Requirement 23 /req/CF-netCDF/VerticalCoordinateUnitsAttribute: For any `Vertical Coordinate` the `units` attribute values **shall** be: units of pressure as listed in the file `udunits.dat` (e.g. `bar`, `millibar`, `decibar`, `atmosphere (atm)`, `pascal (Pa)`, and `hPa`) `|` units of length as listed in the file `udunits.dat` (e.g. `meter, metre, m, kilometer, km)` `|` other units listed in the file `udunits.dat` that may under certain circumstances reference vertical position such as units of density or temperature.

7.6.4 Dimensionless Vertical coordinates

Dimensionless vertical coordinates are defined in the Appendix D of [8]. The `standard_name` attribute associates a coordinate with its definition which provides a mapping between the dimensionless coordinate values and dimensional values that can positively and uniquely indicate the location of the data.

The `formula_terms` attribute is used to associate terms in the definitions with variables in a netCDF file; for this reason it is mandatory.

Copyright © 2010 Open Geospatial Consortium

Requirement 24 /req/CF-netCDF/DimensionlessVerticalCoordinate: Any `Dimensioless Vertical Coordinate` **shall** be defined in Appendix D of [8].

Requirement 25 /req/CF-netCDF/DimensionlessVerticalCoordinateFormula_TermsAttribute: Any `Dimensioless Vertical Coordinate` **shall** define the `formula_terms` attribute.

Example of Atmosphere sigma coordinate

```
float lev(lev) ;
   lev:long_name = "sigma at layer midpoints" ;
   lev:positive = "down" ;
   lev:standard_name = "atmosphere_sigma_coordinate" ;
   lev:formula_terms = "sigma: lev ps: PS ptop: PTOP" ;
```

The `formula_terms` attribute associates the variable `lev` with the term `sigma`, the variable `PS` with the term `ps`, and the variable `PTOP` with the term `ptop`. Thus the pressure at gridpoint `(n,k,j,i)` would be calculated by:

```
p(n,k,j,i) = PTOP + lev(k)*(PS(n,j,i)-PTOP)
```

7.6.5 Time Coordinate

The `units` attribute takes a string value formatted as per the recommendations in the `Udunits` package [9].

Requirement 26 /req/CF-netCDF/TimeCoordinateUnitsAttribute: For any `Time Coordinate` the `units` attribute values **shall** be units of time as listed in the file `udunits.dat` (e.g. `day` or `(d)`,`hour` or `(hr, h)`,`minute` or `(min)`, and `second` or `(sec, s)`).

Example of Time axis:

```
double time(time) ;
   time:long_name = "time" ;
   time:units = "days since 1990-1-1 0:0:0" ;
```

7.6.5.1 Calendar attribute

To calculate a new date and time given a base date, base time and a time increment one must know what calendar to use. For this purpose it is important the calendar be specified by the attribute `calendar` which is assigned to the time coordinate variable.
The `calendar` attribute may be set to `none` in climate experiments that simulate a fixed time of year or when a non-standard calendar is being used -e.g. non-standard paleoclimate eras .

Requirement 27 /req/CF-netCDF/TimeCoordinateCalendarAttribute: Any `Time Coordinate` **shall** define the `calendar` attribute

In addition, the values currently defined for `calendar` are: `gregorian` or `standard | proleptic_gregorian | noleap` or `365_day | all_leap` or `366_day | 360_day | julian | none`.

Example of Perpetual time axis

```
variables:
   double time(time) ;
      time:long_name = "time" ;
      time:units = "days since 1-7-15 0:0:0" ;
      time:calendar = "none" ;
data:
   time = 0., 1., 2., ...;
```

If none of the standard calendars (e.g., calendars appropriate to a different paleoclimate era), a nonstandard calendar must be defined. The lengths of each month are explicitly defined with the `month_lengths` attribute of the time axis. If leap years are included, then two other attributes of the time axis should also be defined: `leap_year` and `leap_month`.

Requirement 28 /req/CF-netCDF/TimeCoordinateNonstandardCalendar: Any `Time Coordinate` characterized by a nonstandard calendar **shall** define the `month_lengths` attribute to specify its calendar.

In addition, the attribute value **shall** be for non-leap year: a vector of size 12, specifying the number of days in the months from January to December.

In addition, for leap year two other attributes of the time axis **shall** be defined: `leap_year` and `leap_month`.

7.6.6 Auxiliary Coordinate Variable

In cases where coordinate variables are not applicable (e.g. a `CF-Variable` characterized by spatiotemporal dimensions that are not latitude, longitude, vertical, or time dimensions), `Auxiliary Coordinate Variables` are defined. The

`Auxiliary Coordinate Variables` are identified (and linked to the `CF-Variable`) through the `coordinates` attribute defined by the `CF-Variable`.

Unlike `Coordinate Variables`, there is no relationship between the name of an auxiliary coordinate variable and the name(s) of its dimension(s).

Requirement 29 /req/CF-netCDF/AuxiliaryCoordinateVariable: An `Auxiliary Coordinate Variable` **shall** be identified by the `coordinates` attribute defined by a `netCDF Variable`.

In addition, The value of the `coordinates` attribute is a blank separated list of the names of `Auxiliary Coordinate Variables`

7.6.7 Scalar Coordinate Variables

When a variable has an associated coordinate which is single-valued, that coordinate may be represented as a scalar variable. Since there is no associated dimension these scalar coordinate variables should be attached to a data variable via the `coordinates` attribute.

Once a name is used for a scalar coordinate variable it cannot be used for a 1D coordinate variable. For this reason it is not allowed using a name for a scalar coordinate variable that matches the name of any dimension in the netCDF dataset.

Example of `Scalar Coordinate Variables` (i.e. `atime` and `P500` variables) [8]:

```
dimensions:
   lat = 180 ;
   lon = 360 ;
   time = UNLIMITED ;
variables:
   double atime
      atime:standard_name = "forecast_reference_time" ;
      atime:units = "hours since 1999-01-01 00:00" ;
   double time(time);
      time:standard_name = "time" ;
      time:units = "hours since 1999-01-01 00:00" ;
   double lon(lon) ;
      lon:long_name = "station longitude";
      lon:units = "degrees_east";
   double lat(lat) ;
      lat:long_name = "station latitude" ;
      lat:units = "degrees_north" ;
   double p500
      p500:long_name = "pressure" ;
      p500:units = "hPa" ;
      p500:positive = "down" ;
   float height(time,lat,lon);
      height:long_name = "geopotential height" ;
```

```
       height:standard_name = "geopotential_height" ;
       height:units = "m" ;
       height:coordinates = "atime p500" ;
data:
   time = 6., 12., 18., 24. ;
   atime = 0. ;
   p500 = 500.
```

Requirement 30 /req/CF-netCDF/ScalarCoordinateVariable: A `Scalar Coordinate Variable` **shall** define a coordinate which is single-valued.

In addition, the `Scalar Coordinate Variable` name **shall not** match the name of any dimension in the netCDF dataset.

7.7 Coordinate Systems

7.7.1 Independent Latitude, Longitude, Vertical, and Time Axes

When each of a variable's spatiotemporal dimensions is a latitude, longitude, vertical, or time dimension, then each axis is identified by a coordinate variable.

7.7.2 Horizontal Coordinate Reference Systems, Grid Mappings, and Projections

When the coordinate variables for an horizontal grid are not longitude and latitude, it is required that the true latitude and longitude coordinates be supplied via the `coordinates` attribute [8].

If in addition it is desired to describe the mapping between the given coordinate variables and the true latitude and longitude coordinates, the attribute `grid_mapping` may be used to supply this description.

This attribute is attached to data variables so that variables with different mappings may be present in a single file. The attribute takes a string value which is the name of another variable in the file that provides the description of the mapping via a collection of attached attributes. This variable is called *a grid mapping variable* and is of arbitrary type since it contains no data. Its purpose is to act as a container for the attributes that define the mapping.

The one attribute that all grid mapping variables must have is `grid_mapping_name` which takes a string value that contains the mapping's name. The other attributes that define a specific mapping depend on the value of `grid_mapping_name`. The valid values of `grid_mapping_name` along with the attributes that provide specific map parameter values are described in the Appendix F of [8].

When the coordinate variables for an horizontal grid are longitude and latitude, a grid mapping variable with `grid_mapping_name` equal to `latitude_longitude` may be used to specify the ellipsoid and prime meridian.

Requirement 31 /req/CF-netCDF/HorizontalCRS: For any `CF Variable` defined on an horizontal grid not defined on latitude and longitude dimensions, true latitude and longitude coordinates **shall** be supplied as `Coordinate Variables` and associated via the `coordinates` attribute.

In addition: the attribute `grid_mapping` may be used to supply the description of the mapping between the given grid coordinate variables and the true latitude and longitude coordinates.

In addition: the `grid_mapping` attribute takes a string value which is the name of a `Grid Mapping Variable`

Requirement 32 /req/CF-netCDF/GridMappingVariable: Any `Grid Mapping Variable` **shall** define the `grid_mapping_name` attribute..

Example of Horizontal grid data characterized by a grid mapping for rotated pole grid (i.e. `T(let, rlat, rlon)`)

```
dimensions:
   rlon = 128 ;
   rlat = 64 ;
   lev = 18 ;
variables:
   float T(lev,rlat,rlon) ;
      T:long_name = "temperature" ;
      T:units = "K" ;
      T:coordinates = "lon lat" ;
      T:grid_mapping = "rotated_pole" ;
   char rotated_pole
      rotated_pole:grid_mapping_name = "rotated_latitude_longitude" ;
      rotated_pole:grid_north_pole_latitude = 32.5 ;
      rotated_pole:grid_north_pole_longitude = 170. ;
   float rlon(rlon) ;
      rlon:long_name = "longitude in rotated pole grid" ;
      rlon:units = "degrees" ;
      rlon:standard_name = "grid_longitude";
   float rlat(rlat) ;
      rlat:long_name = "latitude in rotated pole grid" ;
      rlat:units = "degrees" ;
      rlon:standard_name = "grid_latitude";
   float lev(lev) ;
      lev:long_name = "pressure level" ;
      lev:units = "hPa" ;
   float lon(rlat,rlon) ;
      lon:long_name = "longitude" ;
      lon:units = "degrees_east" ;
   float lat(rlat,rlon) ;
      lat:long_name = "latitude" ;
      lat:units = "degrees_north" ;
```

7.8 Grid Cells

When gridded data does not represent the point values of a field but instead represents some characteristic of the field within cells of finite "volume," a complete description of the variable should include metadata that describes the domain or extent of each cell, and the characteristic of the field that the cell values represent [8].

7.8.1 Cell Boundary Variable

The attribute `bounds` is used to represent cells to the appropriate coordinate variable(s). The value of `bounds` is the name of the variable that contains the vertices of the cell boundaries.

This type of variable is referred as "boundary variable." A boundary variable will have one more dimension than its associated `Coordinate` or `Auxiliary Coordinate Variable`. The additional dimension must the most rapidly varying one, and its size is the maximum number of cell vertices.

Since a boundary variable is considered to be part of a coordinate variable's metadata, it is not necessary to provide it with attributes such as `long_name` and `units`.

This approach supports both regular and non-regular grids (i.e. non-contiguous intervals defined on a given axis).

Requirement 33 /req/CF-netCDF/BoundaryVariable: Any `Boundary Variable` **shall** define one more dimension than its associated `Coordinate` or `Auxiliary Coordinate Variable`.

In addition, The additional dimension **shall** the most rapidly varying one, and its size is the maximum number of cell vertices

Example of cell boundaries on a latitude and longitude axes.

```
dimensions:
  lat = 45;
  lon = 57;
  nv = 2; // number of vertices

variables:
  double lat(lat);
    lat:long_name = ”latitude”;
    lat:units = ”degrees_north”;
    lat:bounds = “lat_bnds”;
  double lon(lon);
    lon:long_name = ”longitude”;
    lon:units = ”degrees_east”;
    lon:bounds = “lon_bnds”;
  double lat_bnds(lat, nv);
  double lon_bnds(lon, nv);
```

7.8.2 Cell Measure Variable

For some calculations, information is needed about the size, shape or location of the cells that cannot be deduced from the coordinates and bounds without special knowledge that a generic application cannot be expected to have [8].

To indicate such extra information, a `cell_measures` attribute may be defined for a variable. This is a string attribute comprising a list of blank-separated pairs of words of the form "`measure: name`". For the moment, "`area`" and "`volume`" are the only defined measures, but others may be supported in future. The "name" is the name of the variable containing the measure values, which we refer to as a "measure variable".

The dimensions of the measure variable must be the same as or a subset of the dimensions of the variable to which they are related. In the case of area, for example, the field itself might be a function of longitude, latitude, and time, but the variable containing the area values would only include longitude and latitude dimensions.

The measure variable must have a `units` attribute and may have other attributes such as a `standard_name`.

Requirement 34 /req/CF-netCDF/CellMeasures: `cell_measures` attribute **shall** be a string attribute comprising a list of blank-separated pairs of words of the form "*measure: name*".

Requirement 35 /req/CF-netCDF/MeasureVariable: `Mesure Variable` dimensions **shall** be the same as or a subset of the dimensions of the `CF Variable` to which they are related.

Example of cell areas for a spherical geodesic grid.

```
dimensions:
   cell = 2562 ; // number of grid cells
   time = 12 ;
   nv = 6 ; // maximum number of cell vertices
variables:
   float PS(time,cell) ;
      PS:units = "Pa" ;
      PS:coordinates = "lon lat" ;
      PS:cell_measures = "area: cell_area" ;
   float lon(cell) ;
      lon:long_name = "longitude" ;
      lon:units = "degrees_east" ;
      lon:bounds="lon_vertices" ;
   float lat(cell) ;
      lat:long_name = "latitude" ;
      lat:units = "degrees_north" ;
```

```
      lat:bounds="lat_vertices" ;
   float time(time) ;
      time:long_name = "time" ;
      time:units = "days since 1979-01-01 0:0:0" ;
   float cell_area(cell) ;
      cell_area:long_name = "area of grid cell" ;
      cell_area:standard_name="area";
      cell_area:units = "m2"
   float lon_vertices(cell,nv) ;
   float lat_vertices(cell,nv) ;
```

7.8.3 Cell Methods

To describe the characteristic of a field that is represented by cell values, the `cell_methods` attribute of the variable is defined. This is a string attribute comprising a list of blank-separated words of the form "*name: method*". Each "*name: method*" pair indicates that for an axis identified by *name*, the cell values representing the field have been determined or derived by the specified *method* [8].

For example, if data values have been generated by computing time means, then this could be indicated with `cell_methods="t: mean"`, assuming here that the name of the time dimension variable is "t".

In the specification of the `cell_methods` attribute, *name* can be a dimension of the variable, a scalar coordinate variable, a valid standard name, or the word "`area`". (See Section 7.3.4, "Cell methods when there are no coordinates" of [8] concerning the use of standard names in cell_methods).

The values of *method* should be: `point`, `sum`, `mean`, `maximum`, `minimum`, `mid_range`, `standard_deviation`, `variance`, `mode`, and `median`.

Requirement 36 /req/CF-netCDF/CellMethods: `cell_methods` attribute **shall** be a string attribute comprising a list of blank-separated words of the form "*name: method*".

In addition, *name* can be: a `Dimension` of the variable (defining the attribute), a `Scalar Coordinate Variable`, a valid standard name, or the word "`area`".

In addition, the value of *method* should be: `point`|`sum`|`mean` | `maximum` | `minimum`|`mid_range` | `standard_deviation`|`variance`, `mode` | `median`.

Example: 12-hourly timeseries of pressure, temperature and precipitation from a number of stations, where pressure is measured instantaneously, maximum temperature for the preceding 12 hours is recorded, and precipitation is accumulated in a rain gauge [8].

```
dimensions:
```

```
    time = UNLIMITED; // (5 currently)
    station = 10;
    nv = 2;
variables:
    float pressure(station,time)(time,station);
       pressure:long_name = "pressure";
       pressure:units = "kPa";
       pressure:cell_methods = "time: point";
    float maxtemp(station,time)(time,station);
       maxtemp:long_name = "temperature";
       maxtemp:units = "K";
       maxtemp:cell_methods = "time: maximum";
    float ppn(station,time)(time,station);
       ppn:long_name = "depth of water-equivalent precipitation";
       ppn:units = "mm";
       ppn:cell_methods = "time: sum";
    double time(time);
       time:long_name = "time";
       time:units = "h since 1998-4-19 6:0:0";
       time:bounds = "time_bnds";
    double time_bnds(time,nv);
data:
    time = 0., 12., 24., 36., 48.;
    time_bnds = -12.,0., 0.,12., 12.,24., 24.,36., 36.,48.;
```

7.9 Discrete Sampling Geometries

Datasets representing `Discrete Sampling Geometries` -such as time series, vertical profiles and trajectories- are characterized by a dimensionality that is lower than that of the space-time region that is sampled; discrete sampling geometries are typically "paths" through space-time.

CF ver. 1.6 specification [16] represents `Discrete Sampling Geometries` datasets as `Feature Collections` which are a collection of `Feature` (i.e. instances) made up of data `Elements`. **Figure 5** and **Figure 6** shows the conventions model.

Copyright © 2011 Open Geospatial Consortium

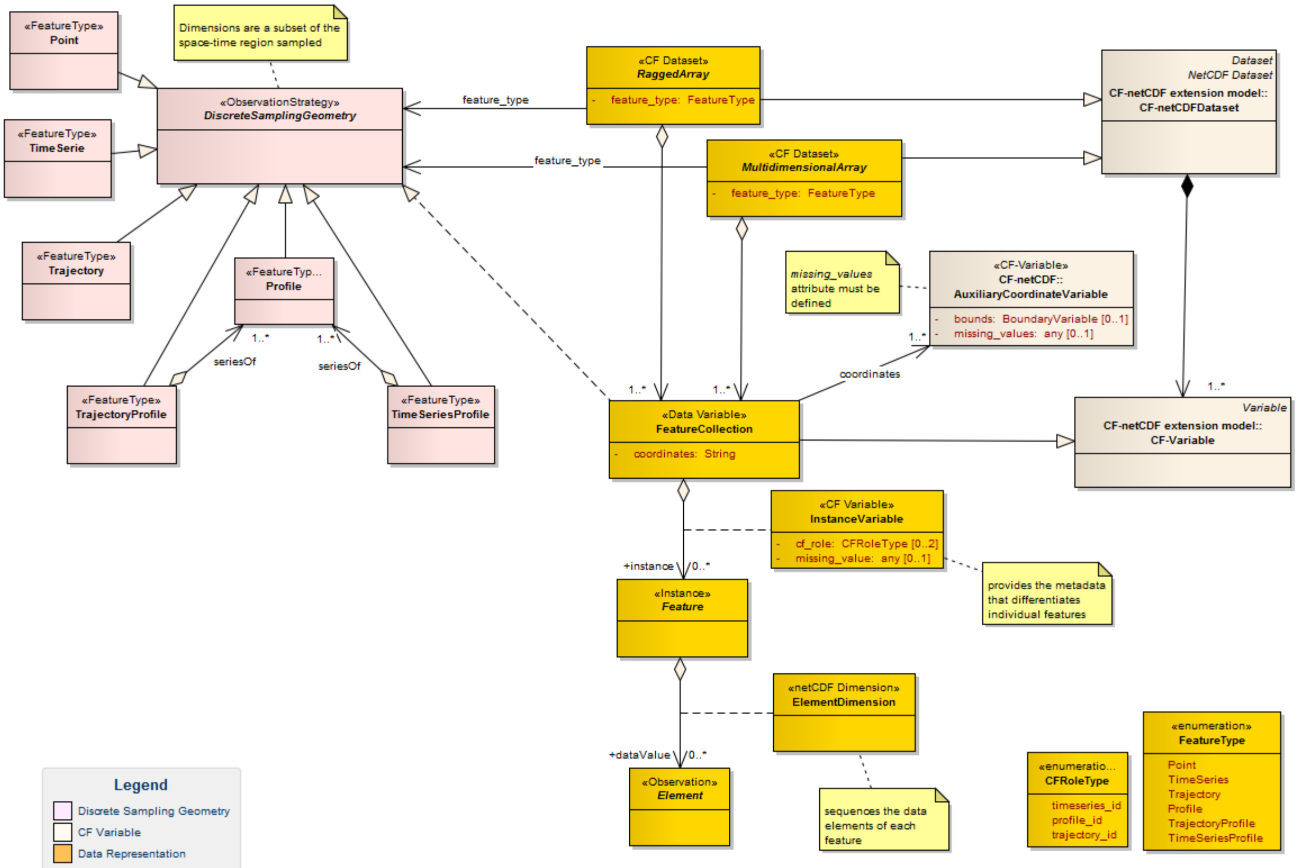

Figure 5 - `Discrete Sampling Geometries` data model: context

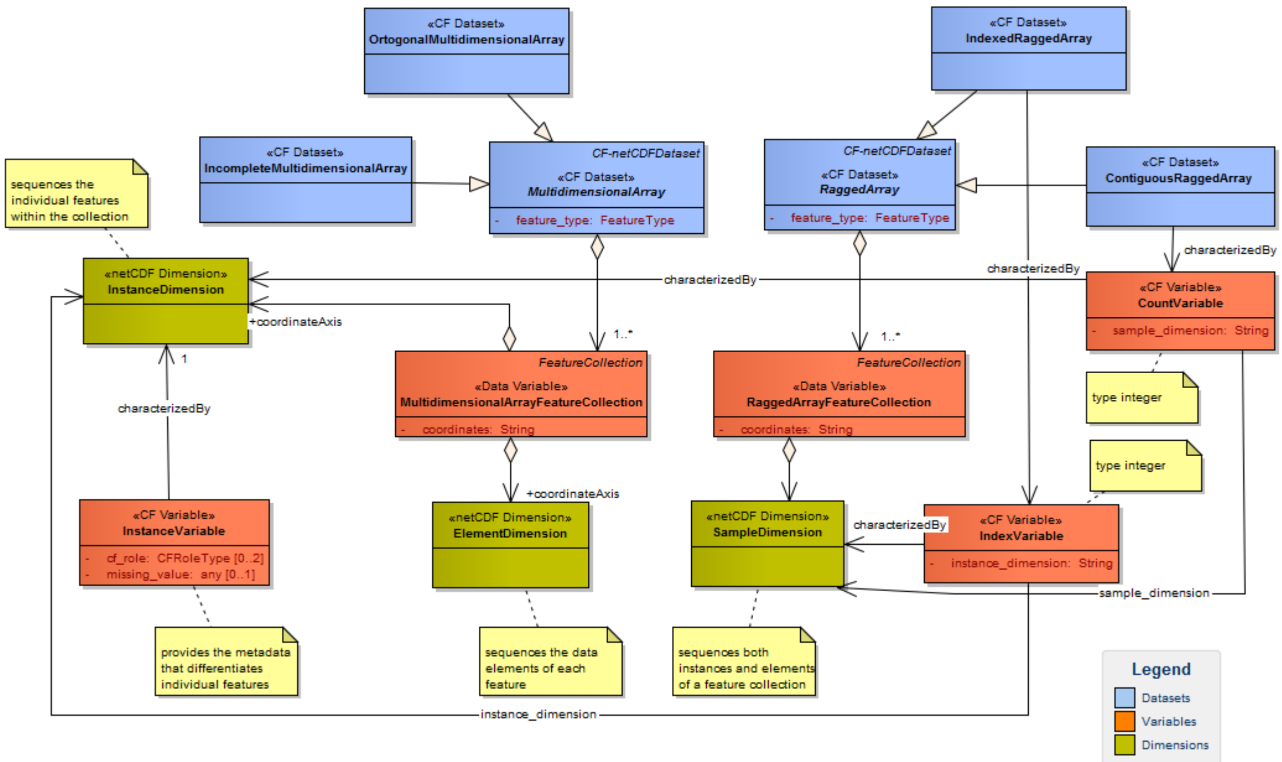

Figure 6 - `Discrete Sampling Geometries` data model: data representations

Requirement 37 /req/CF-netCDF/DistrictSamplingGeometriesModel: Any data instantiating a concrete `CF-netCDF DistrictSamplingGeometry dataset` shall conform with the UML diagrams in Figure 5 and Figure 6Figure 4Figure 1.

7.9.1 Features and Feature Types

The features contained within a collection must always be of the same type; and all the collections contained in a CF file must be of the same feature type.

Requirement 38 **/req/CF-netCDF/FeatureCollection**: `CF-netCDF Feature Collection` **shall** contain `Feature` instances of the same type.

Requirement 39 **/req/CF-netCDF/Feature**: `CF-netCDF Dataset` **shall** contain `Feature Collection` variables of the same feature type.

For the scope of this specification, Feature Types are listed in **Table 1** along with their mandatory dimensions [16].

Table 1 - Mandatory dimensions for Feature Type

featureType	Description of a single feature with this discrete sampling geometry	
	Form of a data variable containing values defined on a collection of these features	**Mandatory space-time coordinates for a collection of these features**
point	a single data point (having no implied coordinate relationship to other points)	
	data(i)	x(i) y(i) t(i)
timeSeries	a series of data points at the same spatial location with monotonically increasing times	
	data(i,o)	x(i) y(i) t(i,o)
trajectory	a series of data points along a path through space with monotonically increasing times	
	data(i,o)	x(i,o) y(i,o) t(i,o)
profile	an ordered set of data points along a vertical line at a fixed horizontal position and fixed time	
	data(i,o)	x(i) y(i) z(i,o) t(i)
timeSeriesProfile	a series of profile features at the same horizontal position with monotonically increasing times	
	data(i,p,o)	x(i) y(i) z(i,p,o) t(i,p)
trajectoryProfile	a series of profile features located at points ordered along a trajectory	
	data(i,p,o)	x(i,p) y(i,p) z(i,p,o) t(i,p)

i=instance dimension; o, p=element dimensions; x and y=horizontal dimensions; z=vertical dimension; t=temporal dimension.

Requirement 40 /req/CF-netCDF/FeatureTypeDimension: Any `CF-netCDF Feature Collection` variable implementing a `Feature type` **shall** define the `Dimensions` specified in **Table 1**.

7.9.2 Collections, instances and elements

Referring to **Table 1**, the `Instance Dimension` (dimension with subscript i) identifies a particular feature within a collection of features. `Instance Variable` is a one-dimensional variable (defined in a Discrete Geometry CF `Dataset`), which has *only* this dimension (such as x(i) y(i) and z(i) for a `TimeSeries`. `Instance Variables` provide the metadata that differentiates individual features [16].

`Feature` data values are defined as `Elements.` The `Element Dimensions` (dimensions with subscripts o and p) distinguish the data elements that compose a single `Feature.` For example in a collection of `TimeSeries` features, each time series instance, i, has data values at various times, o. In a collection of `Profile` features, the subscript, o, provides the index position along the vertical axis of each profile instance [16]

7.9.3 Representations of collections of features in data variables

The individual `Features` within a `Collection` need not necessarily contain the same number of `Elements`. For example, observed *in situ* time series will commonly contain unique numbers of time points, reflecting different deployment dates of the instruments. Other data sources, such as the output of numerical models, may commonly generate features of identical size.

Four types of representation are possible [16], see **Figure 5**:

- two `Multidimensional Array` representations, in which each `Feature` instance is allocated the identical amount of storage space. In these representations the `Instance Dimension` and the `Element Dimension`(s) are distinct CF `Coordinate Axes`;
- two `Ragged Array` representations, in which each `Feature` is provided with the minimum amount of space that it requires. In these representations the instances of the individual `Features` are stacked sequentially along the same array dimension as the `Elements` of the `Features`; this combined dimension is defined as `Sample Dimension`.

Requirement 41 /req/CF-netCDF/MultidimensionalArray: Any Multidimensional Array Feature Collection **shall** have both an Instance Dimension and an Element Dimension.

Requirement 42 /req/CF-netCDF/RaggedArray: Any `Ragged Array Feature Collection` **shall** have a `Sample Dimension`.

In addition, the `Sample Dimension` is occupied by both an `Instance Dimension` and an `Element Dimension.`

7.9.4 Orthogonal multidimensional array representation

The `Orthogonal Multidimensional Array` representation is the simplest representation; it can be used if each `Feature` instance in the `Collection` has identical `Coordinates` along the `Element Dimension` of the `Features`.

Requirement 43 /req/CF-netCDF/OrthogonalMultidimensionalArray: Any `Feature` instance of an `Orthogonal Multidimensional Array Feature Collection` **shall** have identical `Coordinates` along the `Element Dimension`.

7.9.5 Incomplete multidimensional array representation

The `Incomplete Multidimensional Array` representation can used if the features within a collection do not all have the same number of `Elements`, but sufficient storage space is available to allocate the number of `Elements` required by the longest `Feature` to all `Features`. That is, `Features` that are shorter than the longest `Feature` must be padded with missing values to bring all instances to the same storage size [16].

7.9.6 Contiguous ragged array representation

The `Contiguous Ragged Array` representation can be used only if the size of each `Feature` is known at the time that it is created [16].

In this representation, the `Dataset` contains a `Count Variable`, which must be of type integer and must have the `Instance Dimension` as its sole dimension. The `Count Variable` contains the number of `Elements` that each `Feature` has.

This representation and its `Count Variable` are identifiable by the presence of an attribute, `sample_dimension`, found on the `Count Variable`, which names the `Sample Dimension` being counted.

Requirement 44 /req/CF-netCDF/ ContiguousRaggedArray: Any `Contiguous Ragged Array` dataset **shall** have a `Count Variable`.

Requirement 45 /req/CF-netCDF/ ContiguousRaggedArraySampleDimension: Any `Count Variable` **shall** define the attribute `sample_dimension`.

In addition, the attribute `sample_dimension` names the `Sample Dimension` being counted.

7.9.7 Indexed ragged array representation

The `Indexed Ragged Array` representation stores the features interleaved along the `Sample Dimension` in the `Feature Collection` data variable. The canonical use case for this representation is the storage of real-time data streams that contain reports from many sources.

In this representation, the `Dataset` contains an `Index Variable`, which must be of type integer, and must have the `Sample Dimension` as its single dimension.

The `Index Variable` contains the zero-based index of the `Feature` to which each `Element` belongs.

This representation and its `Index Variable` are identifiable by the presence of an attribute, `instance_dimension` which names the dimension of the `Instance Variables` (i.e. the `Instance Dimensions`).

Requirement 46 /req/CF-netCDF/IndexedRaggedArray: Any `Indexed Ragged Array` dataset **shall** have an `Index Variable`.

In addition, the `Index Variable` **must** be of type `integer`.

In addition, the `Index Variable` **must** the `Sample Dimension` as its single dimension.

Requirement 47 /req/CF-netCDF/IndexedRaggedArrayInstanceDimension: Any `Index Variable` **shall** define the attribute `instance_dimension`.

In addition, the attribute `instance_dimension` names the `Instance Dimension` characterizing the `Instance Variables` of the dataset

7.9.8 FeatureType attribute

The global attribute, `feature_type`, is required for all `Discrete Sampling Geometry` representations.

Requirement 48 /req/CF-netCDF/Feature_type: Any `Ragged Array` and Multidimensional Array dataset **shall** define an attribute `feature_type`.

In addition, the attribute `feature_type` **must** be: `"point"|"timeSeries"|trajectory|profile|timeSeriesProfile|trajectoryProfile`.

7.9.9 Coordinates and metadata

Every `Element` of every `Feature` must be unambiguously associated with its space and time coordinates and with the feature that contains it. The `coordinates` attribute must be attached to every `Feature Collection` data variable to indicate the spatiotemporal coordinate variables that are needed to geo-locate the data.

Requirement 49 /req/CF-netCDF/FeatureCollectionCoordinates: Any `Feature Collection` data variable **shall** define an attribute `coordinates`.

The `cf_role` attribute of `Instance Variable` identifies the roles of variables that identify features in discrete sampling geometries.

Requirement 50 /req/CF-netCDF/Cf_role: The attribute `cf_role` **must** be: `"timeseries_id"|"profile_id"|"trajectory_id"`.

7.9.10 Missing Data

`Auxiliary Coordinate Variables` (spatial and time), characterizing `Feature Collection`, must contain missing values to indicate a void in data storage in the file but must not have missing data for any other reason.

Requirement 51 /req/CF-netCDF/MissingData: `Auxiliary Coordinate Variables` characterizing `Feature Collection` **shall** define the attribute `missing_values`.

7.9.11 Examples

Annotated examples of `Discrete Sampling Geometries` are in Appendix A9 of [16].

8 CF-netCDF Mapping to ISO Coverage Model

8.1 ISO Coverages and Grid Coverages

According to ISO 19123 [4], a coverage is a *feature* (e.g. a `Feature Collection` instance) that acts as a *function* to return values from its *range* for any direct position within its *spatial, temporal or spatiotemporal domain.*

The spatiotemporal domain of a discrete coverage may be a regular or semi-regular tessellation of the extent of the coverage. Point sets (multi-points coverages) and other sets of non-conterminous geometric objects do not form tessellations.

Grid coverages employ a systematic tessellation of the domain. The principal advantage of such tessellations is that they support a sequential enumeration of the elements of the domain, which makes data storage and access more efficient. The tessellation may represent how the data were acquired or how they were computed in a model. The domain of a grid coverage is a set of grid points [4].

8.2 Dimensionality Challenges

To explicitly map the CF-netCDF array data model (e.g. meteo-ocean multi-dimensional observation and physical model outputs) to the ISO Coverage data model, there is a need to address structural and semantics differences, applying the appropriate constraints and, hence, performing a mediation process.

Points to consider include:

- CF-netCDF data model supports datasets characterized by multiple domains (e.g. more than one coordinate system is defined for a dataset), whereas an ISO coverage is characterized by a single coordinate system.
- CF-netCDF data model supports datasets characterized by arbitrary multi-dimensional domains (e.g. collections of spatiotemporal coverages), whereas an ISO coverage domain is either 2-D (space), 3-D (2D + vertical dimension or 2D + time), 4-D (2D + vertical dimension + time).
- Most commonly, the domain axes of a CF-netCDF dataset coincide with reference system axes. However, CF-netCDF allows arbitrary domain shapes, i.e. domain axes ordering. Thus, it is possible to have a variable v1 defined on a <x, y, t, z> domain and a variable v2 defined on a <z, x, t, y> domain. Since there is a fixed enumeration of allowed compound CRSs in ISO coverages, the transformation of such generalized domain coordinates to ISO CRS coordinates may not be an affine transformation. In other words, mapping CF-netCDF domains to ISO (geo)rectified domains may require axes reshaping and reordering.

In this document, a two steps approach is adopted to address these mapping issues: a first step consists in defining appropriate profiles for both data models. The second step deals with defining a set of mapping constraint rules, here expressed as conformance requirements.

8.3 CF-netCDF model mapping to ISO Coverage types

Clearly, CF-netCDF datasets encode discrete observations and measurements –i.e. *discrete coverages.*

CF-netCDF datasets (array oriented datasets) are mainly gridded and multi-point data. Thus, this specification will deal with multi-dimensional gridded point and multi-point data –i.e. *gridded point coverages* and *multi-point coverages*.

8.3.1 CF-netCDF and Continuous coverages

In most cases, a continuous coverage is also associated with a discrete coverage that provides a set of control values to be used as a basis for evaluating the continuous coverage. Evaluation of the continuous coverage at other direct positions is done by interpolating between the geometry value pairs of the control set. This often depends upon additional geometric objects constructed from those in the control set; these additional objects are typically of higher topological dimension than the control objects.

In ISO 19123, such objects are called "geometry value objects". A geometry value object is a geometric object associated with a set of geometry value pairs that provide the control for constructing the geometric object and for evaluating the coverage at direct positions within the geometric object.

A common example of geometry value object is represented by quadrilateral grid cell whose vertices are represented by four grid points (i.e. the set of geometry value pairs).

In the netCDF domain, the continuous quadrilateral grid coverage type is associated to a discrete grid point coverage type by sharing the same geometry grid and matrix values; the two coverage subclasses share the `GridValueMatrix` object and the derived `GridPointValuePair` objects. The real difference consists in the realization of the `locate()` operation, which is inherited from the Coverage super-type. Therefore, "the principal use of discrete point coverages is to provide a basis for continuous coverage functions, where the evaluation of the continuous coverage function is accomplished by interpolation between the points of the discrete point coverage".

The same approach is applied in the case of a multi-point coverage (Discrete Point Coverage in ISO 19123). *The principal use of discrete point coverages is to provide a basis for continuous coverage functions, where the evaluation of the continuous coverage function is accomplished by interpolation between the points of the discrete point coverage. Most interpolation algorithms depend upon a structured pattern of spatial relationships between the points. This requires either that the points in the spatial domain of the discrete point coverage be arranged in a regular way, or that the spatial domain of the continuous coverage be partitioned in a regular way in relation to the points of the discrete point coverage. Grid coverages employ the first method; Thiessen polygon and TIN coverages employ the second* [4].

In the case of netCDF data, the interpolation methods specified by the ISO continuous coverage classes do not apply in general. In fact, in most cases, any scientifically realistic interpolation depends on the physics of the situation as well as the geometry. Hence, any realistic interpolation is actually data dependent.

Therefore the netCDF data types don't implement the evaluation operation using interpolation methods.

They are mapped to the ISO discrete coverages because they actually represent sampled points in a continuous space where the intermediate values depend on the solution to physics-based equations that depend on the values of the range data.

8.3.2 Mapping rules

The high-level abstract mapping model is depicted in **Figure 7**.

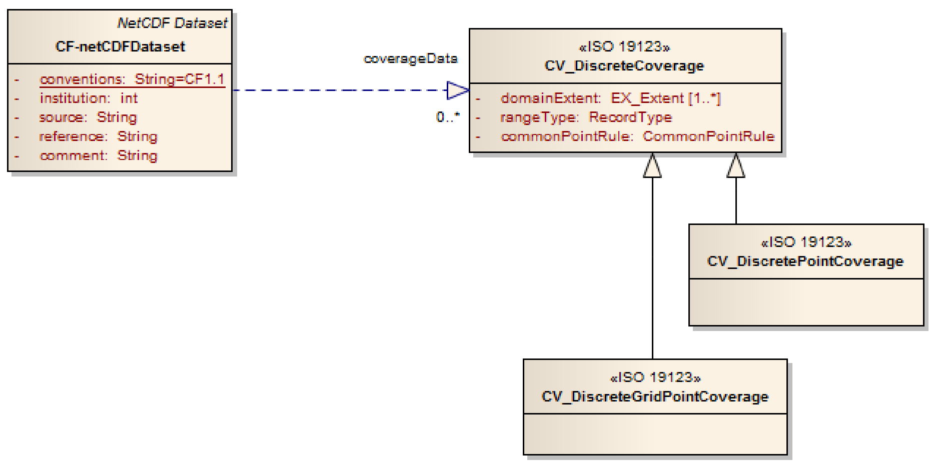

Figure 7. CF-netCDF dataset mapping to ISO coverage types

As far as the CF-netCDF and ISO19123 models are concerned, the following profiling constraints are introduced.

For the present specification, any group of CF-netCDF data variables that share the same set of spatial/temporal coordinate variables (e.g. `Feature Collection` objects) can be mapped to a single discrete coverage, `ISO19123:CV_DiscreteCoverage`.

For the present specification, the following discrete coverage sub-types are considered: `ISO19123:CV_DiscreteGridPointCoverage` and `ISO19123:CV_DiscretePointCoverage`.

Requirement 52 /req/CF-netCDF/discreteGridPointCoverage: Any group of CF-netCDF data variables that share the same set of spatial/temporal coordinate variables **shall** realize `ISO19123:CV_DiscreteCoverage` sub-types:

Copyright © 2011 Open Geospatial Consortium

`ISO19123:CV_DiscreteGridPointCoverage` and `ISO19123:CV_DiscretePointCoverage`.

In addition, the `ISO19123:CV_CoverageFunction` shall not define any interpolation method.

Figure 8 depicts the `ISO19123:CV_DiscretePointCoverage` data model considered for this specification.

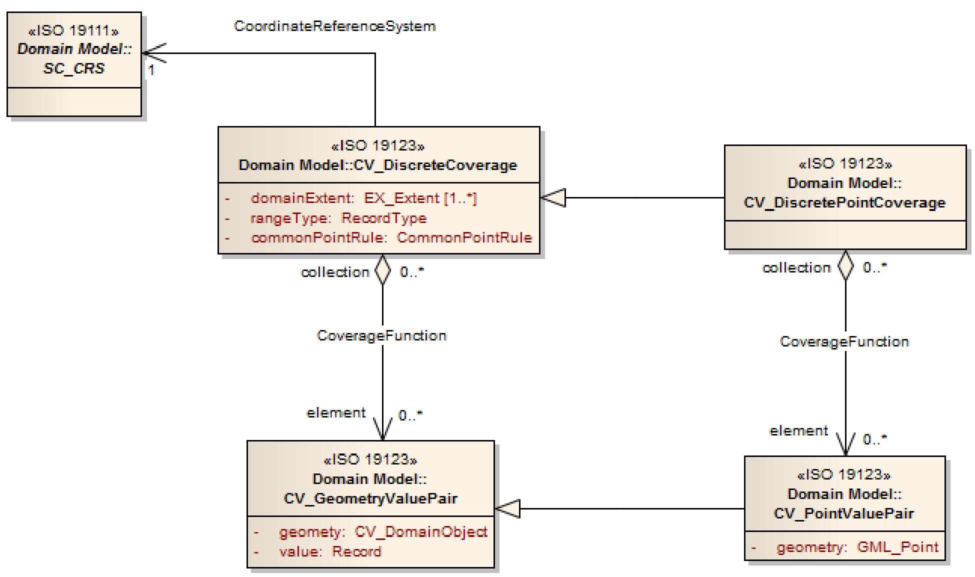

Figure 8. ISO19123:CV_DiscretePointCoverage data model

Figure 9 depicts the `ISO19123:CV_DiscreteGridPointCoverage` data model considered for this specification.

Copyright © 2010 Open Geospatial Consortium

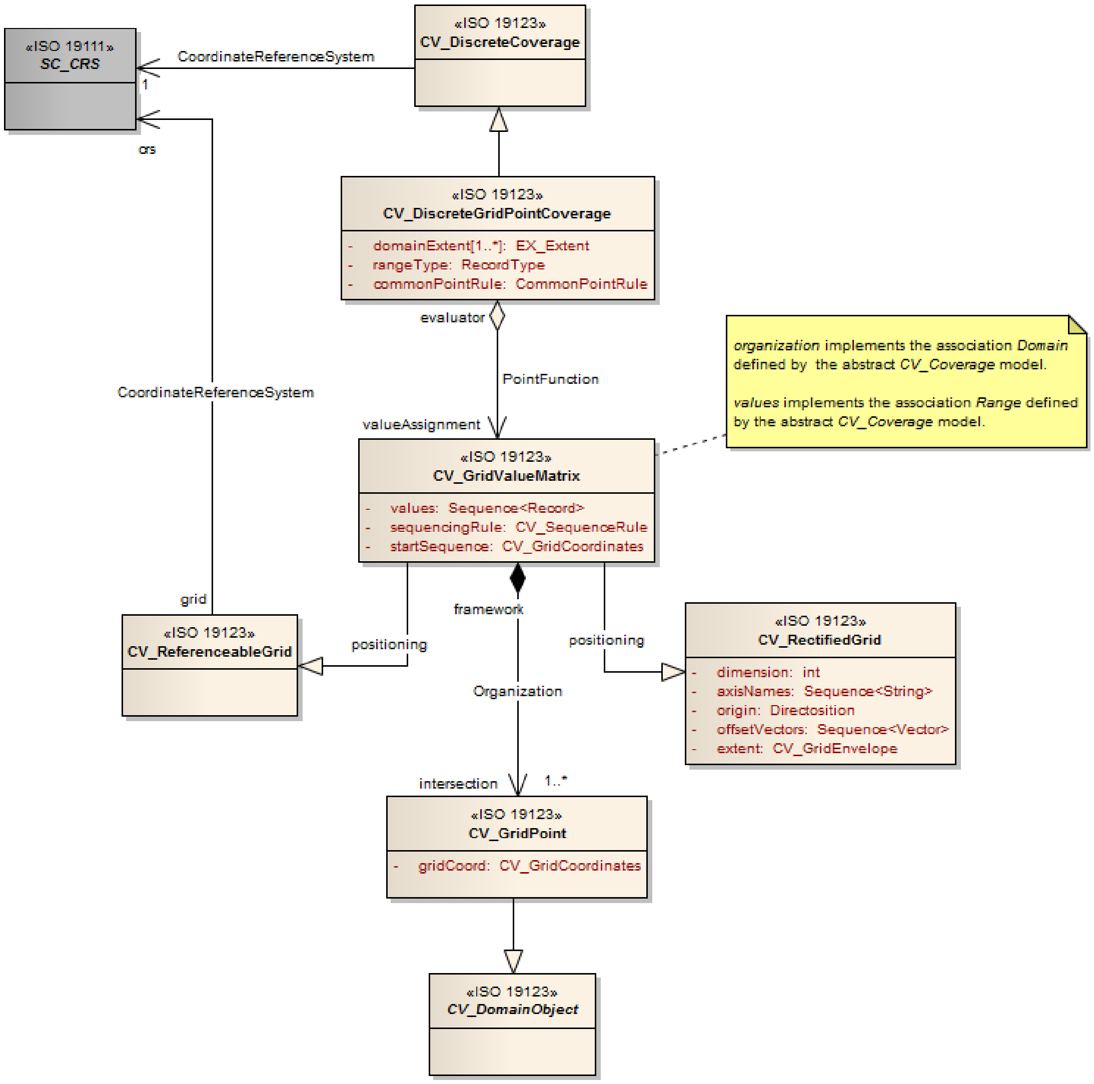

Figure 9. ISO19123:CV_DiscreteGridPointCoverage data model

The abstract mapping between `CF-netCDF` data and `ISO19123:CV_DiscreteCoverage` is showed in **Figure 10.**

Copyright © 2011 Open Geospatial Consortium

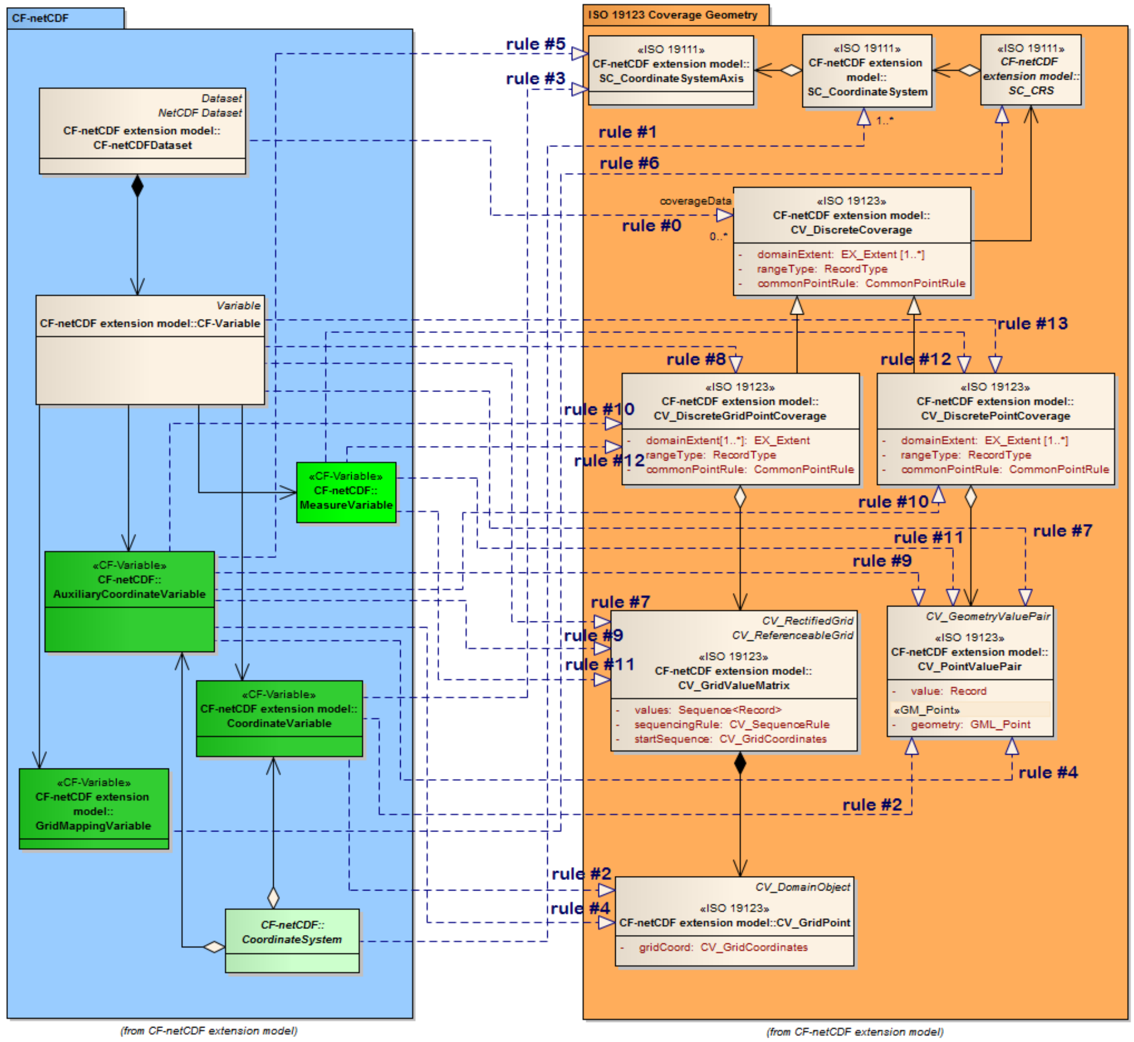

Figure 10. CF-netCDF dataset mapping ISO19123:CV_DiscreteCoverage

8.3.2.1 Coordinate System

A set of spatial/temporal `Coordinate Variables` realizes the domain of the coverage, whose geometry is represented by either a single grid (`ISO19123:CV_Grid`) or subtype of it.

Requirement 53 /req/CF-netCDF/CoverageDomain: the set of spatial/temporal `Coordinate Variables`, shared by CF-netCDF data `Variables`, **shall** realize an `ISO19123:CV_DiscreteGridPointCoverage.CV_GridValueMatrix.CV_GridPoint.gridCoord` or a subtype of it;

NOTE

The domain of each coverage may be described by the extent of the related coordinate axis variables.

CF-netCDF dataset Coordinate System is the composition of the set of coordinate variables / auxiliary coordinate variables commonly shared by CF-netCDF data variables. The coordinate variables / auxiliary coordinate variables realize the coordinate axes.

CF Auxiliary Coordinate Variable are used for encoding non standard spatial/temporal dimension (e.g. non monotonically crescent axes).

Requirement 54 /req/CF-netCDF/DatasetCS: the CF-netCDF Coordinate System type, which is comprised of the set of spatial/temporal coordinate variables shared by CF-netCDF data variables, **shall** realize ISO19123:CV_DiscreteCoverage.SC_CRS.SC_CoordinateSystem or a subtype of it (i.e. ISO19123:CV_DiscreteGridPointCoverage.SC_CRS.SC_CoordinateSystem or ISO19123:CV_DiscretePointCoverage.SC_CRS.SC_CoordinateSystem).

Additionally, the spatial/temporal Coordinate Variables and its subtypes (i.e. Vertical Coordinate, Latitude Coordinate, Longitude Coordinate, Time Coordinate elements), and the Auxiliary Coordinate Variables **shall** realize the related set of SC_CRS.SC_CoordinateSystem.SC_CoordinateSystemAxis

The grid mapping and/or projection information maps to the ISO1911:CRS and the associated ISO1911:CoordinateSystem (e.g. the units of the coordinate system).

Requirement 55 /req/CF-netCDF/DatasetCRS: the Grid Mapping Variable projection information characterizing the set of spatial/temporal coordinate variables, shared by CF-netCDF data variables, **shall** realize ISO19123:CV_DiscreteCoverage.SC_CRS.SC_CoordinateSystem.type or a subtype of it.

NOTES

To model "engineering" gridded datasets –i.e. datasets which are referred to a local defined coordinate system- the CRS is an application defined (i.e. engineering) CRS, characterized by an "engineering" datum (see ISO 19111). The associated Coordinate System has the origin and offset attributes coincident with the data grid Cartesian system origin and axes versors (i.e. unit vectors).

A Scalar Coordinate Variable is a coordinate variable that contains coordinate data. Functionally, it is equivalent to either a size one coordinate variable or a size one auxiliary coordinate variable

8.3.2.2 Coverage function

The data variables in a CF-netCDF dataset make up a discrete coverage *range* with each variable being a separate range field.

Requirement 56 /req/CF-netCDF/DatasetRange: the CF-netCDF data `Variables` sharing the common set of spatial/temporal `Coordinate Variables`, **shall** realize one of the following objects:
(a) `ISO19123:CV_DiscreteGridPointCoverage.CV_GridValuesMatrix.values` or a subtype of it;
(b) `ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.value`

Additionally, the CF `Variable` data values generate the `CV_GeometryValuePair` record value(s): (a) `CV_GridValueMatrix.values.record` entry (i.e. `AttributeName, Any`); (b) `CV_PointValuePair.value.record` entry (i.e. `AttributeName, Any`).

NOTE

The attribute *values: Sequence <Record>* shall be a sequence of N feature attribute value records where N is the number of grid points (or multi-point instances) within the section of the grid (or multi-point domain) specified by *extent* .

The discrete coverage *range* is a list of records with an attribute for every related CF-netCDF variable sharing a common set of spatial/temporal coordinate variables.

Requirement 57 /req/CF-netCDF/DatasetRangeType: the CF-netCDF data `Variables` sharing a common set of spatial/temporal coordinate variables/auxiliary coordinate variables, **shall** realize one of the following objects:
(a) `CV_DiscreteGridPointCoverage.RangeType.AttributesType` (i.e. `AttributeName, TypeName`) or a subtype of it;
(b) `CV_DiscretePointCoverage.RangeType.AttributesType` (i.e. `AttributeName, TypeName`) or a subtype of it.

CF-netCDF dataset may be characterized by arbitrary multi-dimensional domains, whereas an ISO coverage domain is either 2-D (space), 3-D (2D + vertical dimension or 2D + time), 4-D (2D + vertical dimension + time). Thus, the extra dimensions (called parametric dimensions and described by the `CF Auxiliary Coordinate Variable` entity) must be treated as variables realizing the coverage data *range* (i.e. CF_Variable).

Requirement 58 /req/CF-netCDF/DatasetParametricRange: the CF-netCDF `Auxiliary Coordinate Variables` (part of the dataset shared coordinate system) which are not space or time dimension, **shall** realize one of the following objects:
(a) `ISO19123:CV_DiscreteGridPointCoverage.CV_GridValuesMatrix.values` or a subtype of it;
(b) `ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.value`

Requirement 59 /req/CF-netCDF/DatasetParametricRangeType: the CF-netCDF `Auxiliary Coordinate Variables` (part of the dataset shared coordinate system) which are not space or time dimension, **shall** realize one of the following objects:
(a) `CV_DiscreteGridPointCoverage.RangeType.AttributesType` (i.e. `AttributeName, TypeName`) or a subtype of it;
(b) `CV_DiscretePointCoverage.RangeType.AttributesType` (i.e. `AttributeName, TypeName`).

A CF-netCDF cell `Measure Variable` may be associated with a variable to indicate extra information about the spatial properties of a variable's grid cells (or point cells).

`Measure Variable` data values must be treated as a *range* dimension of coverage data.

Requirement 60 /req/CF-netCDF/MeasureVariableMapping#1: any CF-netCDF `Measure Variable` (characterising a CF `Variable`) **shall** realize one of the following objects:
(a) `ISO19123:CV_DiscreteGridPointCoverage.CV_GridValuesMatrix.values` or a subtype of it;
(b) `ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.value`

Requirement 61 /req/CF-netCDF/MeasureVariableMapping#2: any CF-netCDF `Measure Variable` (characterising a CF `Variable`) **shall** realize one of the following objects:
(a) `CV_DiscreteGridPointCoverage.RangeType.AttributesType` (i.e. `AttributeName, TypeName`) or a subtype of it;
(b) `CV_DiscretePointCoverage.RangeType.AttributesType` (i.e. `AttributeName, TypeName`).

NOTE:

A `Scalar Coordinate Variable` is a coordinate variable that contains coordinate data. Functionally, it is equivalent to either a size one coordinate variable or a size one auxiliary coordinate variable

8.3.2.3 CF Discrete Sampling Geometries

In keeping the previous mapping requirements, **Figure 11** depicts the `CF Discrete Sampling Geometries` mapping to `ISO19123:DiscreteCoverage`.

Copyright © 2011 Open Geospatial Consortium

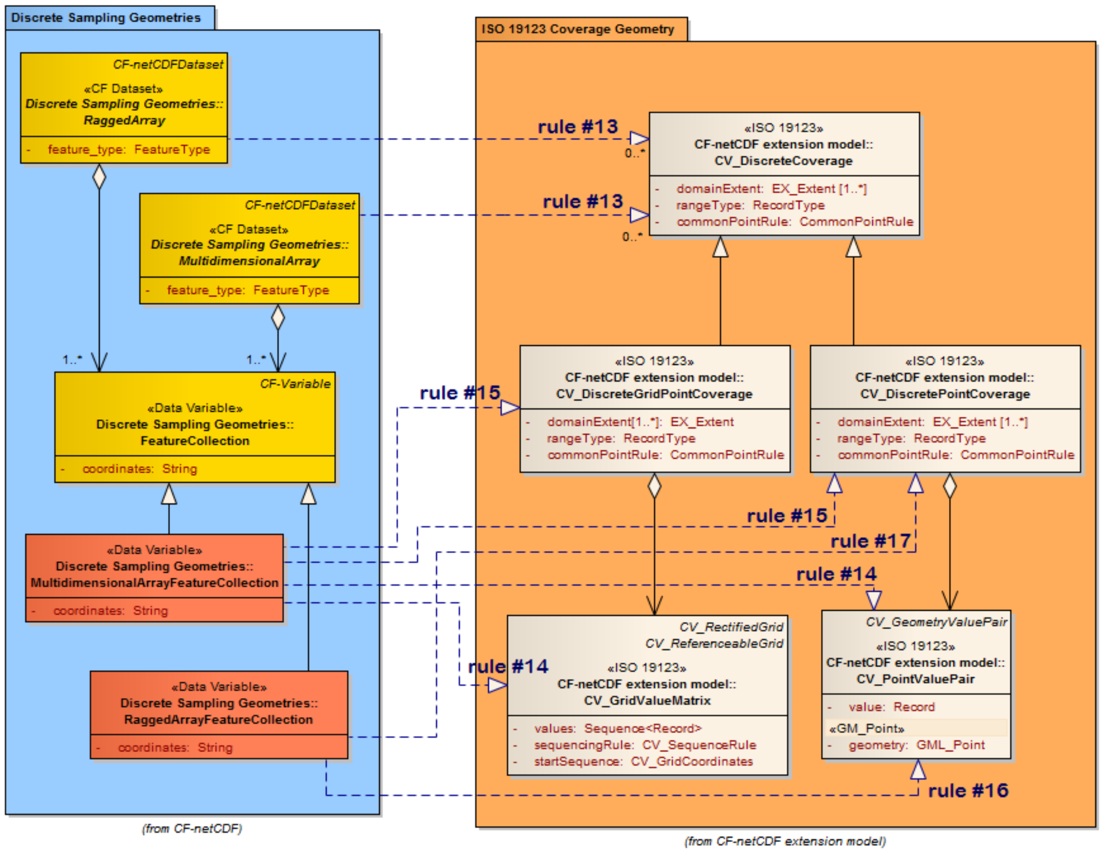

Figure 11 - CF-netCDF Discrete Sampling Geometries mapping ISO19123: DiscreteCoverage

CF `RaggedArray` and `MultidimensionalArray` entities are CF dataset sub-types; hence, they generate one or more `ISO19123:CV_DiscreteCoverage` instances.

Requirement 62 /req/CF-netCDF/DiscreteSamplingGeometriesMapping#1: Any CF-netCDF `Ragged Array` **shall** realize `ISO19123:CV_DiscreteCoverage` sub-type `ISO19123:CV_DiscreteGridPointCoverage`

In addition, the `ISO19123:CV_CoverageFunction` shall not define any interpolation method.

Requirement 63 /req/CF-netCDF/DiscreteSamplingGeometriesMapping#2: Any CF-netCDF `Multidimensional Array` **shall** realize `ISO19123:CV_DiscreteCoverage` sub-types: `ISO19123:CV_DiscreteGridPointCoverage` or `ISO19123:CV_DiscretePointCoverage`.

In addition, the `ISO19123:CV_CoverageFunction` shall not define any interpolation method.

A CF-netCDF `Feature Collection` is a CF `Variable`; hence, it generates an `ISO19123:CV_CoverageFunction` instance. For CF `Multidimensional Array Feature Collection`, each feature instance is allocated the identical amount of storage space. Both *orthogonal multidimensional array* and *incomplete multidimensional array* representations may encode feature instances in the collection which have identical coordinates along the element axis of the features. Therefore, it is possible to map those arrays on either `ISO19123:CV_DiscreteGridPointCoverage.CV_GridValuesMatrix` or `ISO19123:CV_DiscretePointCoverage.CV_PointValuePair`.

`Ragged Array Feature Collection` comprises features which are not characterized by identical coordinates along the element axis of the features. In these representations the instances of the individual features are stacked sequentially along the same array dimension as the elements of the features. Hence, they can be mapped (in an unambiguous way) only to `ISO19123:CV_DiscretePointCoverage.CV_PointValuePair`

Requirement 64 /req/CF-netCDF/MultidimensionalArrayFeatureCollectionMapping#1: Any CF-netCDF `Multidimensional Array Feature Collection` **shall** realize one of the following objects:
(a) `ISO19123:CV_DiscreteGridPointCoverage.CV_GridValuesMatrix.values` or a subtype of it;
(b) `ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.value`

Requirement 65 /req/CF-netCDF/MultidimensionalArrayFeatureCollectionMapping#2: Any CF-netCDF `Multidimensional Array Feature Collection` **shall** realize one of the following objects:
(a) `CV_DiscreteGridPointCoverage.RangeType.AttributesType` (i.e. `AttributeName, TypeName`) or a subtype of it;
(b) `CV_DiscretePointCoverage.RangeType.AttributesType` (i.e. `AttributeName, TypeName`) or a subtype of it.

Requirement 66 /req/CF-netCDF/RaggedArrayFeatureCollectionMapping#1: Any CF-netCDF `Multidimensional Array Feature Collection` **shall** realize an `ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.value` or a subtype of it.

Requirement 67 /req/CF-netCDF/RaggedArrayFeatureCollectionMapping#2: Any CF-netCDF `Ragged Array Feature Collection` **shall** realize an

`ISO10123:CV_DiscretePointCoverage.RangeType.AttributesType` (i.e. `AttributeName, TypeName`) or a subtype of it.

8.3.3 Mapping rules Summary

A summary of the mapping rules are reported in the **Table 2**, **Table 3** and **Table 4**.

Table 2. Summary of relationships between CF-netCDF3 and CV_DiscreteGridPointCoverage models: main packages

CF-netCDF concept	ISO Discrete Coverage concept	Mapping Cardinality (obligation)	Description and constraints	Rule #
CF Dataset	*CV_DiscreteGridPointCoverage* or *CV_DiscretePointCoverage*	1 to 0..n	Grouping the CF_Variables defined in a dataset by their CoordinateSystem, a *CV_DiscreteCoverage* may be defined for each group. The *CV_DiscreteCoverage* is realized as either a *CV_DiscreteGridPointCoverage* or a *CV_DiscretePointCoverage*. The association of groups and coverages may not be one-to-one, since the concept of coordinate system in CF-netCDF is wider than CRS (see table Table 4). Hence, some of the obtained coverages may be further grouped together. It is possible that a CoordinateSystem entity does not contain any axes allowed in coverage CRS (i.e. only parametric dimension axes); the associated variables would then originate no *CV_DiscreteCoverage* instance.	0
CF Variable *and sub-types related to a CoordinateSystem*	*CV_CoverageFunction*	1 to 1	The CF Variable belongs to a group of variables which is mappable to ISO *CV_DiscreteCoverage* See **Table 3**	---
Coordinate	`CS_CRS`	1 to 1	The CF-netCDF CoordinateSystem belongs to a group of variables which is	---

System			mappable to ISO *CV_DiscreteCoverage* See **Table 4**	
Discrete Sampling Geometry	*CV_DiscreteGridPointCoverage* or *CV_DiscretePointCoverage*	1 to 0..n	A `CF Discrete Sampling Geometry` is a type of `CF Dataset`	13
Feature Collection	*CV_CoverageFunction*	1 to 1	A `CF Feature Collection` is a type of `CF Variable` See **Table 5**	---

8.3.3.1 Coverage Function

Table 3. Summary of relationship between CF-netCDF and CV_DiscreteCoverage models: Coverage Function package

Copyright © 2011 Open Geospatial Consortium

CF-netCDF concept	ISO Discrete Coverage concept	Mapping Cardinality	Description and constraints	Rule #
Dataset CF Variable	ISO19123:CV_DiscreteGridPointCoverage.CV_GridValuesMatrix.values or ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.value	1 to 1	The set of CF Variables comprising the group that shares a common Coordinate System (whose axes are the set of spatial/temporal CF Coordinate Variables/Auxiliary Coordinate Variables). The CF Variable data values generate the range set record values, as either: CV_GridValueMatrix.values.record entry (i.e. AttributeName, Any) or CV_DiscretePointCoverage.CV_PointValuePair.value entry (i.e. AttributeName, Any)	7
Dataset CF Variable	CV_DiscreteGridPointCoverage.RangeType.AttributesType entry (i.e. AttributeName, TypeName) or ISO19123:CV_DiscretePointCoverage. RangeType.AttributesType entry (i.e. AttributeName, TypeName)	1 to 1	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/Auxiliary Coordinate Variable (defined by the shared Coordinate System). Thus, the CF Variable properties (i.e. *name* and *type*) realize the CV_DiscreteGridPointCoverage.RangeType.AttributesType entry (i.e. AttributeName, TypeName) or the analogous for multi-point coverages	8
Dataset Auxiliary Coordinate Variable	ISO19123:CV_DiscreteGridPointCoverage.CV_GridValuesMatrix.values or ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.value	1 to 1	A CF-netCDF Auxiliary Coordinate Variable may be used to realize a non spatial/temporal dimension (e.g. *pressure*, *density*, *salinity*). That is not allowed in a coverage CRS (i.e. parametric dimension axes). Thus, the auxiliary coordinate variable does not behave like a discrete coverage *domain* axis; on the contrary, it behaves like a discrete	9

			coverage *range* axis	

Dataset Auxiliary Coordinate Variable	`CV_DiscreteGridPointCoverage.RangeType.AttributesType entry (i.e. AttributeName, TypeName)`
	
	or
	
	`ISO19123:CV_DiscretePointCoverage. RangeType.AttributesType entry (i.e. AttributeName, TypeName)`
		1 to 1	A CF-netCDF `Auxiliary Coordinate Variable` may be used to realize a non spatial/temporal dimension (e.g. *pressure, density, salinity*).
			
			That is not allowed in a coverage CRS (i.e. parametric dimension axes). Thus, the auxiliary coordinate variable does not behave like a discrete coverage *domain* axis; on the contrary, it behaves like a discrete coverage *range* axis.
				10

Dataset MeasureVariable	`ISO19123:CV_DiscreteGridPointCoverage.CV_GridValuesMatrix.values`
	
	or
	
	`ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.value`
		1 to 1	A CF-netCDF cell `Measure Variable` is associated with a variable to indicate extra information about the spatial properties of a variable's grid cells (or point cells).
			
			A CF-netCDF cell `Measure Variable` must be treated as a range variable of the coverage.
				11

Dataset MeasureVariable	`CV_DiscreteGridPointCoverage.RangeType.AttributesType entry (i.e. AttributeName, TypeName)`
	
	or
	
	`ISO19123:CV_DiscretePointCoverage. RangeType.AttributesType entry (i.e. AttributeName, TypeName)`
		1 to 1	A CF-netCDF cell `Measure Variable` is associated with a variable to indicate extra information about the spatial properties of a variable's grid cells (or point cells).
			
			A CF-netCDF cell `Measure Variable` must be treated as a range variable of the coverage
				12

Note: A `Scalar Coordinate Variable` is a coordinate variable that contains coordinate data. Functionally, it is equivalent to either a size one coordinate variable or a size one auxiliary coordinate variable

8.3.3.2 Coverage Coordinate System and Grid Geometry

Table 4. Summary of relationship between the CF-netCDF and the DiscreteCoverage profile models: Coordinate System package

CF-netCDF concept	ISO Discrete Coverage concept	Mapping Cardinality	Description and constraints	Rule #

Coordinate System	`CV_DiscreteGridPointCoverage.SC_CRS.SC_CoordinateSystem`
	
	or
	
	`CV_DiscretePointCoverage.SC_CRS.SC_CoordinateSystem`
		1 to 1	Parametric coordinate systems are allowed in CF-netCDF but not in ISO CRS[1]. 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 CF-netCDF Coordinate System entity does not contain any axes allowed in ISO coverage CRS (i.e. only parametric dimension axes);
			
			Only spatial and temporal coordinates in a CF-netCDF Coordinate System become part of a coverage CRS, whereas parametric dimension axes are mapped to compound range set components.
				1

Spatial/temporal CF Coordinate Variable or subtypes (`Latitude Coordinate`, `Longitude Coordinate`, `Vertical Coordinate`, `Time Coordinate`)	`ISO19123:CV_DiscreteGridPointCoverage.CV_GridValueMatrix.CV_GridPoint.gridCoord` or sub-types (e.g. `ISO19123:CV_GridValueMatrix`) or `ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.geometry.GM_Point`	1 to 1	The set of monotonically crescent spatial/temporal coordinate variables (shared by a group of `CF Variables`) maps to the discrete coverage domain set, whose geometry is represented by either a single grid `ISO19123:CV_Grid` or a set of points, `GM_Points`	2
Spatial/temporal CF CoordinateVariable or subtypes (`Latitude Coordinate`, `Longitude`	`CV_DiscreteGridPointCoverage.SC_CRS.SC_CoordinateSystem.SC_CoordinateSystemAxis` or	1 to 1	A CF-netCDF spatial/temporal `Coordinate Variable` shared by a group of CF Variables.	3

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

Coordinate, Vertical Coordinate, Time Coordinate)	CV_DiscretePointCoverage.SC_CRS.SC_CoordinateSystem. SC_CoordinateSystemAxis			
CF Auxiliary Coordinate Variable	ISO19123:CV_DiscreteGridPointCoverage.CV_GridValueMatrix.CV_GridPoint.gridCoord or sub-types (e.g. ISO19123:CV_GridValueMatrix) or ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.geometry.GM_Point	1 to 1	A CF-netCDF Auxiliary Coordinate Variable realizes a "non-standard" or non monotonically crescent spatial/temporal dimension, which is shared by a group of CF Variables (e.g. CF Feature Collections). Coordinate data values map to the discrete coverage domain set, whose geometry is represented by either a single grid ISO19123:CV_Grid or a set of points, GM_Points	4
CF Auxiliary Coordinate Variable	CV_DiscreteGridPointCoverage.SC_CRS.SC_CoordinateSystem.SC_CoordinateSystemAxis or CV_DiscretePointCoverage.SC_CRS.SC_CoordinateSystem. SC_CoordinateSystemAxis	1 to 1	A CF-netCDF Auxiliary Coordinate Variable realizes a "non-standard" or non monotonically crescent spatial/temporal dimension, which is shared by a group of CF Variables (e.g. CF Feature Collections).	5
Grid Mapping Variable	ISO19123:CV_DiscreteGridPointCoverage.SC_CRS.SC_CoordinateSystem.type	1 to 1	CF Grid Mapping Variable is used as a container for attributes that define a specific grid mapping.	6

8.3.3.3 CF Discrete Sampling Geometries

Table 5. Summary of relationship between the CF-netCDF and the DiscreteCoverage profile models: Discrete Sampling Geometries package

CF-netCDF concept	ISO Discrete Coverage concept	Mapping Cardinality	Description and constraints	Rule #

CF Multidimensional Array Feature Collection	`ISO19123:CV_DiscreteGridPointCoverage.CV_GridValuesMatrix.values`
	
	or
	
	`ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.value`
		1 to 1	A `CF-netCDF Multidimensional Array Feature Collection` is a CF Variable whose feature instances are defined on a set of common element axes	14

CF Multidimensional Array Feature Collection	`CV_DiscreteGridPointCoverage.RangeType.AttributesType entry (i.e. AttributeName, TypeName)`
	
	or
	
	`ISO19123:CV_DiscretePointCoverage. RangeType.AttributesType entry (i.e. AttributeName, TypeName)`
		1 to 1	A `CF-netCDF Multidimensional Array Feature Collection` is a CF Variable whose feature instances have common dimensions along the set of common element axes.	15

CF Ragged Array Feature Collection	`ISO19123:CV_DiscretePointCoverage.CV_PointValuePair.value`	1 to 1	A `CF-netCDF Multidimensional Array Feature Collection` is a `CF Variable` whose feature instances have different dimensions along the common set of element axes.	16
CF Ragged Array Feature Collection	`ISO19123:CV_DiscretePointCoverage. RangeType.AttributesType entry (i.e. AttributeName, TypeName)`	1 to 1	A `CF-netCDF Multidimensional Array Feature Collection` is a `CF Variable` whose feature instances have different dimensions along the common set of element axes.	17

9 Bibliography

[1] OGC 10-090, NetCDF Core, version 1.0

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

[3] OGC 10-92, NetCDF Binary Encoding Extension Standard: NetCDF Classic and 64-bit Offset Format, version 1.0.0.

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

[5] UNIDATA netCDF home available at: http://www.unidata.ucar.edu/software/netcdf/

[6] NetCDF documents available at: http://www.unidata.ucar.edu/software/netcdf/docs/

[7] NetCDF Users Guide available at: http://www.unidata.ucar.edu/software/netcdf/docs/netcdf.html

[8] NetCDF Climate and Forecast (CF) Metadata Conventions, Version 1.5, 25 October, 2010; available at: http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.5/cf-conventions.pdf

[9] UDUNITS Software Package, UNIDATA Program Center of the University Corporation for Atmospheric Research (available at http://www.unidata.ucar.edu/software/udunits/).

[10] CF Conformance Requirements and Recommendations, available at: ?? http://cf-pcmdi.llnl.gov/conformance

[11] UNIDATA Best Practice recommendations on writing netCDF files, available at: http://www.unidata.ucar.edu/packages/netcdf/BestPractices.html

[12] NetCDF XML encoding (ncML) available at: http://www.unidata.ucar.edu/software/netcdf/ncml/

[13] 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

[14] S. Nativi and B, Domenico, "Enabling interoperability for Digital Earth: Coverage access services", International Journal of Digital Earth, Volume 2, Supplement 1, 2009, Pages: 79-104, Taylor & Francis ed., ISSN: 1753-8955 (electronic) 1753-8947 (paper), 2009.

[15] Mazzetti, P.; Nativi, S., J. Caron, "RESTful Implementation of Geospatial Services", International Journal of Digital Earth, Volume 2, Supplement 1, 2009, pages: 40-61, Taylor & Francis ed., ISSN: 1753-8955 (electronic) 1753-8947 (paper), 2009, 2009.

[16] NetCDF Climate and Forecast (CF) Metadata Conventions, Version 1.6, 4 May, 2011; available at: ???

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

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

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-DataModel

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

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