Open Geospatial Consortium Inc.

Date: 2008-03-19

Reference number of this OGC™ Document: 07-067r5

Version: 1.1.2, Version 1.1 Corrigendum 2 release

Category: OGC® Implementation Standard

Editors: Arliss Whiteside, John D. Evans

Web Coverage Service (WCS) Implementation Standard

Warning

This document is an OGC Standard. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OGC® Implementation Standard
Document subtype:	(none)
Document stage:	Approved Corrigendum
Document language:	English

Contents Page

Figures

Tables

Page

i.   Preface

This document defines a coorigendum for the OGC Web Coverage Service Interface Standard (WCS) version 1.1.2, or version 1.1 Corrigendum 2 release. This Corrigendum is based on WCS version 1.1.0 [OGC 06-083r8] and 1.1.1 [OGC 07-067r2]. It also builds on on OWS Common version 1.1.0 [OGC06-121r3].

ii.   Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 06-121r3], 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 contributed and supported work on this OGC standard.

- BAE SYSTEMS E&IS
- Australia Commonwealth Scientific and Industrial Research Organisation (CSIRO)
- CubeWerx Inc.
- George Mason University
- Jacobs University Bremen / rasdaman GmBH
- IONIC SOFTWARE s.a.
- U.S. National Aeronautics and Space Administration (NASA)
- U.S. National Geospatial-Intelligence Agency (NGA)
- Oracle Corp.
- PCI Geomatics

iv.   Document contributors

OGC's Web Coverage Service 1.1 Revision Working Group members are listed below.

Name	Organization
Baumann, Peter	Jacobs University Bremen / rasdaman GmBH
Cox, Simon	CSIRO
Di, Liping	George Mason University
Donéa, Luc	IONIC SOFTWARE
Evans, John	NASA / GST, Inc
Herring, John	Oracle Corporation
Keens, Steven	PCI Geomatics
Roswell, Charles	NGA
Vretanos, Panagiotis	CubeWerx, Inc.
Whiteside, Arliss	BAE Systems E&IS

v. Revision history

This WCS version 1.1.2 is Corrigendum 2 of WCS 1.1, as documented in [OGC 07-066r3], further correcting WCS 1.1.1 (Corrigendum 1 of WCS 1.1). Corrigendum 2 fixes minor inconsistencies, spelling mistakes, and adds some minor clarifications.

WCS 1.1.1 was Corrigendum 1 of WCS 1.1, as documented in [OGC 07-066r2]. Briefly, the changes were:

a) Fixed minor inconsistencies, spelling mistakes, and added some minor clarifications

b) Aligned with OWS-Common version 1.1.0

WCS 1.1.0 was almost a complete rewrite of WCS 1.0.0, with many editorial changes and detailed streamlining of schema files resulting from OWS Common conformance. However, the two versions of the specification share the same overall structure. WCS 1.1.0 introduced (only) the following functional changes:

a) Use of GridCRS in coverage descriptions and requests
(See 9.3.1.2, 10.2.1.1, 10.2.2, 10.3.4, 10.3.6, and Annexes F and G)

b) Hierarchical coverage descriptions (See 8.3.4.4)

c) Multiple fields per coverage
(See 9.3.1.3, 9.3.2.1, 10.2.1.2, 10.2.2.1, 10.2.2.2, and Tables 18, 26-29)

d) XML and (optionally) asynchronous GetCoverage responses (See 10.3.11)

e) Different approach to coverage formats; reliance on encoding profiles (See 9.3.2.2)

Many of the changes were the direct or indirect result of addressing written change proposals, as summarized in OGC 06-116.

vi. Changes to the OGC Abstract Specification

OGC® Abstract Specification Topic 6: The Coverage Type and its Subtypes (which includes ISO 19123) requires the following change to accommodate the technical contents of this document:

Spatial interpolation needs to be a property of each field in the range of a coverage, not a property of the coverage or of its range as a whole. Currently, the interpolationType is an attribute of the CV_ContinuousQuadrilateralGridCoverage class, which is associated to one CV_GridValuesMatrix. This CV_GridValuesMatrix contains a sequence of values of Record type, where each Record can contain multiple "fields". However, different fields may require the use of different interpolation types; and more than one interpolation method may be associated with a single field.

vii.   Future work

Future versions of this WCS specification are expected to consider various expansions of the abilities specified herein, some adding abilities that were deliberately not included in Version 1.1.0. Some of the possible expansions thus include:

a) Expand supported coverage types beyond grid coverages.

b) Expand ability to retrieve elevation subset of a coverage, beyond current regularly spaced (grid) elevations.

c) Expand ability to retrieve spatial subset of a coverage, beyond current regularly spaced (grid) positions.

d) Expand ability to store coverage outputs, for later retrieval.

e) Add WSDL description of SOAP encodings

f) Add ability to Retrieve multiple coverages in one GetCoverage operation request

g) Add optional interface operation(s) to input new coverages and update existing coverages offered by a WCS server

h) Expand Annex A to specify abstract test suites.

Foreword

This version 1.1.1 of the Web Coverage Service (WCS) Standard supersedes the previous version 1.1.0 [OGC 06-083r8]. Technical changes from the version 1.0 include building on the OGC Web Services Common Specification [OGC 06-121r3] and a substantially revised Capabilities schema; new schemas and syntax for operation requests (GetCoverage, DescribeCoverage); and integration with GML 3.1. The changes in WCS 1.1.1 from WCS 1.1.0 are summarized in [OGC 07-066].

This document includes eight annexes; Annexes A, E, F, and H are normative, and Annexes B, C, D, and G are informative.

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-varying phenomena.

A WCS provides access to potentially detailed and rich sets of geospatial information, in forms that are useful for client-side rendering, multi-valued coverages, and input into scientific models and other clients. The WCS may be compared to the OGC Web Map Service (WMS) and the Web Feature Service (WFS); like them it allows clients to choose portions of a server's information holdings based on spatial constraints and other criteria.

Unlike the WMS [OGC 06-042], which portrays spatial data to return static maps (rendered as pictures by the server), the Web Coverage Service provides available data together with their detailed descriptions; defines a rich syntax for requests against these data; and returns data with its original semantics (instead of pictures) which may be interpreted, extrapolated, etc. – and not just portrayed.

Unlike WFS [OGC 04-094], which returns discrete geospatial features, the Web Coverage Service returns coverages representing space-varying phenomena that relate a spatio-temporal domain to a (possibly multidimensional) range of properties.

The Web Coverage Service provides three operations: *GetCapabilities*, *DescribeCoverage,* and *GetCoverage*. The *GetCapabilities* operation returns an XML document describing the service and brief descriptions of the coverages that clients may request. Clients would generally run the *GetCapabilities* operation and cache its result for use throughout a session, or reuse it for multiple sessions. When the *GetCapabilities* operation does not return such descriptions, then equivalent information must be available from a separate source, such as an image catalog.

The *DescribeCoverage* operation lets clients request a full description of one or more coverages served by a particular WCS server. The server responds with an XML document that fully describes the identified coverages.

The *GetCoverage* operation is normally run after *GetCapabilities* and *DescribeCoverage* operation responses have shown what requests are allowed and what data are available. The *GetCoverage* operation returns a coverage (that is, values or properties of a set of geographic locations), encoded in a well-known coverage format. Its syntax and semantics bear some resemblance to the WMS *GetMap* and WFS *GetFeature* requests, but several extensions support the retrieval of coverages rather than static maps or discrete features.

Web Coverage Service (WCS) Implementation Specification

1   Scope

This document specifies how a Web Coverage Service (WCS) describes and delivers multi-dimensional coverage data over the World Wide Web. This version of the Web Coverage Service is limited to describing and requesting grid (or "simple") coverages.

Grid coverages have a domain comprised of regularly spaced locations along 0, 1, 2, or 3 axes of a spatial coordinate reference system. Their domain may also have a time dimension, which may be regularly or irregularly spaced. A coverage defines, at each location in the domain, a set of fields that may be scalar-valued (such as elevation), or vector-valued (such as brightness values in different parts of the electromagnetic spectrum). These fields (and their values) are known as the range of the coverage.

The WCS interface, while limited in this version to regular grid coverages, is designed to extend in future versions to other coverage types defined in OGC Abstract Specification Topic 6, "The Coverage Type" [OGC 00-106].

2   Conformance

Conformance with this specification shall be checked using all the relevant tests specified in Annex A (normative).

3   Normative references

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.

IETF RFC 2387, E. Levinson, *The MIME Multipart/Related Content-type*

IETF RFC 2396, T. Berners-Lee, R. Fielding, U.C. Irvine, L. Masinter: Uniform Resource Identifiers (URI): Generic Syntax. IETF, RFC 2396, August 1998

ISO 19123:2005, *Geographic Information — Coverage Geometry and Functions*

OGC 00-106, *The OpenGIS™ Abstract Specification Topic 6: The Coverage Type and its Subtypes*

OGC 04-021r3, OGC™ Catalogue Services Specification

OGC 05-094r1, *GML 3.1.1 CRS support profile*

OGC 05-095r1, *GML 3.1.1 common CRSs profile*

OGC 05-096r1, *GML 3.1.1 grid CRSs profile*

OGC 05-103, *Geographic information — Spatial referencing by coordinates* (ISO/DIS 19111)

OGC 06-042, *Web Map Service version 1.3*

OGC 06-121r3, *OGC Web Services Common Specification, version 1.1.0*

W3C Note, J. Barton, S. Thatte, H. Nielsen, *SOAP Messages With Attachments*, December 11, 2000

NOTE   The OGC Web Services Common Specification (OGC 06-121r3) contains a list of normative references that are also applicable to this Implementation Specification.

In addition to this document, this specification includes normative XML Schema files. These are posted online at the URL http://schemas.opengis.net/wcs/1.1.2/. These XML Schema files are also bundled with the present document. In the event of a discrepancy between the bundled and online versions of the XML Schema files, the online files shall be considered authoritative.

4   Terms and definitions

For the purposes of this specification, the definitions given in OWS Common Implementation Specification [OGC 06-121r3] and in OGC Abstract Specification Topic 2: Spatial referencing by coordinates [OGC 05-103] and shall 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

4.2
georectified grid
grid having regular spacing in a projected or geographic coordinate reference system (CRS)

NOTE   A grid for which there is a linear relationship between the grid coordinates and those of a projected or geographic coordinate reference system.

4.3
georeferenced grid
grid that is not georectified, but is associated with (one or more) coordinate transformations which relate the image or engineering CRS to a projected or geographic CRS

NOTE   These coordinate transformations are usually not affine or simple, and are usually empirically determined. (Synonym: *georeferenceable*).

4.4
offered coverage
coverage that is available from a WCS server

NOTE   Part or all of an offered coverage can be retrieved by a client using the GetCoverage operation. An offered coverage may be physically stored in one or more parts, locally or remotely, or may be a virtual coverage that can be assembled by the server when needed. The CoverageDescription can be retrieved by a client using the DescribeCoverage operation. The CoverageSummary may be retrievable by a client using the GetCapabilities operation.

4.5

output coverage

coverage that is provided in the response to a GetCoverage operation

NOTE   An output coverage is usually a part of an offered coverage and may be resampled into a different Coordinate Reference System (CRS).

4.6

union of bounding boxes

combined area of multiple bounding boxes

NOTE   This combined area will not be rectangular.

5   Conventions

5.1   Symbols (and abbreviated terms)

Most of the abbreviated terms listed in Subclause 5.1 of the OWS Common Implementation Specification [OGC 06-121r3] also apply to this document.

5.2   UML notation

All the diagrams that appear in this specification are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of the OGC Web Services Common Implementation Specification [OGC 06-121r3].

5.3   Used parts of other documents

This document uses significant parts of document [OGC 06-121r3]. To reduce the need to refer to that document, this document copies some of those parts with small modifications. To indicate those parts to readers of this document, the largely copied parts are shown with a gray background (15%).

5.4   Platform-neutral and platform-specific specifications

As specified in Subclause 5.4 of OWS Common [OGC 06-121r3], this document includes both platform-neutral and platform-specific specifications.

EXAMPLES 1   Platform-neutral specifications are contained in Subclauses 8.2.1, 8.3.1, 8.3.2, 8.3.4.1 to 8.3.4.5, 9.2.1, 9.3.1, 10.2.1, and 10.3.11.

EXAMPLES 2   Platform-specific specifications for KVP encoding are contained in Subclauses 8.2.2, 9.2.2, and 10.2.2.

EXAMPLES 3   Platform-specific specifications for XML encoding are contained in Subclauses 8.2.3, 8.3.4.7, 8.3.5, 9.2.3, 9.3.3, 10.2.3, and 10.3.13.

5.5 Data dictionary tables

The UML model data dictionary is specified herein in a series of tables. The contents of the columns in these tables are described in Subclause 5.5 of [OGC 06-121r3]. The contents of these data dictionary tables are normative, including any table footnotes.

6 WCS overview

The specified Web Coverage Service (WCS) interface supports electronic retrieval of geospatial data as "coverages" – that is, digital geospatial information representing space-varying phenomena. A WCS provides access to potentially detailed and rich sets of geospatial information, in forms that are useful for client-side rendering and input into scientific models and other clients.

The WCS interface specifies three operations that may be requested by a WCS client and performed by a WCS server:

a) GetCapabilities (required implementation by servers) – This operation allows a client to request the service metadata (or Capabilities) document. This XML document describes the abilities of the specific server implementation, usually including brief descriptions of the coverages available on the server. This operation also supports negotiation of the specification version being used for client-server interactions. Clients would generally request the *GetCapabilities* operation and cache its result for use throughout a session, or reuse it for multiple sessions. When the *GetCapabilities* operation does not return descriptions of its available coverages, that information must be available from a separate source, such as an image catalog.

b) DescribeCoverage (required implementation by servers) – This operation allows a client to request full descriptions of one or more coverages served by a particular WCS server. The server responds with an XML document that fully describes the identified coverages.

c) GetCoverage (required implementation by servers) – This operation allows a client to request a coverage comprised of selected range properties at a selected set of geographic locations. The server extracts the response data from the selected coverage, and encodes it in a known coverage format. The *GetCoverage* operation is normally run after *GetCapabilities* and *DescribeCoverage* operation responses have shown what requests are allowed and what data are available.

Many of these interface aspects that are common with other OGC Web Services (OWSs) are specified in the OGC Web Services Common Specification [OGC 06-121r3]. These common aspects are normatively referenced herein, instead of being repeated in this specification.

Figure 1 is a simple UML class diagram summarizing the WCS interface. This class diagram shows that the WebCoverageServer interface class inherits the getCapabilities operation from the OGCWebService interface class, and adds the getCoverage and describeCoverage operations. (This capitalization of names uses the OGC/ISO profile of UML.) A more complete UML model of the WCS interface is provided in Annex D (informative).
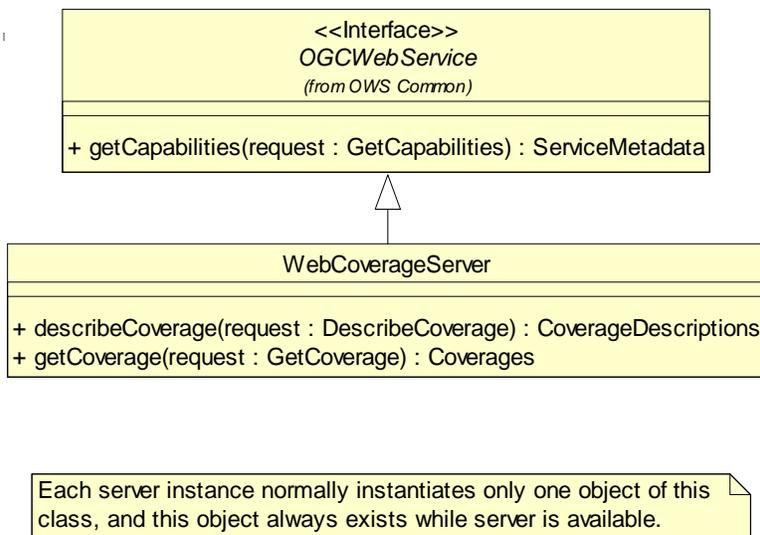
Figure 1 — WCS interface UML class diagram

NOTE   In this UML diagram, the request and response for each operation is shown as a single parameter that is a data structure containing multiple lower-level parameters, which are discussed in subsequent clauses. The UML classes modeling these data structures are included in the complete UML model in Annex D.

Each of the three operations is described in more detail in subsequent clauses.

7   Shared aspects

7.1   Introduction

This clause specifies aspects of the WCS behavior that are independent of particular operations or are shared by several operations.

7.2   Operation request encoding

All WCS operation requests may use HTTP GET with KVP encoding or use HTTP POST with either XML or SOAP encoding, as specified in Clause 11 of [OGC 06-121r3].

7.3   XML Schema notes

The data type of many parameters is specified as "Character String type, not empty". In the XML Schemas specified, these parameters are encoded with the xsd:string type, which does NOT require that these strings not be empty.

7.4   Operation response encoding and reserved characters

All operation responses shall comply with the XML syntax. In addition, several parameter values provided in *CoverageSummary* and *CoverageDescription* operation responses shall be usable literally ("as-is") in subsequent *DescribeCoverage* and *GetCoverage* operation requests to the specific server, encoded in KVP or XML. Therefore, in stating values for these parameters, servers shall avoid, encode, or escape any "special" characters that could be interpreted as part of either the XML markup ('<' '>' '--' '&' *etc.*) or the KVP syntax ('?' '=' '&' *etc.*) of subsequent requests.

In particular, to support XML requests, operation responses shall avoid special characters, or encode or escape them using character or numeric entities (e.g., ">" or ">"in lieu of ">") as defined in XML [W3C 2004]. To support KVP requests, operation responses shall avoid special characters, or URL-encode them (e.g., "%3F" for "?") as defined in [RFC 2396].

7.5 CRS definitions and references

7.5.1 WCS use of CRSs

WCS operation requests and responses frequently include references to a defined Coordinate Reference System (CRS), and sometimes include a CRS definition. A defined CRS is required for spatial referencing by coordinates, and is a coordinate system which is related to an object by a datum. This specification makes use of material in several other OGC documents that specify how to define and reference CRSs.

In many cases, WCS operation requests and responses transfer only references to defined CRSs, not the actual CRS definitions. Each referenced CRS shall be defined somewhere, but that definition may not be electronically accessible or encoded in XML. Nonetheless, the referenced definitions often need to be used by WCS server software.

7.5.2 CRS definitions

OGC Abstract Specification Topic 2 "*Geographic information — Spatial referencing by coordinates*" [OGC 05-103], which contains ISO/DIS 19111, specifies how to define many types of CRSs, including the Datums and Coordinate Systems that define a CRS. Clause 7 of that document specifies the following types of CRSs, all of which might be used with a WCS grid coverage:

a) GeodeticCRS (formerly GeographicCRS and GeocenticCRS)

b) ProjectedCRS

c) VerticalCRS

d) CompoundCRS

e) ImageCRS

f) DerivedCRS

g) EngineeringCRS

Clause 12 of GML 3.1.1 [OGC 03-105r1] specifies how to XML encode the definitions of all these types of CRSs, based on an older version of OGC Abstract Specification Topic 2 [OGC 04-046r3]. Three GML 3.1.1 profiles extract the parts of GML 3.1.1 that appear useful to a WCS, namely:

a) OGC 05-094r1, *GML 3.1.1 CRS support profile*, which is used by the following two profiles

b) OGC 05-095r1, *GML 3.1.1 common CRSs profile*, which covers GeographicCRS, ProjectedCRS, VerticalCRS, and CompoundCRS definitions

c) OGC 05-096r1, *GML 3.1.1 grid CRSs profile*, which covers ImageCRS and DerivedCRS definitions

Annex F in this document specifies how to XML encode a GridCRS definition, in a simplified manner that is loosely based on a DerivedCRS.

7.5.3 CRS references

Clause 10.3 of OWS Common [OGC 06-121r3] specifies how CRSs shall be referenced. A URL can be used to reference a definition that is available using this URL. When not in the same XML document, those definitions shall be electronically available using this URL. Alternately, a URN in the "ogc" URN namespace can be used, and often should be used, to reference a CRS definition identified in that "ogc" URN namespace.

The format of URNs in the "ogc" namespace is recommended in Clause 7 of OGC Best Practices Paper [OGC 06-023r1], *"Definition identifier URNs in OGC namespace".* Some recommended URN formats include parameter values that allow that URN to be used for an entire family of CRSs (such as different origins).

Numerous specific URN values for CRSs and components are currently specified in:

a) Clause 8 and Annex A of [OGC 06-023r1], "Definition identifier URNs in OGC namespace", for numerous CRSs and components

b) Clause 10 of OGC 05-096r1, GML 3.1.1 grid CRSs profile, for ImageCRSs, DerivedCRSs, and components (including URNs for most possible 2D ImageCRSs)

c) Subclause F.3 in this document, for three specific OperationMethods (expected to be referenced by most 2D GridCRS definitions)

7.5.4 ImageCRSs used in WCS

WCS 1.1 uses the specialized form of ImageCRSs specified in the GML 3.1.1 grid CRSs profile [OGC 05-096r1 with Corrigendum 1 OGC 06-111 and Corrigendum 2 OGC 06-146]. Each coordinate value in an ImageCRS is the index of a spatial position in the image, with units of Grid Spacings, see Subclause 10.9 of [OGC 05-096r1]. To refer to a recorded pixel value, each coordinate value will be an integer. However, to refer to a general position in an image, each coordinate value may include a fractional value to reference positions between recorded pixels.

NOTE 1   An ImageCRS is defined in OGC Abstract Specification Topic 2, which includes a copy of ISO 19111. That ImageCRS is specialized in the GML 3.1.1 grid CRSs profile. Such an ImageCRS has only spatial coordinate axes. A CRS for an image with both spatial and temporal axes can be defined as a CompoundCRS, as defined in OGC Abstract Specification Topic 2. However, no examples of this are now included in this document.

WCS 1.1 uses only ImageCRSs with the origin at the centre of the first recorded pixel, to be consistent with ISO 19123. Negative coordinate values shall not be used, since they would reference pixels that do not exist. Therefore, the ImageCRS identifier should be changed when the origin pixel is changed, from the origin of an offered coverage to the origin of an output coverage that is derived from that offered coverage.

NOTE 2   One possible way to generate a different ImageCRS Identifier for an output image that has a different origin than the offered coverage would be to extend the ImageCRS Identifier of the offered

coverage. This extension could encode the position of the output image's origin pixel, relative to the offered image's origin pixel. This relative origin identifier could contain one (positive) integer for each dimension of the ImageCRS. For example, if the offered coverage's ImageCRS is "urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:ABC123", the output coverage's ImageCRS might be "urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:ABC123DE:987:654".

When such an ImageCRS is referenced, the first recorded pixel shall be the first pixel in this ImageCRS. For a spatial 2D image, the first recorded pixel has the indices 0,0. For a spatial 3D image, the first recorded pixel has the indices 0,0,0. For a spatial 1D image, the first recorded pixel has the index 0.

In WCS 1.1, all 2D ImageCRSs should be referenced using the URN form specified in Subclause 10.2 of [OGC 05-096r1], namely "urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:TBD". Each specific ImageCRS shall be for the image, or image group, whose alphanumeric character string identifier is substituted for the "TBD" in this URN. When these URNs are used, there is no need to provide a definition for each specific ImageCRS, since the definition of this ImageCRS is completely specified in Clause 10 of [OGC 05-096r1], with the exception of the image identifier that is directly included in this URN.

7.5.5   GridCRSs used in WCS

WCS 1.1 uses the GridCRSs specified in Annex F of this document. Each coordinate value in a GridCRS is the index of a spatial position in the grid coverage, with units of Grid Spacings (like an ImageCRS). (Coordinate values in a GridCRS are NOT in the referenced GridBaseCRS.) To refer to a recorded pixel, each coordinate value will be an integer. However, to refer to a general position in a grid coverage, each coordinate value may include a fractional value to reference positions between recorded pixels.

The origin of a GridCRS may be at either of the following alternatives:

a)   The origin may be the first pixel recorded for that grid coverage (similar to an ImageCRS). In this case, negative coordinate values shall not be used, since they would reference pixels that do not exist. Also, the GridCRS definition identifier should be changed when its first pixel is changed, from the origin of an offered coverage to the origin of an output coverage that is derived from that offered coverage. Specifically, the GridOrigin value(s) should be changed.

b)   The origin may be located at any grid point in the specified grid. In this case, negative and positive coordinate index values may be used, since they reference pixel positions that are defined, but may not be recorded in the grid coverage. In this case, the GridCRS origin pixel (with 0 valued indices) may be anywhere in the pixels recorded, or may be anywhere outside the pixels recorded, and thus not recorded. In both cases, the same GridCRS definition may be used for grid coverages recorded with different first pixels, from the same or different original coverages or images. Also, the GridCRS definition identifier should be changed when any other part of the GridCRS definition is changed, either the GridOrigin, GridOffsets, or GridBaseCRS.

In WCS 1.1, the definition of each referenced GridCRS is always included in the same operation request or response. This is done since many GridCRSs will be unique. No URN form for referencing GridCRSs is currently specified, but probably should be

defined. Such a GridCRS may then be referenced as a SupportedCRS or in a BoundingBox, included in an operation request or response.

7.5.6   WCS use of GridCRSs

In WCS 1.1, a georectified image or coverage always uses a GridCRS, for both offered and output coverages. Each GridCRS specifies the origin and offsets between recorded coverage pixels, in the referenced GridBaseCRS. For a georectified image or coverage, the referenced GridBaseCRS shall be a ground space CRS, which may be a GeographicCRS, ProjectedCRS, VerticalCRS, or CompoundCRS that combines a VerticalCRS with a 2D GeographicCRS or ProjectedCRS.

NOTE 1   An ImageCRS is not used for a georectified image because a GridCRS provides more useful information. Specifically, that GridCRS specifies the location of the grid points in the referenced GridBaseCRS, and that information is often needed to usefully exploit this coverage. If used, an ImageCRS would provide only the identification of the image, allowing finding the image pixels. However, an ImageCRS requires that the origin be the first pixel recorded. Furthermore, an ImageCRS should only be used for an image, not for a more general grid coverage.

In WCS 1.1, an unrectified image may use either an ImageCRS or a GridCRS with its GridBaseCRS being an ImageCRS, for both offered and output coverages. Such a GridCRS may be used when an unrectified image was previously processed, to change the pixel spacing or origin. That GridCRS specifies the origin and offsets between the recorded coverage pixels, each in the referenced original ImageCRS before the known previous processing. If such previous processing is not known, an ImageCRS for the current image is used.

NOTE 2   A separate ImageCRS is not used for an unrectified image when a GridCRS is known and used, because that GridCRS allows providing more useful information. Also using an ImageCRS separate from the GridBaseCRS of that GridCRS would not provide needed additional information. The only information in a WCS 1.1 ImageCRS is the identification of the image, allowing finding the image pixels. For an offered coverage, that image identification is not needed separate from the Identifier in the CoverageDescription. For an output coverage, that image identification should be assigned by the WCS client.

When an unrectified image uses a GridCRS with its GridBaseCRS being an ImageCRS, that ImageCRS should be referenced using a URN, and no separate ImageCRS definition is required. The GridOrigin and GridOffsets parameters specify the pixel positions in this GridCRS relative to the referenced ImageCRS. For example, these parameters can specify different pixel spacings and origin, of the GridCRS relative to the referenced original ImageCRS (before known previous processing). If no previous processing is known, a GridCRS may be used with GridOrigin values of zero, and GridOffsets values that specify the same pixel spacings.

In WCS 1.1, no GridCRS should be used whose GridBaseCRS is another GridCRS, for either a georectified or unrectified image. This limitation is partially because WCS 1.1 provides no place to transfer the definition of a GridCRS that is used as the GridBaseCRS of another GridCRS which is for an offered image. A URL to a location where the GridCRS definition is stored might be used, except for difficulties in implementing such URL access.

7.5.7 Current WCS uses of CRSs

WCS 1.1 operation requests and responses frequently include references to a defined Coordinate Reference System (CRS), and sometimes include a CRS definition. Table 1 summarizes these CRS uses in this version of this OGC Standard. In Table 1:

a) In the left column, all the WCS 1.1 data structures and parameters that define or reference CRSs are listed in **Bold**. All of these are CRS references (with type URI) except for GridCRS (three times), which are GridCRS definitions.

b) In the left column, indented lists are used to identify which larger data structures and operation request or response contains each listed CRS reference or definition. The multiplicity of most data structures and parameters is indicated using UML notation.

c) In the other three columns, the allowed types of CRSs are listed for each CRS reference or definition, for each of three conditions based on the nature of the offered coverage. These conditions map to the use cases described in Annex G and listed in the table header.

d) In the second column, the allowed types of CRSs are listed for a (previously) georectified offered coverage.

e) In the third column, the allowed types of CRSs are listed for an unrectified offered coverage that is not (known by the WCS server to be) georeferenced. This column also applies if the unrectified offered coverage is georeferenced, but the WCS server is not requested to apply, or cannot apply, the (known) georeferencing coordnate transformation.

f) In the right column, the allowed types of CRSs are listed for an unrectified offered coverage that is georeferenced when the georeferencing coordinate transformation IS used by the GetCoverage operation to georectify that image. If the unrectified coverage is georeferenced but the georeferencing coordinate transformation is NOT used by the GetCoverage operation to georectify that image, then the third column applies except for including the georeferencing coordinate transformation with the output coverage.

Table 1 — WCS uses of CRSs

Operation (direction) Data structure name Data structure name	Allowed types of CRS, for offered coverages that are:		
	Georectified (Use cases 1, 2, 6, 7)	**Unrectified and unreferenced** (Use cases 3, 4, 5, 8)	**Unrectified and georeferenced used** (Use cases 9, 10)
GetCoverage (request) GetCoverage	N/A	N/A	N/A
DomainSubset [0..1] BoundingBox [1] **crs** [1]	See SupportedCRS in CoverageDescription	CRS of offered coverage [a]	See SupportedCRS in CoverageDescription
Output [1] **GridCRS** [0..1] (definition)	Specified GridCRS, or will use GridCRS of offered coverage	Specified GridCRS, or will use CRS of offered coverage [a]	Specified GridCRS
GridBaseCRS [1]	See SupportedCRS in CoverageDescription	ImageCRS of offered or original coverage	See SupportedCRS in CoverageDescription

GetCoverage (response) Coverage	N/A	N/A	N/A
BoundingBox [1] **crs** [1]	Reference to GridCRS below	GridCRS or ImageCRS of output coverage	Reference to GridCRS below
GridCRS [0..1] (definition)	GridCRS of output coverage	GridCRS of output coverage, when used	GridCRS of output coverage
GridBaseCRS [1]	See SupportedCRS in CoverageDescription	ImageCRS of offered coverage	See SupportedCRS in CoverageDescription
DescribeCoverage (response) CoverageDescription	N/A	N/A	N/A
SupportedCRS [0..*]	GeographicCRS or ProjectedCRS [b]	ImageCRS [c]	GeographicCRS or ProjectedCRS [b]
SpatialDomain [1]	N/A	N/A	N/A
BoundingBox [1..*] **crs** [1]	GridCRS of offered coverage [d]	CRS of offered coverage [a, d]	CRS of offered coverage [a, d]
GridCRS [0..1] (definition)	GridCRS of offered coverage	GridCRS of offered coverage, if used	GridCRS of offered coverage, if used
GridBaseCRS [1]	See SupportedCRS in CoverageDescription	ImageCRS	ImageCRS
ImageCRS [0..1]	(none)	ImageCRS of offered coverage, if used	ImageCRS of offered coverage, if used

GetCapabilities (response)
  Capabilities
    CoverageSummary
	N/A		

WGS84BoundingBox [0..*] **crs** [0..1]	WGS84 (a GeographicCRS)		
SupportedCRS [0..*]	See SupportedCRS in CoverageDescription		

a   The CRS of an unrectified offered coverage may be an ImageCRS or a GridCRS with a GridBaseCRS that is an ImageCRS.

b   In addition to a GeographicCRS or ProjectedCRS, other ground space CRSs may be referenced, including a VerticalCRS or a CompoundCRS that combines a VerticalCRS with a 2D GeographicCRS or ProjectedCRS. These CRS types are sometimes referred to as ground CRSs.

c   An ImageCRS for an unrectified offered image shall be listed as a SupportedCRS, so that it may be referenced as the GridBaseCRS of a GridCRS. This ImageCRS shall be the ImageCRS of that unrectified offered image, or the ImageCRS that is referenced as the GridBaseCRS of the GridCRS that is used by that unrectified offered image.

d   Followed by zero or more of any SupportedCRS listed in the CoverageDescription.

7.6   BoundingBox use in WCS

7.6.1   General terms

This specification uses the BoundingBox and WGS84BoundingBox data structures, specified in Subclause 10.2 of OWS Common [OGC 06-121r3], to express the minimum rectangular bounding region around (available or requested) coverage data. In a Bounding Box data structure, the LowerCorner defines the edges of the (available or requested) region in the directions of decreasing coordinate values in a CRS (normally the algebraic minimum coordinates), and the UpperCorner defines the edges in the directions of increasing coordinate values (usually the algebraic maximum coordinates). For continuous (interpolatable) grid coverages, these values shall be computed from the areas of all the (available or requested) coverage grid cells with all corners at recorded grid points.

NOTE 1   Inclusion of the areas of all grid cells is not necessarily the same as inclusion of all the grid points, when the grid coverage spans the antimeridian as discussed below.

In keeping with ISO 19123, WCS considers only grids whose points lie at the corners (not the centres) of the grid cells. This document thus primarily considers "continuous" grid coverages, not "discrete" grid coverages, as those terms are defined in ISO 19123. As illustrated in Figure 2 below, grid points (a, b, c, d) define a grid cell (U) and are surrounded by "sample spaces" (A, B, C, D) as those terms are used in Subclause 8.2.2 of ISO 19123.
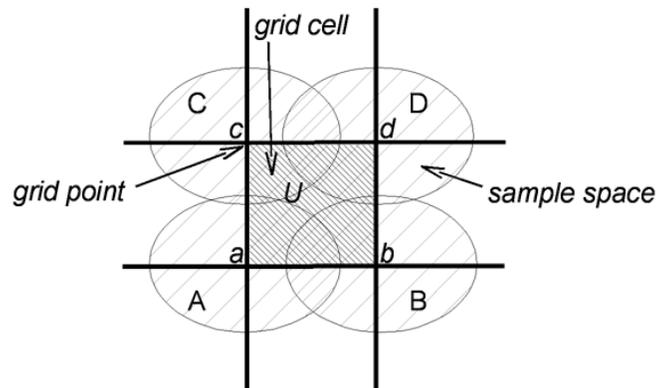
Figure 2 — Grid points, grid cells, and sample spaces

NOTE 2   For these sample spaces, this document avoids using terms such as "sample grid" or "pixels", although such terms are often used elsewhere.

The following subclauses clarify the use of such a bounding box near discontinuities in a geographic CRS, and the handling of grid points at the edges of the coverage.

7.6.2   Spanning the antimeridian of a geographic CRS

In a geographic CRS, longitude values are often defined with a finite extent (e.g., from -180 degrees to +180 degrees). The minimum and maximum longitude limits define a single line (on the ellipsoid, sphere, or cylinder), known as the antimeridian, across which longitude values are discontinuous: as this line is crossed, longitude changes abruptly (e.g., going West from a little more than -180 degrees to a little less than +180 degrees). This necessitates an "extended" interpretation of the Bounding Box definition:

> A Bounding Box defined in a geographic CRS (or a WGS84BoundingBox) whose LowerCorner longitude is greater than that of its UpperCorner shall describe a region that crosses the longitude discontinuity.

EXAMPLE   A bounding region encompassing the Pacific Ocean could be described by a BoundingBox with a positive LowerCorner longitude and a negative UpperCorner longitude.

NOTE   Users and implementers may avoid using the above "extended" interpretation of the Bounding Box by EITHER (a) using a CRS whose value discontinuity lies outside the minimum bounding region (e.g., a geographic CRS with a different prime meridian); OR (b) treating longitude as unbounded but periodic (using the XML Schema strings "-INF" and "INF" to denote negative and positive infinity). (See also Subclause 10.2.5 of OWS Common [OGC 06-121r3].)

If the data represented by the coverage is intended to be continuous (or interpolatable) across the antimeridian, then a server should perform spatial interpolation near and across the antimeridian just as it does elsewhere.

7.6.3   Treatment of edge grid points

The spatial extent of a grid coverage extends only as far as the outermost grid points contained in the bounding box. It does NOT include any area (partial or whole grid cells or sample spaces) beyond those grid points.

NOTE   This bounding box is NOT the extent sometimes considered, which also includes rectangular sample spaces (pixels) centered on the outermost grid points – as indicated in Subclause 7.3.3 6 of WMS 1.3 [OGC 04-042]. Such pixel extents are often poor approximations of the sensor physics / grid data collection process.

EXAMPLE   The extent of a global data set in WGS84 geographic decimal degrees with a grid spacing of 1.0 degree might be expressed as a WGS84BoundingBox with LowerCorner and UpperCorner longitudes of (respectively) -179.5 and +179.5, or -179.0 and +180.0, when that data set is not considered continuous (interpolatable) across the antimeridian at 180 degrees. When that data set is interpolatable across the antimeridian, the LowerCorner and UpperCorner longitudes (respectively) might be -180 and +180 (with redundant first and last columns of data), or –INF and INF.

8   GetCapabilities operation (mandatory)

8.1   Introduction

The mandatory GetCapabilities operation allows WCS clients to retrieve service metadata from a WCS server. The response to a GetCapabilities request shall be an XML document containing service metadata about the server, usually including summary information about the data collections from which coverages may be requested. This clause specifies KVP and XML encoding of a GetCapabilities request and the XML document that a WCS server shall return to describe its capabilities.

8.2   GetCapabilities operation request

8.2.1   Request contents

The GetCapabilities operation request shall be as specified in Subclauses 7.2.2 through 7.2.4 of [OGC 06-121r3]. The value of the "service" parameter shall be "WCS". The allowed set of service metadata (or Capabilities) XML document section names and meanings shall be as specified in Tables 6 and 10 in Subclauses 7.3.3 and 7.4.2 of [OGC 06-121r3].

The "Multiplicity and use" column in Table 3 of [OGC 06-121r3] specifies the optionality of each listed parameter in the GetCapabilities operation request. Table 2 specifies the implementation of those parameters by WCS clients and servers.

13

Table 2 — Implementation of parameters in GetCapabilities operation request

Names	Multiplicity	Server implementation	Client implementation
service	One (mandatory)	Each parameter shall be implemented by all servers [a]	Each parameter shall be implemented by all clients, using specified value
request	One (mandatory)		
Accept Versions	Zero or one (optional)	Shall be implemented by all servers	Should be implemented by all software clients, using allowed values
Sections	Zero or one (optional)	Each parameter should be implemented by each server [a, b]	Each parameter may be implemented by each client [c]
update Sequence	Zero or one (optional)		
Accept Formats	Zero or one (optional)		

a   Server shall check that each value received is an allowed value.

b   When this parameter is not implemented or not received, server shall provide the specified default response. When implemented and received, server shall provide specified response.

c   When this parameter is not provided, client shall expect the specified default response. When provided, client shall allow default or specified response.

All WCS servers shall implement HTTP GET transfer of the GetCapabilities operation request, using KVP encoding. Servers may also implement HTTP POST transfer of the GetCapabilities operation request, using XML encoding only.

8.2.2   KVP encoding

The KVP encoding of a WCS GetCapabilities operation request shall be as specified in Table 5 in Subclause 7.2.2 of [OGC 06-121r3].

8.2.3   XML encoding

The XML Schema fragment for encoding a WCS GetCapabilities operation request extends ows:GetCapabilitiesType in owsGetCapabilities.xsd from [OGC 06-121r3]:

```
<element name="GetCapabilities">
  <annotation>
    <documentation>Request to a WCS to perform the GetCapabilities
operation. This operation allows a client to retrieve a Capabilities XML
document providing metadata for the specific WCS server. In this XML
encoding, no "request" parameter is included, since the element name
specifies the specific operation. </documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="ows:GetCapabilitiesType">
        <sequence>
        <attribute name="service" type="ows:ServiceType" use="required"
fixed="WCS"/>
      </extension>
    </complexContent>
  </complexType>
</element>
```

8.2.4   SOAP encoding

WCS servers may implement SOAP version 1.2 transfer of the GetCapabilities operation request as specified in Annex E, using the XML encoding specified above.

8.3   GetCapabilities operation response

8.3.1   Normal response

The normal response to a GetCapabilities operation request shall be a service metadata (or Capabilities) document as specified in Subclause 7.4.2 of OWS common [OGC 06-121r3], and shown in the UML class diagram in Figure 3. The WCS service metadata document shall contain the four optional sections specified in Table 3. Depending on the values in the Sections parameter of the GetCapabilities operation request, any combination of these sections may be requested and returned.
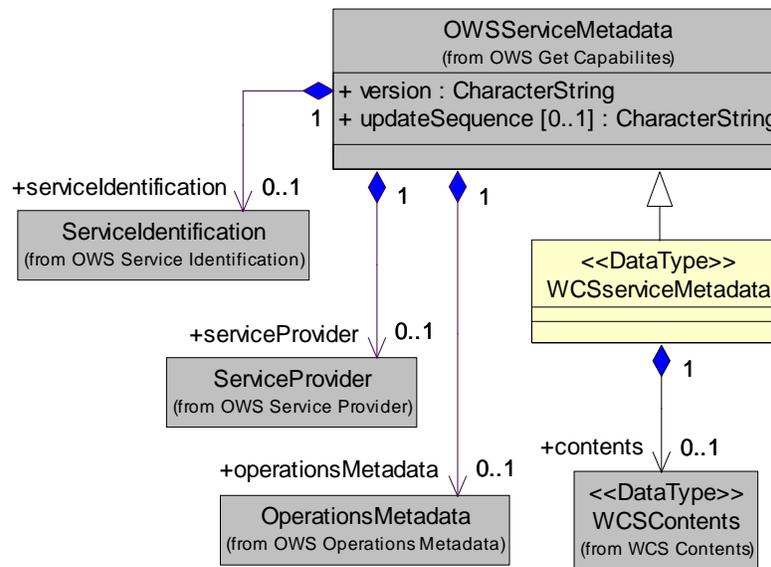
Figure 3 — WCS Service Metadata UML class diagram

NOTE   To reduce the need for readers to refer to another document, the first three sections listed below are copied from Table 10 in Subclause 7.4.2 of [OGC 06-121r3].

Table 3 — Section name values and contents

Section name	Contents
ServiceIdentification	Metadata about this specific server. The schema of this section shall be the same as for all OWSs, as specified in Subclause 7.4.4 and owsServiceIdentification.xsd of [OGC 06-121r3].
ServiceProvider	Metadata about the organization operating this server. The schema of this section shall be the same for all OWSs, as specified in Subclauses 7.4.5 and owsServiceProvider.xsd of [OGC 06-121r3].
OperationsMetadata	Metadata about the operations specified by this service and implemented by this server, including the URLs for operation requests. The contents and organization of this section shall be the same as for all OWSs, as specified in Subclause 7.4.6 and owsOperationsMetadata.xsd of [OGC 06-121r3].
Contents	Metadata about the data served by this server. For the WCS, this section shall contain or reference brief descriptions of the coverages available from this server, as specified in Subclause 8.3.3 below.

In addition to these sections, each service metadata document shall include the mandatory "version" parameter. Each service metadata document may include the optional "updateSequence" parameter specified in Table 7 in Subclause 7.4.1 of [OGC 06-121r3].

8.3.2   OperationsMetadata section contents

8.3.2.1 Overview

The OperationsMetadata section provides metadata about the operations specified by this service and implemented by this server, including the URLs for operation requests. The contents and organization of this section shall be as specified in Subclauses 7.4.5, 7.4.8, and owsOperationsMetadata.xsd of [OGC 06-121r3].

The mandatory values of various (XML) attributes shall be as specified in Table 4. In Table 4, the "Attribute name" column uses dot-separator notation to identify parts of a parent item. The "Attribute value" column references an operation parameter, in this case an operation name, and the meaning of including that value is listed in the right column.

Table 4 — Required values of OperationsMetadata section attributes

Attribute name	Attribute value	Meaning of attribute value
Operation.name	GetCapabilities	The GetCapabilities operation is implemented by this server.
	DescribeCoverage	The DescribeCoverage operation is implemented by this server.
	GetCoverage	The GetCoverage operation is implemented by this server.

In addition, there are many optional values of "name" attribute and "value" elements within optional Parameter and Constraint elements in the OperationsMetadata section, which may be included when considered useful. These Parameter and Constraint elements are for recording the domains of various parameters and quantities.

EXAMPLE 1   The domain of the "exceptionCode" parameter could record all the codes implemented for each operation by that specific server. Similarly, each of the GetCapabilities operation optional request parameters might have its domain recorded.

EXAMPLE 2   The domain of the "Sections" parameter in the GetCapabilities operation request could record all the sections implemented by that specific server.

8.3.2.2 Indicating allowed encodings

All WCS servers shall specify the encodings that may be sent using HTTP POST transfer of operation requests. Specifically, an ows:Constraint element shall be included, with "PostEncoding" as the value of the "name" attribute and specifying different allowed values for each allowed encoding:

a) The value "SOAP" shall indicate that SOAP encoding is allowed, as specified in Annex E.

b) The value "XML" shall indicate that XML encoding is allowed (without SOAP message encapsulation).

NOTE 1   The value "KVP" is not used, since this WCS standard does not allow KVP encodings of operation requests to be transferred using HTTP POST.

When the HTTP POST connect point URL is different for different encodings of the operation requests, this ows:Constraint element shall be included in each Post element. When the connect point URL is the same for all encodings of all operation requests, this ows:Constraint element should be included in the OperationsMetadata element.

NOTE 2   No ows:Constraint element with the "name" parameter (and XML attribute) value of "InputFormat" is used, since no WCS operation uses an InputFormat parameter (for data).

8.3.2.3 Indicating support for the "store" parameter

All WCS servers shall specify whether or not that server implements the "store" parameter in GetCoverage operation requests, doing this in an ows:Operation element for the GetCoverage operation. When this WCS server implements the "store" parameter, this shall be specified by including an ows:Parameter element with "store" as the "name" attribute value, and both "True" and "False" as allowed Values. When this WCS server does not implement the "store" parameter, this shall be specified by including an ows:Parameter element with "store" as the "name" attribute value, and only "False" as an allowed Value.

8.3.3   Contents section

8.3.3.1 Overview

The Contents section shall contain and/or reference summary metadata about the coverages available. It shall be structured as shown in the UML class diagram in Figure 4. The attributes and associations of the two new classes shall include the data structure and parameters specified in Table 5. Table 6 lists the contents of the CoverageSummary data structure, allowing these summaries to be arranged in a hierarchy.
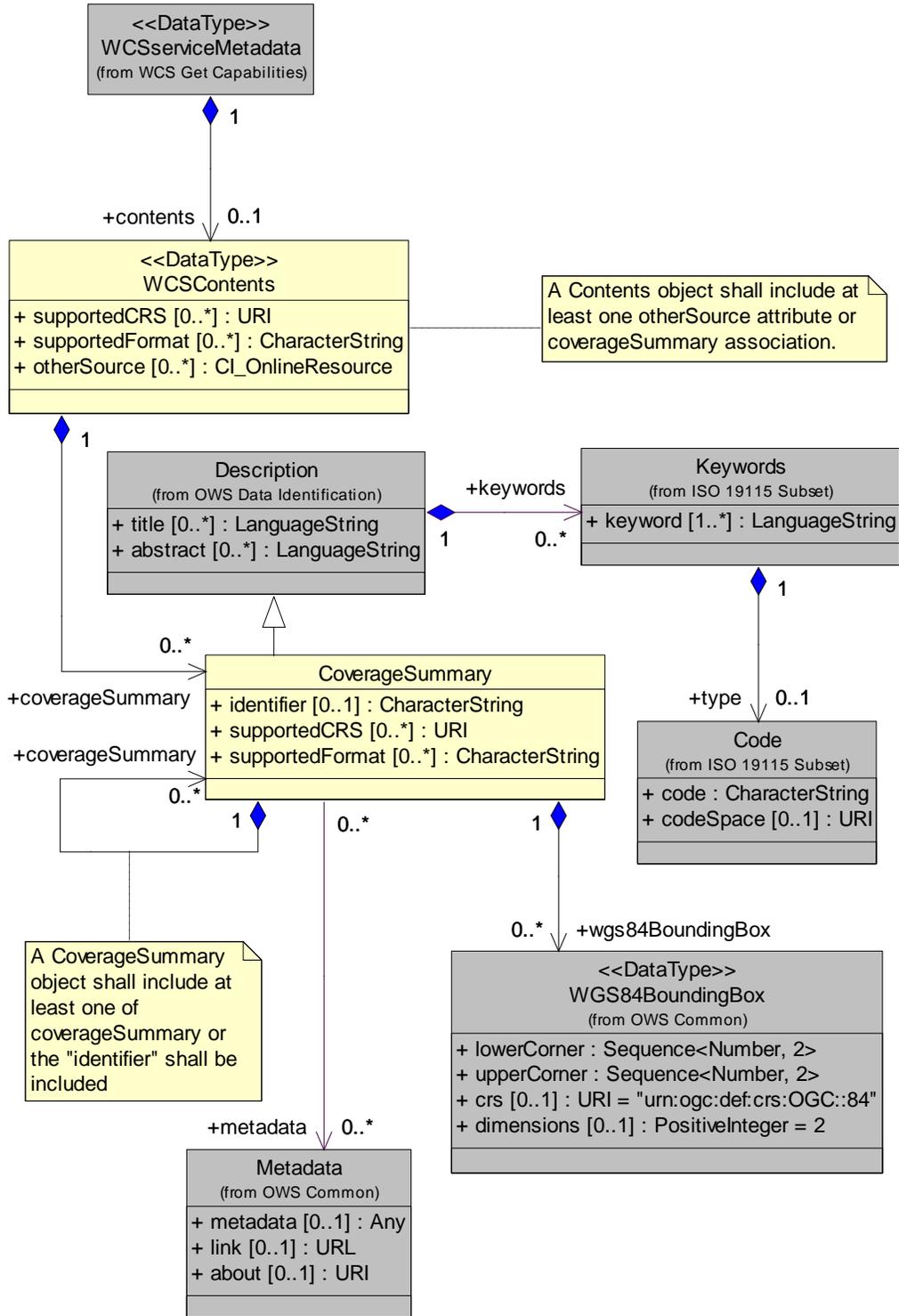
Figure 4 — Contents UML class diagram

Table 5 — Parts of Contents section

Name [a]	Definition	Data type	Multiplicity and use
coverageSummary CoverageSummary	Brief metadata describing a top-level coverage available from this WCS server	CoverageSummary data structure, see Table 6	Zero or more (optional) [b] One for each coverage available, unless OtherSource is used
supportedCRS SupportedCRS	Reference to CRS in which GetCoverage operation response may be expressed.	URI	Zero or more (optional) Include one for each CRS supported [c, d ,f]
supportedFormat SupportedFormat	Identifier of format in which GetCoverage operation response may be encoded	CharacterString type	Zero or more (optional) Include one for each format supported [c, e, f]
otherSource OtherSource	Reference to another source of contents metadata	CI_OnlineResource class in ISO 19115	Zero or more (optional) Include when useful or no CoverageSummaries included

a   See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.

b   Shall be included unless the OtherSource is included and all this metadata is available from that source.

c   There may be a dependency of SupportedCRS on SupportedFormat, as described in Sublause 10.3.5.

d   This list of SupportedCRSs shall be the union of all of the supported CRSs in all of the nested CoverageSummaries.

e   This list of SupportedFormats shall be the union of all of the supported formats in all of the nested CoverageSummaries.

f   Servers should implement this list since it reduces the work clients need to do to determine that they can interoperate with the server. The union is used instead of the intersection to indicate all that can be useed for an offered coverage.

NOTE   The CoverageSummary and OtherSource parts of the Contents section are not mutually exclusive, although we expect that only one will often be included. This document specifies no meaning to inclusion of both parts, with or without overlapping information.

8.3.3.2 CoverageSummary

The CoverageSummary provides a summary-level description of coverage data available from a WCS server. Clients may be able to formulate simple GetCoverage requests based only on this information. However, in order to make more finely tuned GetCoverage requests, clients will usually need to obtain further details about a particular coverage, using the DescribeCoverage operation (see Clause 9). Table 6 lists the contents of the CoverageSummary data structure, allowing these summaries to be arranged in a hierarchy.

NOTE   To reduce the need for readers to refer to another document, all except the CoverageSummary parameters listed below are largely copied from Table 32 in Subclause 10.6.1 of [OGC 06-121r3].

Table 6 — CoverageSummary data structure

Names [a]	Definition	Data type	Multiplicity and use
(Description) [b]	Description of a coverage	Description data structure, see Subclause 10.6.3 of [OGC 06-121r3]	One (mandatory) [b]
metadata Metadata	More metadata about this coverage	Metadata data structure, see Table 23 of [OGC 06-121r3]	Zero or more (optional) Include when useful
wgs84BoundingBox WGS84BoundingBox	Minimum bounding rectangle surrounding coverage, using WGS 84 CRS with decimal degrees and longitude before latitude	WGS84Bounding-Box, see Subclause 10.2 in OGC 06-121r3	Zero or more (optional) Include when useful or needed [d]
supportedCRS SupportedCRS	Reference to GridBaseCRS of GridCRSs in which GetCoverage operation outputs may be expressed	URI	Zero or more (optional) [e, f] Include one for each GridBaseCRS supported
supportedFormat SupportedFormat	Identifier of format in which GetCoverage operation response may be encoded	Character String type, not empty (ows:MimeType)	Zero or more (optional) [e] Include one for each format supported
identifier Identifier [c]	Unambiguous identifier of this coverage, unique for this WCS server.	Character String type, not empty	Zero or one (optional) Include when this coverage is available for DescribeCoverage and GetCoverage requests [g]
coverageSummary CoverageSummary	Brief metadata describing one subsidiary coverage available from this server	Coverage-Summary data structure, see this table	Zero or more (optional) One for each subsidiary coverage, unordered [f]

a   See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.

b   Contents inherited, not included as a named data structure.

c   Values are identifiers in Contents section of GetCapabilities operation response. When the Capabilities XML document does not have any CoverageSummary sections, clients must obtain a valid coverage identifier from another source, such as a catalog service.

d   This WGS84BoundingBox may be approximate, but should be as precise as practical. When multiple WGS84 bounding boxes are included, this shall be interpreted as the union (or combination) of the areas of these bounding boxes. Table 7 below specifies when a WGS84BoundingBox is needed. Currently, this WCS specification provides no way for a server to specify the precision of a bounding box.

e   When included, this list, including all values inherited from parent coverages, shall be an exact copy of the equivalent parameters in the CoverageDescription. Each CoverageSummary shall list or inherit at least one SupportedCRS and one SupportedFormat. There may be a dependency of SupportedCRS on SupportedFormat as described in Subclause 10.3.5.

f   A SupportedCRS may be referenced as the GridBaseCRS of a GridCRS specified in the Output part of a GetCoverage request. In addition, the GetCoverage operation output may be expressed in the ImageCRS of an unrectified offered coverage, instead of in a GridCRS. These SupportedCRSs may also be referenced in the BoundingBox in the DomainSubset part of a GetCoverage request.

g   This Identifier shall be included unless one or more CoverageSummaries are included. When one or more CoverageSummaries are included, inclusion of the Identifier is optional.

8.3.3.3 Reserved characters in CoverageSummary parameter values

The following CoverageSummary parameter values shall be usable literally ("as-is") in DescribeCoverage or GetCoverage operation requests to the originating server, encoded in KVP or XML:

a) Identifier

b) SupportedFormat

In stating values for these parameters, servers shall avoid, encode, or escape certain special characters, as detailed in Subclause 7.4, to avoid misinterpreting them as part of the XML or KVP syntax of subsequent requests.

8.3.3.4   CoverageSummary hierarchy

The CoverageSummary data structures may form a hierarchical tree. When the CoverageSummaries form a tree, a subsidiary (child) CoverageSummary inherits some of the (optional) parameters from its higher level (parent) CoverageSummaries in the hierarchy. In this subclause, a "leaf" CoverageSummary is defined as one that has no subsidiary (child) CoverageSummary. Table 7 specifies the inheritance rules for the inherited parameters.

Table 7 — Inheritance of parts of CoverageSummary data structure

Name	Definition	Inheritance by subsidiary coverages
Description	Description of a coverage	Not inherited [a, b]
Identifier	Unambiguous identifier of this coverage, unique for this WCS server	Not inherited [a, c]
Metadata	Reference to more metadata about this coverage	Not inherited [a, c]
WGS84-BoundingBox	Minimum bounding rectangle surrounding coverage, using WGS 84 CRS with decimal degrees and longitude before latitude	Inherited when not provided [d, e]
SupportedCRS	Reference to GridBaseCRS of GridCRSs in which GetCoverage operation outputs may be expressed	Inherited and possibly added to [f, h]
Supported Format	Identifier of format in which GetCoverage operation response may be encoded	Inherited and possibly added to [g, h]
Coverage Summary	Metadata describing one subsidiary coverage available from this server	Not inherited

a   Although these parameters are not inherited by a subsidiary coverage, the values of these parameters for all higher levels in a hierarchy of coverages may be relevant and are available to clients.

b   A value for this mandatory parameter shall be provided at all levels in the hierarchy.

c   Values for these optional parameters may be provided at all levels in the hierarchy.

d   When no value is provided for a coverage, any value recorded for a higher level in a hierarchy of coverages shall apply to this coverage. When a value is provided for a coverage, it overrides any value recorded for a higher level in a hierarchy of coverages.

e   For each "leaf" CoverageSummary, at least one WGS84BoundingBox shall be either recorded or inherited, for each georectified or georeferenced coverage. (This WGS84BoundingBox is included to simplify searching for coverages that might overlap a specified region.)

f   The full list of CRSs supported by a coverage shall be the union of the CoverageSummary's own supportedCRSs and those supported by all its ancestor CoverageSummaries.

g   The full list of formats supported by a coverage shall be the union of the CoverageSummary's own SupportedFormats and those supported by all its ancestor CoverageSummaries.

h   There can be a dependency of SupportedCRS values on SupportedFormat values, as described in Subclause 10.3.5.

NOTE 1   This specification supports a CoverageSummary hierarchy only as a means to simplify and shorten the Capabilities document. No other possible meanings and uses of a CoverageSummary hierarchy are specified here.

A "leaf" CoverageSummary shall have an Identifier parameter; a non-leaf CoverageSummary may have an Identifier. A WCS server that supplies a CoverageSummary containing an Identifier element shall accept that Identifier value in DescribeCoverage and GetCoverage operation requests. A client shall not issue a DescribeCoverage or GetCoverage request for a coverage whose CoverageSummary has no Identifier parameter.

NOTE 2   This specification does not organize CoverageDescription data structures into a hierarchy, and the CoverageSummary hierarchy is not encoded in the CoverageDescription. A CoverageDescription provides all of the information necessary to request a coverage using the GetCoverage operation, and to correctly interpret the results from that GetCoverage operation. A coverage with a non-leaf CoverageSummary, when it has an Identifier, is described via DescribeCoverage, and requested via GetCoverage, in the same way as one with a leaf CoverageSummary.

NOTE 3   This subclause implies that only coverages which satisfy all the requirements to be correctly called a grid coverage, as specified in ISO 19123, can have an Identifier. Such an offered coverage may be a "virtual" coverage, which is not stored but is assembled (in an unspecified way) by the server in response to a GetCoverage operation.

8.3.3.5 OtherSource

The optional OtherSource parameter references a catalogue server from which coverage metadata is available. This ability is expected to be used by servers with thousands or millions of coverages, for which searching a catalogue is more feasible than retrieving and then searching a very large Capabilities XML document. When no CoverageSummary items are included, and one or more catalogue servers are referenced, this set of catalogues shall contain current metadata summaries for all the coverages currently available from this WCS server, with the metadata for each such coverage referencing this WCS server.

NOTE   There is no requirement that all the coverages available from a specific WCS be listed in the Capabilities document (so that a catalogue may harvest them). For example, a server may need to serve on the order of (~) 10,000,000 distinct coverages with ~10,000 new coverages per day! In that case, this OtherSource may point to one or more catalogues that may be searched. For each catalogued coverage, that catalogue shall list the WCS server(s) that offer(s) that coverage, and the coverage identifier(s) used.

Any catalogue servers that are referenced should implement one of the interfaces specified in the OGC Catalogue Services Specification [OGC 04-021r3], preferably using the CSW protocol binding specified in Clause 10 of [OGC 04-021r3]. For any OGC catalogue protocol binding, an Application Profile for cataloging WCS coverages should be formally specified. Similarly, when no CoverageSummary items are included, a WCS Application Profile should be formally specified.

8.3.3.6 Implementation requirements

The "Multiplicity and use" columns in Table 9 through Table 17 in [OGC 06-121r3], and in Table 5 and Table 6 of this document, specify the optionality of each listed parameter and data structure in the GetCapabilities operation response. All the "mandatory" parameters and data structures shall be implemented by all WCS servers, using a specified value(s).

Implementation by WCS servers of the "updateSequence" parameter defined in Table 9 of [OGC 06-121r3] is optional. As indicated in Table 2 of this document, the "updateSequence" parameter may be implemented by each WCS server, and then shall also be required in the operation response.

All other "optional" parameters and data structures, in the GetCapabilities operation response, should be implemented by all WCS servers using specified values, whenever and wherever each is considered useful metadata for that server.

8.3.3.7 XML encoding

The Contents section of the Capabilities document shall be XML encoded. The XML Schema Document that specifies the Contents section for the WCS is built upon the owsContents.xsd schema, is named wcsContents.xsd, and is attached.

8.3.4   Capabilities document XML encoding

The XML Schema fragment for a WCS service metadata document extends the ows:CapabilitiesBaseType in owsGetCapabilities.xsd from [OGC 06-121r3]:

```
<element name="Capabilities">
   <complexType>
      <complexContent>
         <extension base="ows:CapabilitiesBaseType">
            <sequence>
```

```
                    <element ref="wcs:Contents" minOccurs="0"/>
               </sequence>
          </extension>
     </complexContent>
  </complexType>
</element>
```

8.3.5   Capabilities document SOAP encoding

WCS servers that implement SOAP version 1.2 transfer of GetCoverage operation requests shall also implement SOAP transfer of the corresponding GetCoverage operation responses. These operation responses shall be encoded as specified in Annex E, using the XML encoding referenced above.

8.3.6   Exceptions

When a WCS server encounters an error while performing a GetCapabilities operation, it shall return an exception report message as specified in Clause 8 of [OGC 06-121r3]. The allowed exception codes shall include those listed in Table 8 of Subclause 7.4.1 of [OGC 06-121r3] when the updateSequence parameter is implemented by the server.

8.4   GetCapabilities operation examples

To request a Capabilities document, a client could issue the following GetCapabilities minimal operation request encoded as KVP:

```
http://hostname:port/path?
service=WCS&request=GetCapabilities
```

The corresponding minimum request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities xmlns="http://www.opengis.net/wcs/1.1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1.1
../wcsGetCapabilities.xsd"
service="WCS"/>
```

In response to such a GetCapabilities request, the WCS server might generate a document that looks like the examples given in Subclause C.2: a top-level Capabilities element with all four child elements present – ServiceIdentification, ServiceProvider, OperationsMetadata, and Contents.

Next is a more detailed GetCapabilities operation request, requesting just the Contents section – first in KVP encoding:

```
http://hostname:port/path?
Service=WCS&request=GetCapabilities&
updateSequence=XYZ123&
AcceptVersions=1.1.1,1.0.0&
Sections=Contents&
AcceptFormats=text/xml
```

And in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities xmlns="http://www.opengis.net/wcs/1.1.1"
xmlns:ows="http://www.opengis.net/ows/1.1"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1.1
../wcsGetCapabilities.xsd"
service="WCS" updateSequence="XYZ123">
   <ows:AcceptVersions>
      <ows:Version>1.0.0</ows:Version>
      <ows:Version>1.1.1</ows:Version>
   </ows:AcceptVersions>
   <ows:Sections>
      <ows:Section>Contents</ows:Section>
   </ows:Sections>
   <ows:AcceptFormats>
      <ows:OutputFormat>text/xml</ows:OutputFormat>
   </ows:AcceptFormats>
</GetCapabilities>
```

The response to such a request would be an XML file with a top-level Capabilities element – as illustrated in Subclause C.2, but with ONLY the Contents section (omitting ServiceIdentification, ServiceProvider, and OperationsMetadata).

9   DescribeCoverage operation (mandatory)

9.1   Introduction

Once a client has obtained summary descriptions of a WCS server's available coverages, it may be able to make simple GetCoverage requests. However, in most cases the client will need to issue a DescribeCoverage request to obtain a full description of one or more coverages available. The server responds to such a request with an XML document describing one or more coverages served by that WCS. Each WCS server shall implement the DescribeCoverage operation.

9.2   DescribeCoverage operation request

9.2.1   DescribeCoverage request parameters

A DescribeCoverage operation request lists one or more coverages to be described, each identified by an identifier parameter. A request to perform the DescribeCoverage operation shall use the data structure shown in the UML diagram in Figure 5. The attributes of the new class shall include the parameters listed and defined in Table 8.
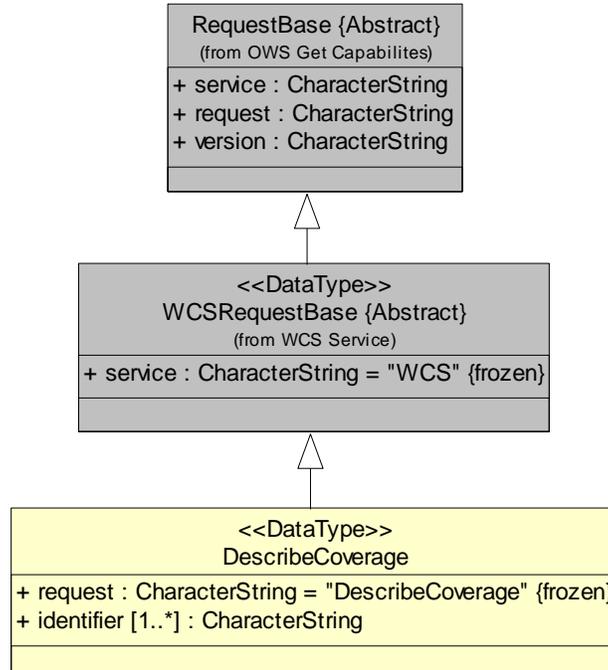
Figure 5 — DescribeCoverage operation request UML class diagram

NOTE 1   To reduce the need for readers to refer to other documents, the first three parameters listed below are largely copied from Table 26 in Subclause 9.2.1 of [OGC 06-121r3]. The Identifier parameter listed below is largely copied from Table 6 above.

Table 8 — Parameters in DescribeCoverage operation request

Names [a]	Definition	Data type and value	Multiplicity and use
service service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, namely "WCS"	One (mandatory)
request request	Operation name	Character String type, not empty Value is operation name, namely "DescribeCoverage"	One (mandatory)
version version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
identifier Identifier	Unambiguous identifier of a coverage, unique for this WCS	Character String type, not empty [b]	One or more (mandatory) Include for each desired coverage
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			
b Values are identifiers in the Contents section of the GetCapabilities operation response. When the Capabilities XML document does not have any Coverage Summary in the Contents section, clients must obtain a valid coverage identifier from another source, such as from a catalog service. See Subclause 8.3.3.1.			

The "Multiplicity and use" column in Table 8 specifies the optionality of each listed parameter in the DescribeCoverage operation request. All parameters in the operation request

are mandatory. That is, WCS clients shall implement all parameters, using the specified values, and WCS servers shall accept all parameters of the DescribeCoverage operation request, checking that each parameter contains the specified value.

9.2.2 DescribeCoverage request KVP encoding

WCS servers may implement KVP encoding for HTTP GET transfer of DescribeCoverage operation requests. Table 10 below specifies the DescribeCoverage request in KVP encoded form.

Table 9 — DescribeCoverage request URL encoding

URL component [a]	Description	Optionality
http://server_address/path/script?	URL of WCS server.	Required
service=WCS	Service name. Shall be "WCS".	Required
request=DescribeCoverage	Request name. Shall be "DescribeCoverage"	Required
version=1.1.2	Request protocol version.	Required
identifiers=identifier1, identifier2, …	A comma-separated list of coverage identifiers to describe (identified by their identifier values in the Capabilities document [b]	Required

a   All parameter names are here listed using mostly lower case letters. However, see Table 1 of [OGC 06-121r3] for UML and XML naming conventions..

b   This single "identifiers" parameter shall correspond to the repeatable "identifier" parameter in the UML model. See footnote b in Table 8.

9.2.3 DescribeCoverage request XML encoding

WCS servers may implement XML encoding for HTTP POST transfer of DescribeCoverage operation requests. The schema fragment that specifies the contents and structure of a DescribeCoverage operation request encoded in XML shall be as specified in the early part of the attached wcsDescribeCoverage.xsd file

9.2.4 DescribeCoverage request SOAP encoding

WCS servers may implement SOAP version 1.2 transfer of the DescribeCoverage operation request as specified in Annex E, using the XML encoding referenced above.

9.3   DescribeCoverage operation response

9.3.1   Normal response parameters

9.3.1.1 Coverage descriptions

The normal response to a valid DescribeCoverage operation request shall describe each coverage requested. More precisely, the response from a DescribeCoverage operation shall use the CoverageDescriptions data structure shown in the UML diagram in Figure 6, which uses the Domain and Range data structures shown in the UML diagrams in Figure 7 through Figure 9. The attributes and associations of the new classes shall include the parameters and data structures listed and defined in Table 10 through Table 20.
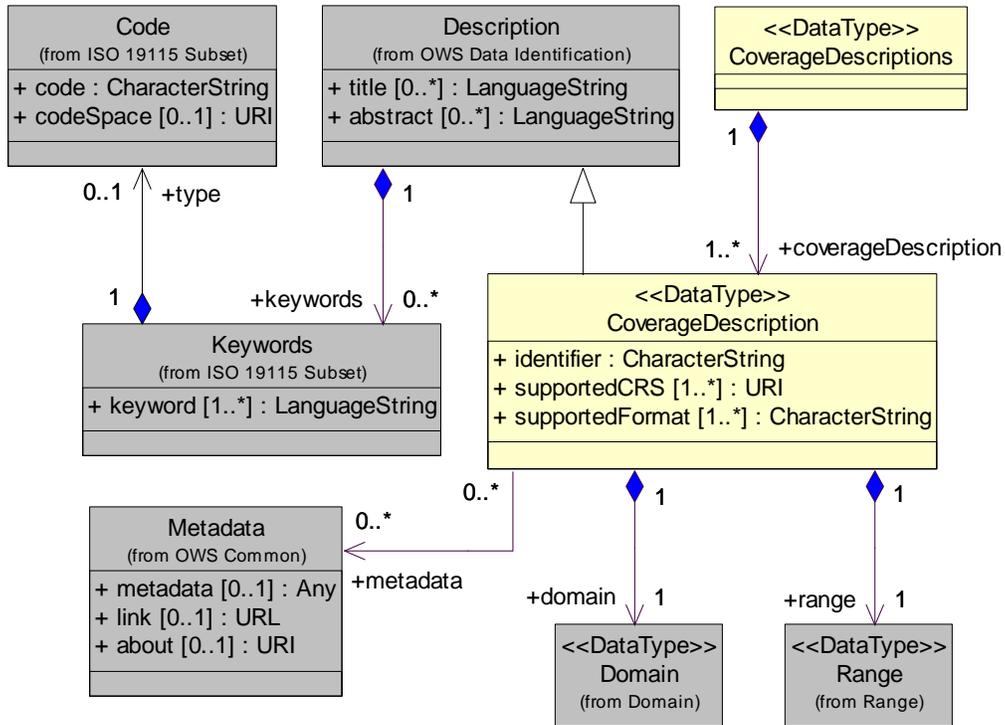
Figure 6 — DescribeCoverage operation response UML class diagram

Table 10 — Parts of CoverageDescriptions data structure

Names [a]	Definition	Data type	Multiplicity
coverageDescription CoverageDescription	Full description of one coverage available from WCS server	CoverageDescription data structure, see Table 11	One or more (mandatory)
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			

NOTE 1   To reduce the need for readers to refer to other parts of this document, the first three parameters listed below are largely copied from Table 6 in this document.

Table 11 — Parts of CoverageDescription data structure

Names [a]	Definition	Data type	Multiplicity and use
(Description) [b]	Description of a coverage	Description data structure, see Subclause 10.6.3 of [OGC 06-121r3]	One (mandatory) [b]
identifier Identifier	Unambiguous identifier of this coverage, unique for this WCS server	Character String type, not empty	One (mandatory)
metadata Metadata	More metadata about this coverage	Metadata data structure, see Table 23 of [OGC 06-121r3]	Zero or more (optional) Include when useful
domain Domain	Available coverage locations in space and/or time	Domain data structure, see Table 12	One (mandatory)
range Range	Available coverage values	Range data structure, see Table 17	One (mandatory)
supportedCRS SupportedCRS	Reference to GridBaseCRS of GridCRSs in which GetCoverage operation outputs may be expressed [c]	URI	One or more (mandatory) Include one for each GridBaseCRS supported [d]
supportedFormat SupportedFormat	Identifier of format in which GetCoverage operation response can be encoded	CharacterString type, not empty	One or more (mandatory) Include one for each format supported [d]

a   See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.

b   Contents inherited, not included as a named data structure. Normally includes only one Metadata item, but may include more than one when used here, especially for an unrectified but georeferenced image coverage.

c   A SupportedCRS may be referenced as the GridBaseCRS of a GridCRS specified in the Output part of a GetCoverage request. The GridBaseCRS of the GridCRS of a georectified offered coverage shall be listed as a SupportedCRS. An ImageCRS for an unrectified offered image shall be listed as a SupportedCRS, so that it may be referenced as the GridBaseCRS of a GridCRS. This ImageCRS shall be the ImageCRS of that unrectified offered image, or the ImageCRS that is referenced as the GridBaseCRS of the GridCRS that is used by that unrectified offered image In addition, the GetCoverage operation output maybe expressed in the ImageCRS or GridCRS of an unrectified offered coverage, instead of in a specified GridCRS. These SupportedCRSs can also be referenced in the BoundingBox in the DomainSubset part of a GetCoverage request.

d   There may be a dependency of a SupportedCRS upon a SupportedFormat: see Subclause 10.3.5.

9.3.1.2 Domain of coverage

The contents of the Domain data structure shall be as specified in Figure 7 and Table 12 through Table 16.
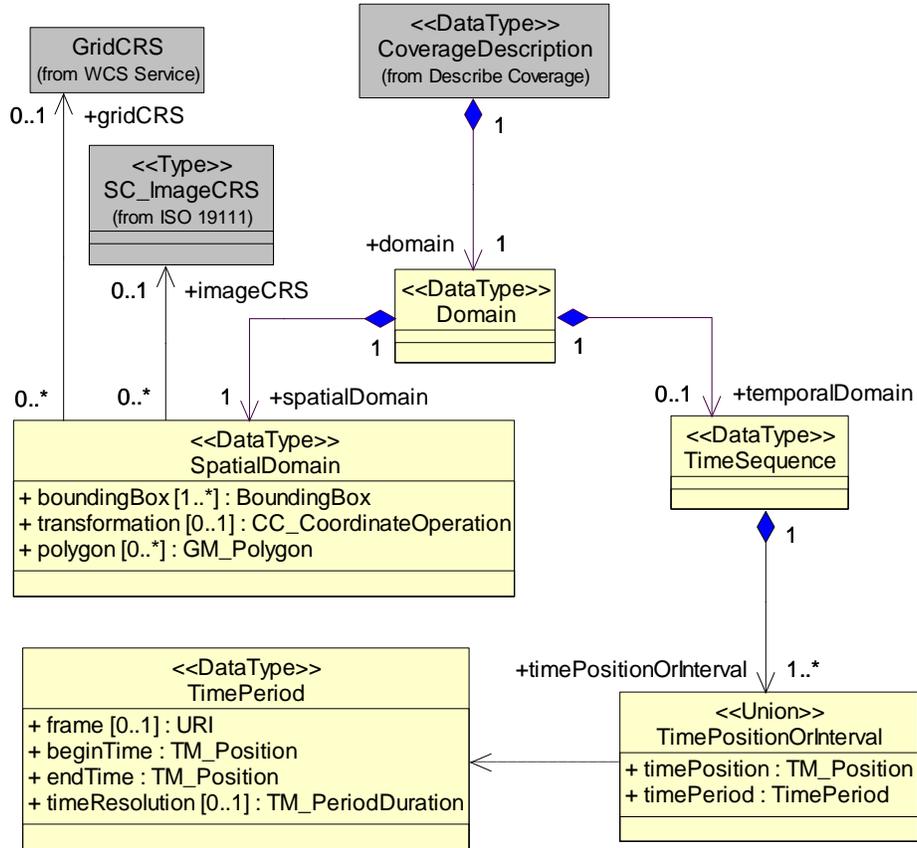
Figure 7 — Domain UML class diagram

Table 12 — Parts of Domain data structure

Names [a]	Definition	Data type	Multiplicity and use
spatialDomain SpatialDomain	Definition of spatial domain of this coverage	SpatialDomain data structure, see Table 13	One (mandatory)
temporalDomain TemporalDomain	Definition of temporal domain of this coverage	TimeSequence data structure, see Table 14	Zero or one (optional) Include when available [b]
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			
b Although optional, the TemporalDomain should be included whenever a value is known or a useful estimate is available, to provide a time for the coverage.			

Table 13 — Parts of SpatialDomain data structure

Names [a]	Definition	Data type	Multiplicity and use
boundingBox BoundingBox	Bounding box describing spatial domain of this coverage [b]	BoundingBox data structure, see Subclause 10.2 in [OGC 06-121r3]	One or more (mandatory) Include for each bounding box useful for describing spatial domain
gridCRS GridCRS	Definition of GridCRS of this coverage	GridCRS data structure, see Annex F	Zero or one (optional) Include when coverage has GridCRS [c]
transformation Transformation	Georeferencing coordinate transformation for unrectified coverage	CC_CoordinateOperation, see ISO 19111 [d]	Zero or one (optional) Include when available [e]
imageCRS ImageCRS	Association to ImageCRS of this unrectified coverage	Association to ImageCRS	Zero or one (optional) Include when coverage does not have GridCRS [f]
polygon Polygon	Polygon whose area covers spatial domain of this coverage [g]	GM_Polygon, see ISO 19107	Zero or more (optional) Include one for each polygon useful to describe domain

a   See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.

b   The first bounding box shall exactly specify the spatial domain in the (one offered) CRS of the offered coverage, thus specifying the available grid row and column indices. For a georectified coverage (that has a GridCRS), this bounding box shall specify the spatial domain in that GridCRS. For an image that is not georectified, this bounding box shall specify the spatial domain in the ImageCRS or GridCRS of that image, whether or not that image is georeferenced.

Additional bounding boxes, if any, shall specify the spatial domain in other CRSs. These additional bounding boxes may be approximate, but should be as accurate as practical. One bounding box could simply duplicate the information in the ows:WGS84BoundingBox in the CoverageSummary; but the intent is to describe the spatial domain in more detail (e.g., in several different CRSs, or several rectangular areas instead of one overall bounding box). Multiple bounding boxes with the same CRS shall be interpreted as an unordered list of bounding boxes whose union covers spatial domain of this coverage. WCS use of a BoundingBox is further specified in specification Subclause 7.5.4.

c   This GridCRS shall be included in the SpatialDomain when the offered coverage has a GridCRS. A georectified coverage shall always have a GridCRS. An unrectified image may have a known GridCRS that has an ImageCRS as its GridBaseCRS. This GridCRS applies to this offered coverage, and specifies its spatial resolution in its GridBaseCRS.

d   This CoordinateOperation can be a Transformation or a ConcatenatedOperation that includes at least one Transformation.

e   A Transformation that is a georeferencing coordinate transformation for an unrectified offered coverage should be included in the SpatialDomain when available and the coverage is georeferenced but not georectified. When included, this Transformation will specify the variable spatial resolution of this non-georectified image. To support use cases 4, 5, 9, or 10 specified in Annex G, a WCS server needs to use a georeferencing coordinate transformation for a coverage that is georeferenced but not georectified. However, a WCS server may opt not to support those use cases, or to use other means (such as a linked metadata file) to convey the georeferencing transformation, or not to make that transformation available to clients at all. Conversely, a client may opt to ignore any Transformation and to treat such a coverage as a non-georeferenced coverage (or to obtain the transformation by some other means).

f   The ImageCRS shall be referenced in the SpatialDomain when the offered coverage does not have a known GridCRS. Such a coverage is always unrectified, but an unrectified coverage may have a GridCRS instead of an ImageCRS. This ImageCRS applies to this offered coverage, but does not (normally) specify its spatial resolution. The ImageCRS and the GridCRS shall be mutually exclusive in this SpatialDomain.

g   Polygons are particularly useful for areas that are poorly approximated by an ows:BoundingBox (such as satellite image swaths, island groups, other non-convex areas).

Table 14 — Parts of TimeSequence data structure

Names [a]	Definition	Data type	Multiplicity
timePositionOrInterval TimePositionOrInterval	Ordered sequence of time positions or intervals [b]	TimePositionOrInterval type, see Table 15	One or more (mandatory)
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			
b The time positions and periods should normally be ordered from the oldest to the newest.			

Table 15 — Parts of TimePositionOrInterval data structure

Names [a]	Definition	Data type	Multiplicity
timePosition TimePosition	A specific time	TM_Position, see ISO 19108 [b, c]	Zero or one (optional) [c]
timePeriod TimePeriod	A period of time	TimePeriod type, see Table 16	Zero or one (optional) [c]
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			
b Values shall be expressed in the ISO 8601:2000 syntax, summarized in Subclause 9.3.2.4.			
c Inclusion of the timePosition or TimePeriod is mutually exclusive.			

Table 16 — Parts of TimePeriod data structure

Names [a]	Definition	Data type	Multiplicity
beginTime BeginTime	Start of this time period	TM_Position, see ISO 19108 [b]	One (mandatory)
endTime EndTime	End of this time period	TM_Position, see ISO 19108 [b]	One (mandatory)
timeResolution TimeResolution	Spacing between valid times within this time period	TM_PeriodDuration, see ISO 19108 [b]	Zero or one (optional) Include when describing periodic samples in time
frame frame	Temporal reference system	URI	Zero or one (optional) Default: "#ISO-8601"
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			
b Values shall be expressed in the ISO 8601:2000 syntax, summarized in Subclause 9.3.2.4.			

NOTE   The TimePeriod data structure is a variation of the GML TimePeriod, which allows the beginning and end of a time-period to be expressed in short-form inline using the Begin and End Position values.

9.3.1.3 Range of coverage

The contents of the Range data structure shall be as specified in Figure 8, Figure 9, and Table 17 through Table 20.
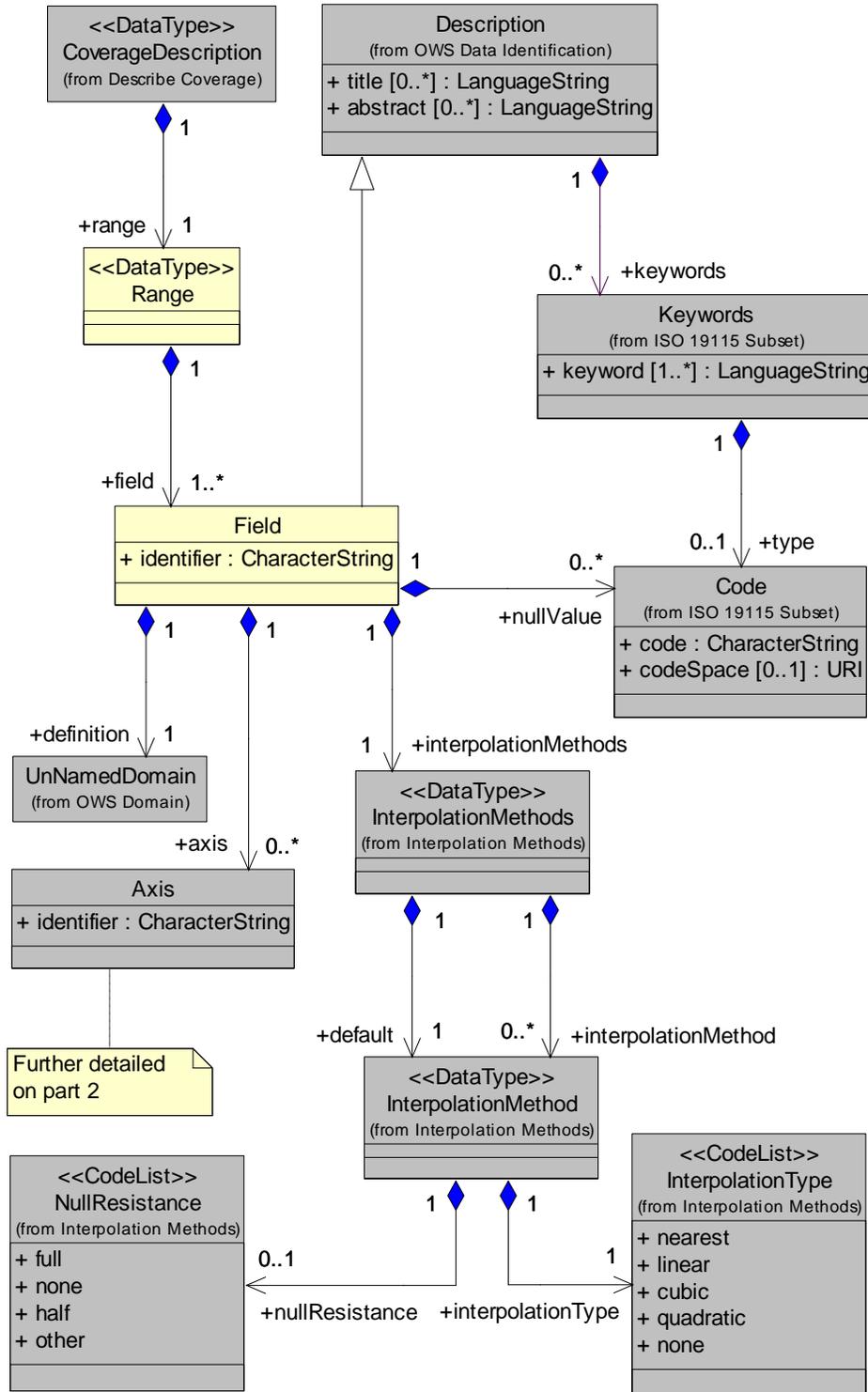
Figure 8 — Range UML class diagram, part 1

Continued
from part 1
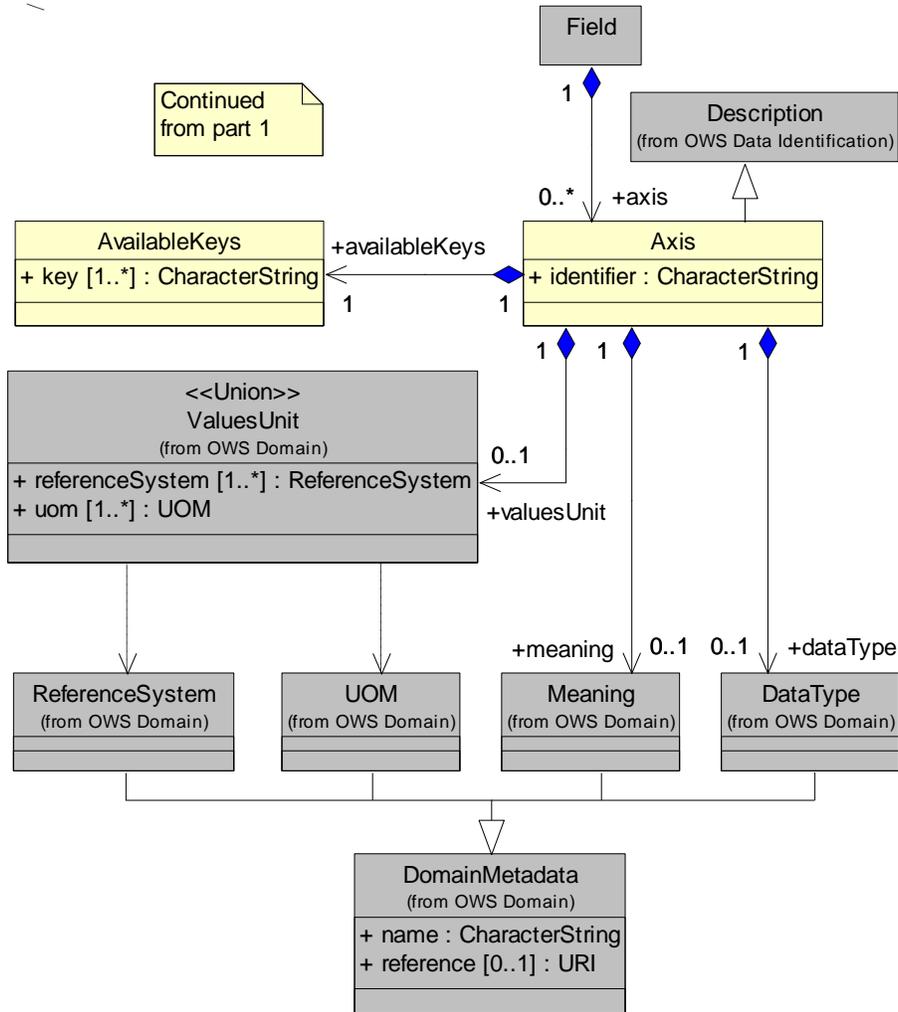
Figure 9 — Range UML class diagram, part 2

Table 17 — Parts of Range data structure

Names [a]	Definition	Data type	Multiplicity and use
field Field	Description of a field in range records, either scalar-valued or vector-valued	Field data structure, see NOTE 1 To reduce the need for readers to refer to other parts of this document, the first two parameters listed below are largely copied from Table 6 in this document. Table 18	One or more (mandatory) One for each field in this Range record
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			

NOTE 1 To reduce the need for readers to refer to other parts of this document, the first two parameters listed below are largely copied from Table 6 in this document.

Table 18 — Parts of Field data structure

Names [a]	Definition	Data type	Multiplicity and use
(Description) [b]	Description of this field	Description data structure, see Subclause 10.6.3 of [OGC 06-121r3]	One (mandatory) [b]
identifier Identifier	Identifier of this field, unique for this coverage	Character string type, not empty	One (mandatory)
definition Definition	Further definition of this field, including meaning, units, etc.	UnNamedDomain data structure, see [OGC 06-121r3] [e]	One (mandatory)
nullValue NullValue	Value used when valid range values are not available [c]	Code data structure, see Subclause 10.6.1 of [OGC 06-121r3]	Zero or more (optional) Include when needed [c]
interpolationMethods InterpolationMethods	Spatial interpolation method(s) that server can apply to this field [d]	InterpolationMethods data structure, see Subclause H.3	One (mandatory)
axis Axis	Axis ("control variable") of vector field for which there are range values	Axis data structure, see NOTE 2 To reduce the need for readers to refer to other parts of this document, the first two parameters listed below are largely copied from Table 6 in this document. The last four parameters are largely copied from Table 37 in Subclause 13.2.1 of [OGC 06-121r3]. Table 19	Zero or more (optional) Include for each axis of a vector field (i.e., one that has axes)
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			
b Contents inherited, not included as a named data structure.			
c The coverage encoding itself may specify a fixed value for null (e.g. "–99999" or "N/A"), but often the choice is up to the provider and must be communicated to the client outside of the coverage itself. The optional codeSpace XML attribute can reference a definition of the reason why this value is null.			
d One of these interpolation methods shall be used when a GetCoverage operation request requires resampling. When the only interpolation method listed is 'none', clients may only retrieve coverages from this offered coverage with its offered CRS at its offered resolution and grid point locations.			
e In the Definition, the AllowedValues should be used to encode the extent of possible values for this field or for all subfields, excluding the Null Value(s). If the extent of possible values is not known, AnyValue should be used.			

NOTE 2   To reduce the need for readers to refer to other parts of this document, the first two parameters listed below are largely copied from Table 6 in this document. The last four parameters are largely copied from Table 37 in Subclause 13.2.1 of [OGC 06-121r3].

Table 19 — Parts of Axis data structure

Names [a]	Definition	Data type	Multiplicity and use
(Description) [b]	Description of this field	Description data structure, see Subclause 10.6.3 of [OGC 06-121r3]	One (mandatory) [b]
Identifier identifier	Identifier of this axis, unique for this field	Character string type, not empty	One (mandatory)
availableKeys AvailableKeys	List of values of keys for this axis	AvailableKeys data structure, see Table 20	One (mandatory)
meaning Meaning	Reference to meaning or semantics of this value or set of values	DomainMetadata data structure, see Subclause 13.2.1 of [OGC 06-121r3]	Zero or one (optional) Include when useful
dataType DataType	Reference to the data type of this set of values	DomainMetadata data structure, see Subclause 13.2.1 of [OGC 06-121r3]	Zero or one (optional) Include when useful
valuesUnit ValuesUnit	Indicates that this quantity has units or reference system, and provides the value used [c]	ValuesUnit data structure, see Subclause 13.2.1 of [OGC 06-121r3]	Zero or one (optional) Include when values have units or reference system
metadata Metadata	Additional metadata about domain of this quantity	ows:Metadata, see Table 23 of OGC 06-121r3	Zero or more (optional) One for each such metadata object [d]

a   See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.

b   Contents inherited, not included as a named data structure.

c   Provides the identifier of the units or reference system used by the AllowedValues or ValuesListReference.

d   These metadata objects can be included in any order. A list of the required and/or optional metadata objects for each quantity can be specified in a WCS Application profile.

Table 20 — Parts of AvailableKeys data structure

Names [a]	Definition	Data type	Multiplicity and use
key Key	Value for a key of this axis	Character string type, not empty [b]	One or more (mandatory) [c] One for each key value [d]

a   See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.

b   Value shall be string encoding of any value of another data type.

c   There will normally be more than one key value for an axis.

d   For numeric keys, signed values shall be ordered from negative infinity to positive infinity.

9.3.1.4 Implementation requirements

The "Multiplicity and use" columns in Table 10 through Table 20 specify the optionality of each listed parameter and data structure in the DescribeCoverage operation response. All the "mandatory" parameters and data structures shall be implemented by all WCS servers, using a specified value(s).

All the "optional" parameters and data structures in the DescribeCoverage operation response should be implemented by all WCS servers using specified values, whenever and wherever each is considered useful metadata for that coverage.

9.3.2   Supporting information

9.3.2.1 Scalar and vector fields

The fields in a coverage range record may be scalar-valued or vector-valued. A scalar-valued field associates a single number (or other "atomic" value) to each location in the domain.

EXAMPLE 1   A scalar-valued field would be useful for terrain elevation or panchromatic (full-spectrum) brightness of a landscape.

A vector-valued field in a coverage range conveys multiple identically defined observations (measuring the same phenomenon and expressing it in same reference system) at each location, recorded for different values ("keys") of one or more control variables ("axes"). Clients may request subsets of such a field by selecting a subset of these keys.

EXAMPLE 2   The range of a hyperspectral image coverage would usually have a single vector-valued field, representing brightness values at different points along the electromagnetic spectrum. The axis of such a field would be the set of spectral bands, or the wavelengths, for which a sensor has recorded brightness values. A client might retrieve the near-infrared portions of a hyperspectral image by requesting only certain keys on an axis named "wavelength."

For a vector field, the **Axis** data structure describes the nature of each control variable, and indicates (as **Keys**) the constraint values available for selecting field subsets. The Key values shall be textual names, which may be numbers encoded as character strings. The axes of a field span a multi-dimensional array of range values, as depicted in Figure 10 below.
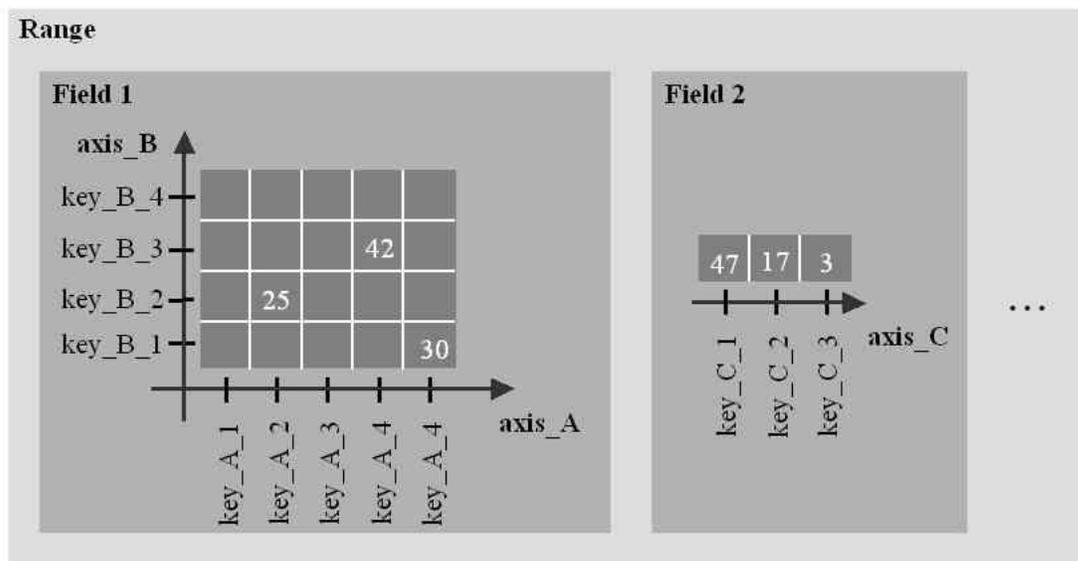
Figure 10 — Example vector fields

Figure 10 shows two example vector fields, one with two axes, and the other with one axis. These define rectangular arrays: that is, for all Keys along each Axis, the corresponding "slices" through the array have the same size.

NOTE 1   If a set of recorded observations has any semantic variation (including any differences in the measurement process or the reference system in which quantities are expressed), then the different observations belong in different fields or possibly different coverages. For instance, atmospheric pressure, water temperature, and land ownership require distinct fields in a coverage range. Distinct fields would also be needed for a multispectral image in which some bands represent emissive radiance, and others reflective radiance.

NOTE 2   When a WCS service offers two sets of data for one area but with different spatial resolutions, then they need to be described as two separate coverages because they cannot share a common domain description.

9.3.2.2 SupportedFormat

The *required* **SupportedFormat** element(s) advertises an output format in which a coverage may be requested from a WCS server. Formats are identified by a MIME type string. Any coverage format that can adequately convey the domain and range of the requested coverage is acceptable. The following formats have been used to convey certain kinds of coverages:

- GeoTIFF <http://www.remotesensing.org/geotiff/geotiff.html>
- HDF-EOS <http://www.hdfeos.org>
- NITF <http://www.ismc.nga.mil/ntb/baseline/1999.html>
- CF-NetCDF <http://www.cgd.ucar.edu/cms/eaton/cf-metadata>

NOTE   This specification deprecates the privileged status given to some of the above formats in WCS 1.0. The above formats are cited merely as examples, with no normative intent.

However, for improved interoperability, clients and servers should use formats defined in WCS encoding extensions adopted by the OGC. These encoding extensions shall contain the following information:

a) MIME type(s) and brief description: a concise overview of the encoding format, including the MIME type string(s) used to refer to it, the files required (e.g. header, dictionary, georeferencing, etc.), and the "role" values in the Xlink references in the GetCoverage response (see Subclauses 10.3.11.2 and H.2.2).

b) Pointers to documentation for the encoding format. This documentation shall clarify how the encoding convention represents locations, times, and physical quantities represented in the dataset.

c) Data model mapping to the Coverage Abstract Specification. This should include conventions for representing the spatio-temporal domain, and for representing the fields in the range. It should also describe limitations of the format for encoding complex coverages, and limitations of the coverage model for representing complex data structures encoded in the format.

d) Metadata mapping and encoding. Mapping of the coverage metadata to the data format encoding's metadata representation, if any. This mapping may address as many of the GetCoverage response metadata items as expressible in the format.

e) Examples: A set of examples of the encoding format, and of corresponding Coverages XML response documents (see Subclauses 10.3.11.2 and H.3).

f) Pointers to software implementing the encoding format. Providers of this software may license it in source code or executable form.

g) Compliance Testing: Pointers to mechanisms for testing whether resulting WCS coverages conform to the encoding format.

The titles of these encoding profiles shall follow the following pattern:

```
WCS 1.1 Extension for [Format] [formatVersion] encoding, [profileVersion]
```

EXAMPLE   WCS 1.1 Application Profile for GeoTIFF 6.0 encoding, 0.9

9.3.2.3 Reserved characters in CoverageDescription parameter values

The following CoverageDescription parameter values shall be usable literally ("as-is") in subsequent GetCoverage operation requests to the originating server, encoded in KVP or XML:

a)   "Identifier" parameter in CoverageDescription

b)   "SupportedFormat" parameter in CoverageDescription

c)   "Identifier" parameter in Field

d)   "InterpolationMethod" parameters in InterpolationMethods in Field

e)   "Identifier" parameter in Axis

f)   "Key" parameters in AvailableKeys in Axis

In stating values for these parameters, servers shall avoid, encode, or escape certain special characters, as detailed in Subclause 7.4, to avoid misinterpreting them as part of the XML or KVP syntax of subsequent requests.

In addition, when the WCS server supports KVP GetCoverage requests, the identifiers for Field, Axis, Key, and InterpolationMethod shall be usable literally within a RangeSubset parameter (see 10.2.2.2 below). This requires avoiding, encoding, or escaping any characters ('[' ']' ':' ';' ',') that are meaningful to the RangeSubset syntax. Servers may use any scheme for encoding or escaping these characters.

NOTE   The method for encoding or escaping special characters for RangeSubset is not specified. This is because clients will present the resulting identifier strings only to the originating server.

9.3.2.4 Summary of ISO 8601 syntax for time positions and time periods

The [ISO 8601:2000] syntax for dates and times may be summarized by the following template (see Annex D of the OGC Web Map Service [OGC 06-042]):

```
ccyy-mm-ddThh:mm:ss.sssZ
```

Where

"ccyy-mm-dd" is the date (a four-digit year, and a two-digit month and day);

"T" is a separator between the data and time strings;

"hh:mm:ss.sss" is the time (a two-digit hour and minute, and fractional seconds);

"Z" represents the Coordinated Universal Time (UTC or "zulu") time zone.

ISO 8601:2000 allows *(i)* up to a 14-digit year, with a negative sign, to denote the distant past; *(ii)* omitting less-significant numbers in the template to reduce the precision of the date and time; and *(iii)* use of other time zones, denoted as in Subclause 5.3.4.1 of ISO 8601:2000. Here a few examples:

EXAMPLE 1   2006   The year 2006

EXAMPLE 2   2006-09   September 2006

EXAMPLE 3   2006-09-27T10:00Z   10 o'clock (Universal Time) on 27 September, 2006

ISO 8601:2000 also provides a syntax for expressing time periods: the designator P, followed by a number of years Y, months M, days D, a time designator T, number of hours H, minutes M, and seconds S. Unneeded elements may be omitted. Here are a few examples:

EXAMPLE 4   P1Y   1 year

EXAMPLE 5   P1M10D   1 month plus 10 days

EXAMPLE 6   PT2H   2 hours

The WCS GetCoverage KVP syntax (defined in 10.2.2) extends ISO 8601:2000 with a syntax for time lists and ranges, as specified in Annex D of [OGC 06-042]. Some examples follow:

EXAMPLE 7   2000-06-23T20:07:48.11Z (A single moment)

EXAMPLE 8   1999-01-01,1999-04-01,1999-07-01,1999-10-01 (A list of four dates)

EXAMPLE 9   1995-04-22T12:00Z/2000-06-21T12:00Z/P1D
              (Daily ("P1D") between 12 noon on April 22 1995 and 12 noon on June 21, 2000)

9.3.3   Normal response XML encoding

The XML Schema fragment that specifies the contents and structure of a DescribeCoverage operation response shall be as specified in the latter part of the attached wcsDescribeCoverage.xsd file.

9.3.4   Normal response SOAP encoding

WCS servers that implement SOAP version 1.2 transfer of DescribeCoverage operation requests shall also implement SOAP transfer of the corresponding GetCoverage operation responses. These operation responses shall be encoded as specified in Annex E, using the XML encoding referenced above.

9.4   DescribeCoverage exceptions

When a WCS server encounters an error while performing a DescribeCoverage operation, it shall return an exception report message as specified in Subclause 7.4 of [OGC 06-121r3]. The allowed standard exception codes shall include those listed in Table 21. For each listed exceptionCode, the contents of the "locator" parameter value shall be as specified in the right column of Table 21.

NOTE   To reduce the need for readers to refer to other documents, all four values listed below are copied from Table 25 in Subclause 8.3 of [OGC 06-121r3].

Table 21 — Exception codes for DescribeCoverage operation

exceptionCode value	Meaning of code	"locator" value
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value [a]
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit "locator" parameter
a When the Coverage parameter is invalid, operation request identifies a Coverage not offered by this server.		

9.5   DescribeCoverage operation examples

The following example requests a description of a single coverage – first in KVP:

```
http://server_address/path/script?service=WCS
&request=DescribeCoverage&version=1.1.1
&identifiers=SPVIEW_530_274
```

And in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<DescribeCoverage xmlns="http://www.opengis.net/wcs/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
../wcsDescribeCoverage.xsd"
service="WCS" version="1.1.2">
   <Identifier>SPVIEW_530_274</Identifier>
</DescribeCoverage>
```

In response to such a request, the WCS server might generate a document like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<CoverageDescriptions xmlns="http://www.opengis.net/wcs/1.1"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
../wcsDescribeCoverage.xsd
http://www.opengis.net/ows/1.1 ../../../ows/1.1.0/owsAll.xsd">
   <CoverageDescription>
      <ows:Title>TBD</ows:Title>
      <ows:Abstract>TBD</ows:Abstract>
      <Identifier>TBD</Identifier>
      <Domain>
         <SpatialDomain>
            <ows:BoundingBox>
               <ows:LowerCorner>-30.00 -30.00</ows:LowerCorner>
               <ows:UpperCorner>30.00 30.00</ows:UpperCorner>
            </ows:BoundingBox>
         </SpatialDomain>
      </Domain>
      <Range>
        <Field>
           <ows:Title>TBD</ows:Title>
           <ows:Abstract>TBD</ows:Abstract>
```

```
                <Identifier>TBD</Identifier>
                <Definition>
                    <ows:AnyValue/>
                </Definition>
                <InterpolationMethods>
                    <InterpolationMethod>linear</InterpolationMethod>
                    <Default>cubic</Default>
                </InterpolationMethods>
            </Field>
        </Range>
        <SupportedCRS>urn:ogc:def:crs:EPSG::XXXX</SupportedCRS>
        <SupportedCRS>urn:ogc:def:crs:EPSG::YYYY</SupportedCRS>
        <SupportedFormat>text/xml</SupportedFormat>
    </CoverageDescription>
</CoverageDescriptions>
```

Clients may request descriptions of several coverages at once, in KVP:

```
http://server_address/path/script?service=WCS
&request=DescribeCoverage&version=1.1.2
&identifiers=Cov1,Cov2,Cov3
```

Or equivalently, in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<DescribeCoverage xmlns="http://www.opengis.net/wcs/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
../wcsDescribeCoverage.xsd"
service="WCS" version="1.1.2">
    <Identifier>Landsat_TM_Mosaic</Identifier>
    <Identifier>WMO_Daily_Temps</Identifier>
    <Identifier>Census_population_tables</Identifier>
</DescribeCoverage>
```

The corresponding response would have multiple `CoverageDescription` elements under the top-level `CoverageDescriptions`.

10 GetCoverage operation (mandatory)

10.1   Introduction

The mandatory GetCoverage operation allows retrieval of subsets of coverages. A WCS server processes a GetCoverage operation request and returns a response to the client that either contains or references the requested coverage(s). The GetCoverage operation shall be implemented by all WCS servers and clients.

10.2   GetCoverage operation request

10.2.1 GetCoverage request parameters

10.2.1.1   Overview

A request to perform the GetCoverage operation shall use the data structure shown in the UML diagram in Figure 11. The RangeSubset data structure is shown in the UML diagram in Figure 12. The attributes and associations of the new classes shall include the parameters and data structures listed and defined in Table 22 through Table 27.
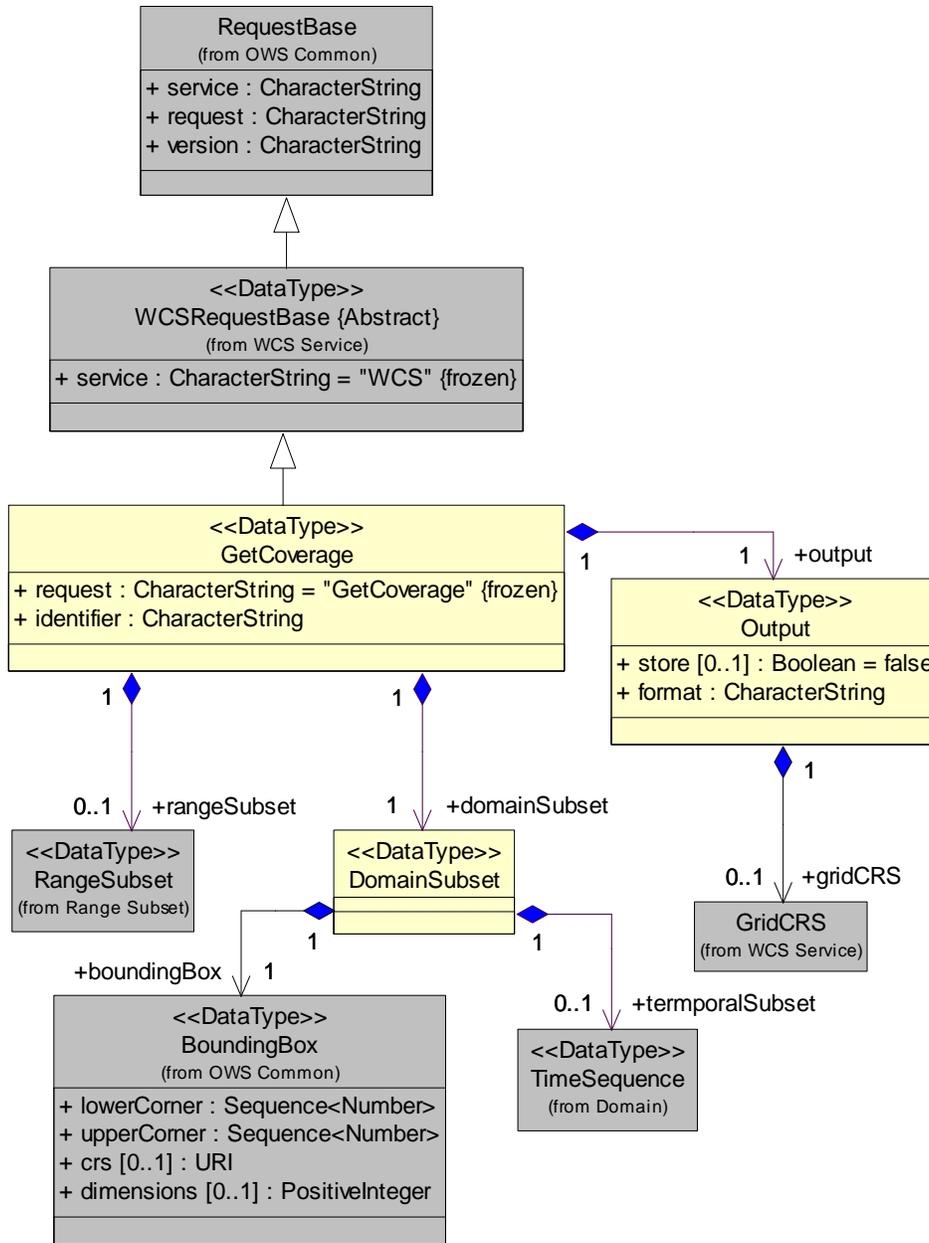
Figure 11 — GetCoverage request UML class diagram

NOTE 1   To reduce the need for readers to refer to other documents, the first three parameters listed below are largely copied from Table 26 in Subclause 9.2.1 of [OGC 06-121r3]. The fourth parameter listed below is largely copied from Table 6 above.

Table 22 — Parts of GetCoverage operation request

Names [a]	Definition	Data type and value	Multiplicity and use
service service	Service type identifier	CharacterString type, not empty Value is OWS type abbreviation, namely "WCS"	One (mandatory)
request request	Operation name	CharacterString type, not empty Value is operation name, namely GetCoverage	One (mandatory)
version version	Specification version for operation	CharacterString type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
identifier Identifier	Unambiguous identifier of coverage to draw from	CharacterString type, not empty [b]	One (mandatory)
domainSubset DomainSubset	Defines desired subset of domain set of identified coverage	DomainSubset data structure, see Table 23	One (mandatory)
rangeSubset RangeSubset	Defines desired subset of range(s) of identified coverage	RangeSubset data structure, see Table 25	Zero or one (optional) [c]
output Output	Specifies desired output options	Output data structure, see Table 24	One (mandatory)
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			
b Values are normally identifiers in Contents section of GetCapabilities operation response. When this information is not in the Contents section of the Capabilities XML document, clients shall obtain a valid coverage identifier from another source, such as a catalog service.			
c Providing a RangeSubset specifies a selection of range components and/or selects spatial interpolation methods for selected range components. Omitting a RangeSubset shall result in all range fields being delivered in their entirety, using the default spatial interpolation methods specified for each field.			

Table 23 — Parts of DomainSubset data structure

Names [a]	Definition	Data type	Multiplicity and use
boundingBox BoundingBox	Desired spatial subset of coverage, in any known CRS	BoundingBox data structure, see Subclause 10.2 of [OGC 06-121r3]	One (mandatory). [b]
temporalSubset TemporalSubset	Desired temporal subset of coverage	TimeSequence data structure, see Table 14	Zero or one (optional) Include when requesting temporal subset. [c]
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			
b To request coverage values at all available spatial locations, use the first BoundingBox supplied in the CoverageDescription for the chosen coverage. WCS use of a BoundingBox is further specified in Subclause 7.5.4.			
c Omit the TemporalSubset to request coverage values for all available times. If a server does not provide a temporal domain in an offered coverage's CoverageDescription, and the server receives a TemporalSubset parameter, the server shall raise an InvalidParameterValue exception.			

Table 24 — Parts of Output data structure

Names [a]	Definition	Data type	Multiplicity and use
gridCRS GridCRS	Definition of Coordinate Reference System in which GetCoverage operation response shall be expressed	GridCRS data structure, see Annex F	Zero or one (optional) Include unless requesting ImageCRS or GridCRS of offered coverage [d]
format Format	Identifier of format in which GetCoverage operation response shall be encoded	CharacterString type [b]	One (mandatory)
store store	"True" value asks server to store all components of the response coverage in a Web-accessible location, referenced by URL(s) in the GetCoverage response. "False" value asks server to return all components of the response coverage synchronously, bundled with the GetCoverage response.	Boolean type	Zero or one (optional) Default is false [c]
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			
b The format identifier value shall be among those listed under supportedFormats in the DescribeCoverage XML response.			
c This parameter should be included only when this parameter is supported by this server, as encoded in the OperationsMetadata section of the Capabilities document.			
d When this GridCRS is not included, the output coverage shall use the gml:ImageCRS or wcs:GridCRS of the offered coverage, normally with the origin at a different grid point. When included, this wcs:GridCRS specifies the desired grid for the output coverage, including the desired grid spacings, one of many possible origin locations, optional rotation, and optional skewing (angle between the axes) in a GridBaseCRS. This wcs:GridCRS is NOT required or expected to specify the actual position of the origin in the GetCoverage operation output, so it shall not be interpreted as being different than the origin directly or indirectly specified by the BoundingBox in the DomainSubset.			

10.2.1.2   RangeSubset

The contents of the RangeSubset data structure shall be as specified in Figure 12 and Table 25 through Table 27.
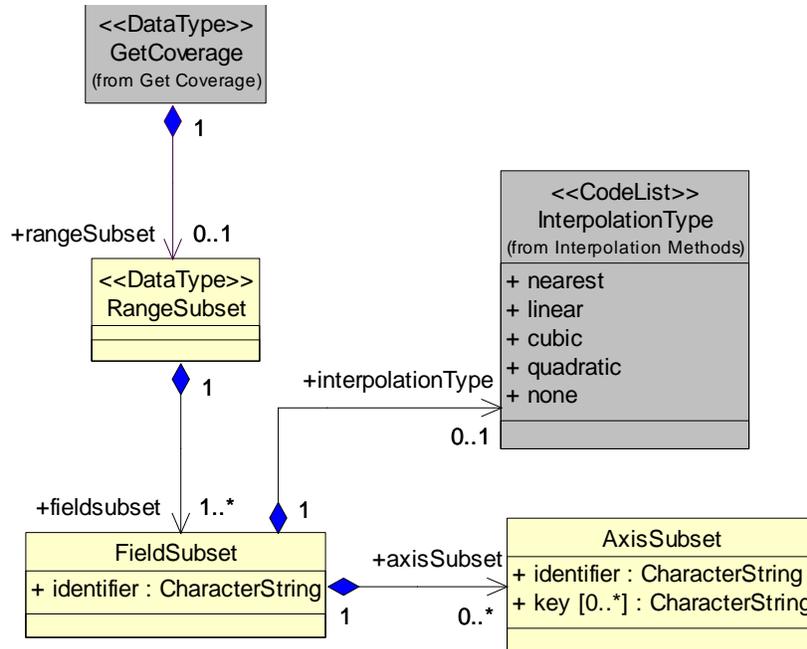
Figure 12 — RangeSubset UML class diagram

Table 25 — Parts of RangeSubset data structure

Names [a]	Definition	Data type	Multiplicity and use
fieldSubset FieldSubset	Requested field defined in range of requested coverage	FieldSubset data structure, see Table 26 below	One or more (mandatory) Include one for each requested field [b]
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			
b To request all values in all fields, with their default spatial interpolations, omit the RangeSubset.			

Table 26 — Parts of FieldSubset data structure

Names [a]	Definition	Data type and value	Multiplicity and use
identifier Identifier	Name of requested field	Character string, not empty [b]	One (mandatory)
interpolationType InterpolationType	Spatial interpolation method to be used to resample values in this field	InterpolationMethod-Type, see Table H.3 [c]	Zero or one (optional) Include to request interpolation method other than default for this field
axisSubset AxisSubset	Desired keys in one Axis for one (vector) field	AxisSubset data structure, see Table 27	Zero or more (optional) Include to request coverage values for only some of keys in an Axis defined for a (vector) field
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			
b This identifier shall be for a Field specified in the CoverageDescription, and shall be unique for this Coverage.			
c This interpolation type shall be one that is identified for this Field in the CoverageDescription. When this parameter is omitted, the interpolation type used shall be the default interpolation method specified for this Field, if any.			

Table 27 — Parts of AxisSubset data structure

Names [a]	Definition	Data type and value	Multiplicity and use
identifier Identifier	Name of requested Axis	Character string, not empty [b]	One (mandatory)
key Key	Desired key in this Axis	Character string, not empty [c]	One or more (mandatory) [d] Include to request coverage values for only some of the keys in this Axis
a	See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.		
b	Axis identifier shall be an Axis listed for the requested coverage field in the DescribeCoverage operation response.		
c	Keys shall be ones listed for this Axis of this Field in the CoverageDescription.		
d	To request field values for all keys on an Axis, omit the Axis from the FieldSubset.		

The range subset of the output coverage shall be a set of fields taken from the range fields of the requested coverage. Each range field shall occur at most once in the RangeSubset list of Fields. Each Field data structure selects a field from the original coverage, optionally selects components of this field, and optionally chooses an interpolation method to be used for this field (and all of its components, if any).

The InterpolationType parameter selects, for each range field, the interpolation method to be applied (if needed) in resampling operations. The InterpolationType value shall be one of those listed for the requested coverage Field in the DescribeCoverage operation response. One and the same interpolation method shall be used by a server for all values in a given (scalar or vector) Field. However, different Fields may employ different interpolation methods. When the GetCoverage request specifies no interpolation method for a Field, then the server shall apply the default interpolation method listed in the coverage's CoverageDescription.

For a vector field, each AxisSubset data structure shall consist of a nonempty subset of the Keys defined for this Axis. Each AxisSubset selects subsets of this field constrained by the given set of Keys. When an Axis Subset is absent, the Field Subset shall be unconstrained along that Axis, with its values delivered in the Axis and Key sequence given in the CoverageDescription. Thus, when no AxisSubset is present for a vector field, the entire field shall be delivered.

NOTE   Subclause 10.3.6 details the effect of the RangeSubset parameter on the result coverage.

10.2.1.3   Implementation requirements

The "Multiplicity and use" columns in Table 22 through Table 27 specify the optionality of each listed parameter and data structure in the GetCoverage operation request. All the "mandatory" parameters and data structures shall be implemented by all WCS clients, using a specified value(s). Similarly, all the "mandatory" parameters and data structures shall be implemented by all WCS servers, checking that each request parameter or data structure is received with any specified value(s).

Implementation of the "store" parameter in the GetCoverage operation request is optional for both clients and servers. When a WCS server that does not implement the "store" parameter receives the "store" parameter with the "False" value, it shall not send an exception report message and shall not store the result coverage. When a WCS server that does not implement the "store" parameter receives the "store" parameter with the "True" value, it shall return an

exception report message with the InvalidParameterValue value of the exceptionCode and "store" as the value of the "locator". Whether or not a WCS server implements the "store" parameter shall be specified in the OperationsMetadata section of the Capabilities document, as specified in Subclause 8.3.2 above.

All WCS servers shall, and all WCS clients should, implement the other "optional" parameters and data structures in the GetCoverage operation request, where appropriate.

10.2.2 GetCoverage request KVP encoding

10.2.2.1 Overview

WCS servers may use KVP encoding for HTTP GET transfer of GetCoverage operation requests. The KVP encoding of the GetCoverage operation request shall be as specified in Table 28. The parameters listed in Table 28 shall be as specified in Table 22 through Table 27 above.

Table 28 — GetCoverage request expressed as Key Value Pairs

Name and example [a]	Optionality & use	Definition and format
http://server_address/path/script?	Mandatory	URL of WCS server GetCoverage operation
service=WCS	Mandatory	Service name, shall be "WCS"
version=1.1.2	Mandatory	Request protocol version
request=GetCoverage	Mandatory	Name of the operation, shall be "GetCoverage"
identifier=identifier	Mandatory	Unique identifier of an available coverage
BoundingBox=47,-71, - 51,66, urn:ogc:def:crs:EPSG:6.6:6326 6405	Mandatory [b]	Request a coverage subset defined by the specified bounding box in the referenced coordinate reference system
TimeSequence= 20060801, 20060811, … OR TimeSequence = 20060801/ 2006-0901 / P1D, …	Optional, include when temporal subset desired other than default [b]	Request a subset corresponding to the specified time instants or intervals, expressed in the extended ISO 8601 syntax defined in Annex D of [OGC 06-042]. (See 9.3.2.4.)
RangeSubset=temp:nearest; radiance[band[1,2,5]] *(see syntax below)*	Optional; include when requesting values for only some of the available fields or field keys; or non-default interpolation types on fields.	Request only some fields OR subsets of some fields (OR list fields to request non-default interpolation on them). *(see syntax below)*
format=image/netcdf	Mandatory	Requested output format of Coverage. Shall be one of those listed in the description of the selected coverage
store=true	Optional, include when supported and not false	Specifies whether response coverage should be stored, remotely from client at network URL, a boolean value
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618	Optional, include all of five GridCRS parameters, except for defaulted parameters, to request output in a CRS other than the ImageCRS or GridCRS [h] of offered coverage	Reference to GridBaseCRS of desired output GridCRS, a URN [c]
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs		Reference to grid type of desired output GridCRS, a URN [c, d]
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS		Reference to coordinate system of desired output GridCRS, a URN [c, e]
GridOrigin=0,0		Position coordinates of one possible grid origin, in GridBaseCRS of desired output GridCRS [c, f]
GridOffsets=0.0707, -0.0707,0.1414,0.1414&		Offsets between adjacent grid points, in GridBaseCRS of desired output GridCRS [c, g]

a   All parameter names are here listed using mostly lower case letters. However, any parameter name capitalization shall be allowed in KVP encoding, see Subclause 11.5.2 of [OGC 06-121r3].

b   To request coverage values at all available spatial locations, use the first BoundingBox supplied in the CoverageDescription for the chosen coverage. WCS use of a BoundingBox is further specified in Subclause 7.5.4.

c   The GridType parameter shall reference the known definition of a grid type for a GridCRS, as specified in Annex F. That GridType definition shall specify the meanings of the other "Grid…" parameters listed, including the number of dimensions in each other "Grid…" parameter.

d   This GridType may be omitted when the GridBaseCRS is 2D, in which case the default value "urn:ogc:def:method:WCS:1.1:2dSimpleGrid" shall be used.

e   This GridCS may be omitted when the GridBaseCRS is 2D, in which case the default value shall be
"urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS".

f   This GridOrigin may be omitted when the GridBaseCRS is 2D, in which case the default value shall be "0.0 0.0".

g   When the GridType is "urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs", there shall be four values in this GridOffsets. The center
two of these offsets will be zero when the GridCRS is not rotated or skewed in the GridBaseCRS.

h   These GridCRS parameters, except for defaulted parameters, shall be included to request output in a CRS other than the
ImageCRS or GridCRS of the offered coverage. To request no interpolation, these parameters should be omitted.

10.2.2.2   RangeSubset KVP syntax

The syntax of the KVP RangeSubset parameter shall conform to the following grammar
(using EBNF (Extended Backus-Naur Form) notation [IETF RFC 2396]):

```
RangeSubset = FieldSubset *( ";" FieldSubset )
FieldSubset = FieldId [ ":" Interpolation ] [ "[" AxisSubsets "]" ]
AxisSubsets = AxisSubset *( "," AxisSubset )
AxisSubset = AxisId "[" Keys "]"
Keys = Key *( "," Key )
FieldId = <field identifier, as defined in Subclause 9.3.1.3 and Table 18>
AxisId = <axis identifier, as defined in Subclause 9.3.1.3 and Table 19>
Key = <key, as defined in Subclause 9.3.1.3 and Table 20>
Interpolation = <interpolation type identifier, as defined in Table H.3>
```

NOTE 1   No extra whitespace is allowed in a RangeSubset string – any whitespace is always considered part
of an adjacent identifier.

NOTE 2   The '[' and ']' characters are not allowed in URLs, and thus need to be encoded according to
Subclause 11.3 of [OGC 06-121r3].

EXAMPLES

```
rangeSubset=radiance;temperature
rangeSubset=radiance:linear;temperature:nearest
    rangeSubset=radiance:linear[wavelengths[L1,L2],forecastTimes[T1];
    temperature:nearest[forecastTimes[T1]]
```

When the WCS server supports KVP GetCoverage requests, the identifiers for FieldID,
AxisID, Key, and InterpolationMethod given in the CoverageDescription shall be used
literally ("as-is") in the RangeSubset parameter, with no additional character encoding or
escaping. (See 9.3.2.3 above.)

10.2.3   GetCoverage request XML encoding

WCS servers may implement XML encoding for HTTP Post transfer of GetCoverage
operation requests. The XML Schema fragment for the GetCoverage operation request shall
be as specified in the attached wcsGetCoverage.xsd file.

10.2.4   GetCoverage request SOAP encoding

WCS servers may implement SOAP version 1.2 transfer of the GetCoverage operation
request as specified in Annex E, using the XML encoding referenced above.

10.3 GetCoverage operation response

10.3.1 Introduction

The normal response to a valid GetCoverage operation request shall be a single coverage extracted from the coverage requested, with the specified spatial reference system, bounding box, size, format, and range subset.

10.3.2 GetCoverage exceptions

When a WCS server encounters an error while performing a GetCoverage operation, it shall return an exception report message as specified in Subclause 7.4 of [OGC 06-121r3]. The allowed standard exception codes shall include those listed in Table 29. For each listed exceptionCode, the contents of the "locator" parameter value shall be as specified in the right column of Table 29.

NOTE   To reduce the need for readers to refer to other documents, the first three values listed below are copied from Table 25 in Subclause 8.3 of [OGC 06-121r3].

Table 29 — Exception codes for GetCoverage operation

exceptionCode value	Meaning of code	"locator" value
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value [a]
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit "locator" parameter
UnsupportedCombination	Operation request contains an output CRS that can not be used within the output format.	Name of Format or name of CRS
NotEnoughStorage	Operation request specifies "store" the result, but not enough storage is available to do this.	None, omit "locator" parameter
[a] The Coverage parameter is not valid when the operation request identifies a Coverage that is not offered by this server. The Format parameter is not valid when the operation request identifies a Format not offered for this coverage by this server.		

10.3.3 Coverage processing

The WCS GetCoverage request invokes some or all of a set of five basic operations:

a) domain subsetting

b) range subsetting

c) resampling

d) data format encoding

e) result file delivery

Processing a GetCoverage request can conceptually be understood as executing the operation sequence shown in Figure 13. Depending on the particular request, some steps may be skipped. While a concrete implementation may choose a different evaluation strategy it shall ensure that the result is identical to the one achieved by the sequence depicted in Figure 13.

```
perform coordinate transformation, including needed
spatial scaling, into target CRS using resampling
(using spatial interpolation as specified)
↓
perform spatial subsetting
↓
perform temporal subsetting
↓
perform range subsetting
↓
perform data format encoding (may involve implicit
range interpolation and, hence, accuracy loss,
depending on the format chosen)
↓
output resulting coverage samples
```

Figure 13 — Conceptual GetCoverage request evaluation sequence

10.3.4   Output coverage CRS

Each output coverage from a GetCoverage operation shall be in the GridCRS that is specified in the Output data structure within the GetCoverage operation request, when such a GridCRS is specified there. When no GridCRS is included, the output coverage shall use the ImageCRS or GridCRS of the offered coverage, normally with the origin at a different grid point.

The offered coverage shall use a GridCRS when available, or an ImageCRS when it is an image that has not been georectified and no GridCRS is known. The offered coverage will have a GridCRS when it is georectified, and may have a GridCRS when it is unrectified with a known GridCRS that has an ImageCRS as its GridBaseCRS. An image that has not been georectified may be georeferenced or not.

When included in the GetCoverage request, this GridCRS specifies the desired grid for the output coverage, including the desired grid spacings, one of many possible origin locations, optional rotation, and optional skewing (angle between the axes) in a GridBaseCRS. However, this GridCRS is NOT required or expected to specify the actual position of the origin in the GetCoverage operation output. The position of the origin in the GetCoverage operation output shall be as directly or indirectly specified by the BoundingBox in the DomainSubset in the GetCoverage operation request.

10.3.5   SupportedFormat and SupportedCRS dependencies

Some formats require that the data be stored in a specific CRS, or the CRS must be chosen from a small list of CRSs. The dependency between CRSes and formats can not be expressed in this specification. When a WCS server receives a request where the requested output format and the requested output CRS are incompatible then the server shall return an UnsupportedCombination exception, see Table 29.

10.3.6   Output coverage spatial extent

The spatial extent of each output coverage shall be a rectangle in the CRS of that output coverage, using null values for any grid points whose values cannot be interpolated (when allowed) from grid point values included in the offered coverage. The BoundingBox of this spatial extent shall be computed, in the CRS of that output coverage, from the DomainSubset BoundingBox in the GetCoverage operation request. The specified extent may be skewed with curved edges in the desired output coverage CRS, although it is rectangular in the CRS of the DomainSubset BoundingBox. This will usually occur when a GridCRS is defined as the output coverage CRS, and this GridCRS (not necessarily the GridBaseCRS of this GridCRS) is different from the CRS used in the DomainSubset BoundingBox. Such a skewed and curved extent in the desired resampled output coverage shall be interpreted as the minimum extent desired.

NOTE 1   WCS use of BoundingBoxes is further specified in specification Subclause 7.5.4.

When the specified extent is skewed/curved in the desired output coverage CRS, the server shall extend this extent to the minimum rectangle surrounding that skewed/curved extent. And the server shall extend this minimum rectangle to the next grid point rows and columns in the output coverage GridCRS or ImageCRS.

NOTE 2   Computing the minimum surrounding rectangle may be done by considering a polygon with vertices at the four corner points of the specified bounding box and at least the center points in each side of that bounding box. The positions of these eight or more polygon vertices may be transformed from the CRS of the DomainSubset BoundingBox to the desired output coverage GridCRS (not necessarily the GridBaseCRS of this GridCRS). The minimum rectangle surrounding all these polygon vertex positions can then be computed, followed by extending each side to the next row or column of grid point positions.

The transformed and made-rectangular extent will often be skewed with curved edges in the offered coverage, although it is rectangular in the output coverage CRS. When the transformed skewed/curved extent does not lie wholly within the offered coverage, padding with null values shall be applied to all output grid points falling outside the offered coverage.

10.3.7   Output coverage temporal extent

The temporal extent of the output coverage shall be the time instance(s) or period(s) requested in the GetCoverage request. For a particular time instance or interval, when the server cannot supply coverage values, it shall substitute null values. When the GetCoverage request specifies no temporal subset, then the returned coverage shall return values for all times in the domain of the requested Coverage.

NOTE   The WCS currently does not allow interpolation of coverage values along the time dimension.

10.3.8   Output coverage range

The range of an output coverage is controlled by the RangeSubset data structure. The output coverage range shall consist of all and only the fields specified in the RangeSubset of the GetCoverage request, ordered as they appear there. When the GetCoverage request has no RangeSubset data structure, the output coverage range shall contain all fields of the offered coverage, in the sequence defined in the CoverageDescription, and with the default interpolation method(s) applied whenever interpolation is required.

In the case of a vector field, the output coverage field shall be limited to the values corresponding to keys in the AxisSubset structure in the GetCoverage FieldSubset. The absence of an AxisSubset on a FieldSubset shall be equivalent to requesting field values for

all keys of that Axis. (Thus, the absence of any AxisSubset at all for a field shall be equivalent to requesting the field in its entirety.)

The InterpolationType parameter selects, for each range field, the interpolation method to be applied whenever spatial interpolation is used during resampling. A server shall use the same interpolation method for all values in a field. However, it may use different interpolation methods for different fields. When the getCoverage request does not specify an interpolation method for a field, then the server shall apply the default interpolation method listed for that field in the CoverageDescription.

EXAMPLE   Assume a coverage with two fields, *radiance* and *temperature*. *radiance* has two Axes, "waveLength" with three keys("0.6", "0.7", and "0.9") and "forecastTime" with four keys ("08:00", "14:00", "20:00", "02:00"). The field *temperature* has the Axis "forecastTimes", with keys "06:00", "12:00", "18:00", "00:00". Both fields are assumed to support interpolation methods "linear" and "cubic". Then, the following RangeSubset clause selects radiance at waveLength keys 0.6 and 0.9 and forecastTime keys 08:00 and 14:00, and temperature at all forecast times. When spatial interpolation is needed, it requests linear interpolation for radiance and the (server-defined) default interpolation method for temperature.

```
RangeSubset=radiance:linear[waveLength[0.6,0.9],forecastTime[08:00,14:00]];
      temperature
```

10.3.9   Output coverage metadata

The metadata listed in Table 30, or the equivalent information, shall be included with each WCS output coverage. Equivalent metadata may be encoded with the coverage data in some possible coverage formats, as specified by a WCS encoding extension document (see 9.3.2.2 above). Such metadata may be XML encoded using the XML data type listed in Table 30. A WCS encoding extensiondocument may specify encoding equivalent metadata both within a specialized coverage format and outside that format, but such duplication is discouraged to avoid consistency issues. When any metadata is encoded both with and outside the coverage data, that duplicated metadata shall be as consistent as possible.

Table 30 — Required output coverage metadata

Names [a]	Definition	XML data type	Multiplicity and use
boundingBox BoundingBox	Bounding box that specifies extent of output coverage	ows:BoundingBoxType in CRS of output coverage [b]	One (mandatory)
gridCRS GridCRS	Definition of GridCRS used by output coverage	wcs:GridCRSType	Zero or one (conditional) Include when GridCRS used by BoundingBox
transformation Transformation	Definition of georeferencing coordinate transformation for output coverage	gml:Transformation or gml:ConcatenatedOperation	Zero or one (conditional) Include when output coverage is not georectified but is georeferenced
a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.			
b The Coordinate Reference System (CRS) of an output coverage may be either a GridCRS or an ImageCRS.			

10.3.10 Output coverage formats

A WCS server shall serve coverages in any of the formats listed as a SupportedFormat for the requested coverage (see 9.3.2.2 above). For most possible formats, more information is required to specify how a WCS output coverage shall (or may) be encoded using that format. That specific information should be specified in a WCS encoding extension document (see 9.3.2.2 above).

NOTE   Encoding may reduce the coverage's range accuracy when the format chosen does not support the coverage's accuracy in terms of range types. For example, GIF can only represent 256 color values, hence a GIF converter will approximate colors while determining a best-match color map. Encoding a result coverage in JPEG (with quality less than 1.0) will incur range value approximation. In all of these cases the pixels' radiometry will be inaccurate.

Regardless of the coverage format, the GetCoverage response shall be encoded as a Coverages XML document that references one or more files, which may be bundled with it or accessible separately via URL(s), as detailed in 10.3.11 below.

10.3.11 Response encodings

10.3.11.1 Overview

The encoding of the GetCoverage response consists of a Coverages XML document (see 10.3.11.2 below), packaged and bundled according to the request encoding and the value of the "store" parameter (see Table 24). WCS servers shall implement response encodings according to Table 31.

Table 31 — GetCoverage response encodings

Request encoding	store=true [a]	store=false *(default)*
KVP or XML	Coverages alone (see 10.3.11.2)	Coverages in a MIME Multipart message (see 10.3.11.3)
SOAP	Coverages alone in a SOAP envelope (see 10.3.11.4)	Coverages in a SOAP envelope with Attachments (see 10.3.11.5)
[a] Applies only when the server supports the "store" parameter (see Subclause 0).		

If the "store" parameter has the value "true", the server shall store the result file(s) at URL-addressable location(s) of its choosing, and return only the Coverages data structure (within a SOAP envelope in the case of a SOAP request) with references to the other files as indicated in 10.3.11.2 and 10.3.11.4 below.

When the "store" parameter is absent or has the value "false", the server shall transfer the complete GetCoverage response to the client, either as a MIME multipart message (for KVP or XML requests) or as a SOAP message with attachments (for SOAP requests). The Coverages shall reference the other parts of the MIME multi-part message (or SOAP attachments) as indicated in 10.3.11.3 below.

10.3.11.2 Coverages data structure

The GetCoverage operation response shall be XML encoded using the Coverages data structure specified in Annex H.2. This Coverages data structure allows the response to

reference multiple files, and shall be output even when the GetCoverage operation response would otherwise contain only one file.

As shown in the UML diagram in Figure H.1, this Coverages object contains one or more Coverage reference group objects, each of which includes one or more Reference elements. A Coverages class in a GetCoverage operation response shall reference all the data files for one output coverage, including the coverage values and any associated metadata.

EXAMPLE   An example XML document using the Coverages element is:

```
<?xml version="1.0" encoding="UTF-8"?>
<Coverages xmlns="http://www.opengis.net/wcs/1.1"
    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
../owsCoverages.xsd http://www.opengis.net/ows/1.1
../../../ows/1.1.0/owsAll.xsd">
    <Coverage>
        <ows:Title>TBD</ows:Title>
        <ows:Abstract>Coverage created from GetCoverage operation request
to a WCS</ows:Abstract>
        <Identifier>TBD</Identifier>
        <Reference xlink:href="http://my.server.com/coverage/image.tiff"
xlink:role="urn:ogc:def:role:WCS:1.1:coverage"/>
        <Reference xlink:href="http://my.server.com/coverage/metadata.xml"
xlink:role="urn:ogc:def:role:WCS:1.1:metadata"/>
    </Coverage>
</Coverages>
```

NOTE   The URN values used for Reference "role" attributes are to be specified in encoding extensions.

10.3.11.3 MIME multipart response

GetCoverage responses transferred using MIME multipart shall be encoded according to [IETF RFC 2387], with the first part containing an XML encoded Coverages element. Every part of the multipart message shall have a content-id. The part containing the Coverages element shall have a content-id equal to "**urn:ogc:wcs:1.1:coverages**"; other parts may have any content-id values. The Coverages element shall reference other parts of the multipart message by their content-id values, prefixed by "**cid:**".

EXAMPLE 1   If a multipart message contains a part with "content-id:ottawa_temp.tiff", then the Coverage element would reference that part with the following XML fragment:

```
<Reference xlink:href="cid:ottawa_temp.tiff"/>
```

NOTE   Using prefixes other than "cid:", the Coverages element can (but normally should not) reference resources external to the MIME multipart message.

10.3.11.4   SOAP response

WCS servers that implement SOAP transfer of GetCoverage operation requests shall also implement SOAP version 1.2 transfer of the corresponding GetCoverage operation responses. These operation responses shall be encoded as specified in Annex E. When the "store" parameter has the value "true", the body of the SOAP envelope shall contain one XML encoded Coverages element with one or more references to URLs of files containing coverage content and metadata.

10.3.11.5   SOAP with Attachments response

When the "store" parameter is absent or has the value "false", responses to GetCoverage SOAP requests shall be encoded as SOAP with Attachments as defined in [W3C Note] (but using SOAP 1.2 rather than SOAP 1.1). These responses shall consist of MIME multipart messages, as specified in 10.3.11.3, with the Coverages element contained within a SOAP 1.2 envelope, encoded as specified in Annex E.

10.3.12 Coverages XML encoding

WCS encoding extensions should be written for all formats, as specified in Subclause 9.3.2.2. These extensions shall specify how any of the coverage metadata specified in Subclause 10.3.9 shall be included in the same file as the coverage grid values. XML encoding is not expected to be used for the coverage grid values part of that file.

For each coverage, XML encoding should be used for the required coverage metadata specified in Subclause 10.3.9, when not included in the same file as the coverage grid values. It is also possible for an encoding extension to specify use of XML encoding for some of the metadata that is included in the same file as the coverage grid values. When XML encoding is used in the same file, the XML encoding referenced in Subclause 10.3.9 shall be used.

EXAMPLE   The GML in JPEG 2000 encoding specification [OGC 05-047r3] could be adapted to encode both coverage grid values and associated metadata.

10.4   GetCoverage operation examples

10.4.1   A minimal GetCoverage request

A client might issue the following GetCoverage operation request with minimum contents – encoded in KVP:

```
http://my.service.org/path/script?
service=WCS
&version=1.1.2
&request=GetCoverage
&identifier=Cov123
&BoundingBox=-71,47,-66,51,urn:ogc:def:crs:OGC:2:84
&format=image/netcdf
```

This corresponds to the following minimum request encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCoverage xmlns="http://www.opengis.net/wcs/1.1"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1 ../wcsGetCoverage.xsd"
service="WCS" version="1.1.2">
   <ows:Identifier>Cov123</ows:Identifier>
   <DomainSubset>
      <ows:BoundingBox crs="urn:ogc:def:crs:OGC:2:84">
         <ows:LowerCorner>-71 47</ows:LowerCorner>
         <ows:UpperCorner>-66 51</ows:UpperCorner>
      </ows:BoundingBox>
   </DomainSubset>
   <Output format="image/netcdf"/>
</GetCoverage>
```

57

This would request the portion of coverage "Cov123" falling within the given longitude and latitude bounds, with output encoded according to the "image/netcdf" convention for WCS (as specified in a separate profile). This request omits the following parameters:

a) TimeSequence (thus requesting coverage values for all available times)

b) RangeSubset (thus requesting all fields of "Cov123" in their entirety);

c) Output CRS parameters: GridBaseCRS, GridType, GridCS, GridOrigin, GridOffsets. This requests output expressed in the CRS of the offered coverage "Cov123".

d) Store (thus requesting the coverage in-line rather than posted to a URL)

10.4.2   A longer GetCoverage request

The following adds a simple range subset and a simple 2D output GridCRS to the previous example. It also asks the server to store the result rather than deliver it inline:

```
http://my.service.org/path/script?
service=WCS&version=1.1.1&request=GetCoverage
&identifier=Cov123&format=image/netcdf
&BoundingBox=-71,47,-66,51,urn:ogc:def:crs:OGC:2:84
&RangeSubset=radiance[waveLength[0.6,0.9]];temperature:cubic
&GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618
&GridOffsets=10,10
&store=true
```

The above KVP request is identical to the previous one except that it requests

a) a subset of the coverage range (only radiance for wavelength values 0.6 and 0.9, with the default spatial interpolation for this field; and temperature with bicubic interpolation);

b) an output grid with 10x10 meter pixels aligned with the northing/easting axes of WGS 84 UTM meters, Zone 18 north (EPSG 32618).

This corresponds to the following XML request:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCoverage xmlns="http://www.opengis.net/wcs/1.1"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1 ../wcsGetCoverage.xsd"
service="WCS" version="1.1.2">
   <ows:Identifier>Cov123</ows:Identifier>
   <DomainSubset>
      <ows:BoundingBox crs="urn:ogc:def:crs:OGC:2:84">
         <ows:LowerCorner>-71 47</ows:LowerCorner>
         <ows:UpperCorner>-66 51</ows:UpperCorner>
      </ows:BoundingBox>
      <TemporalSubset>
         <TimePeriod>
            <BeginPosition>2006-08-01</BeginPosition>
            <EndPosition>2006-09-01</EndPosition>
            <TimeResolution>P1D</TimeResolution>
         </TimePeriod>
      </TemporalSubset>
   </DomainSubset>
   <Output format="image/netcdf">
      <GridCRS>
         <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:32618</GridBaseCRS>
         <GridType>urn:ogc:def:method:WCS:1.1:2dGridin2dCrs</GridType>
```

```
            <GridOrigin>3000 4000</GridOrigin>
            <GridOffsets>6.0 8.0 -8.0 6.0</GridOffsets>
            <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS>
        </GridCRS>
    </Output>
</GetCoverage>
```

The request (whether in KVP or XML) implicitly uses default values for the following GridCRS parameters:

   a) GridType: default is "urn:ogc:def:method:WCS:1.1:2dSimpleGrid"
      (This GridType disallows rotation or skew relative to the GridBaseCRS – therefore
      GridOffsets has only two numbers.)

   b) GridCS: default is "urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS"

   c) GridOrigin: default is "0,0" (KVP) or "0 0" (XML).

10.4.3 Another GetCoverage request

The following request adds a time constraint and non-default gridCRS parameters for the output coverage.

```
http://my.service.org/path/script?
service=WCS&version=1.1.2&request=GetCoverage
&identifier=Cov123&format=image/netcdf
&TimeSequence=2006-08-01/2006-09-01/P1D
&GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618
&GridType=urn:ogc:def:method:WCS:1.1:2dGridin2dCrs
&GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS
&GridOrigin=3000,4000
&GridOffsets=8.0,6.0,-6.0,8.0
```

The TimeSequence KVP parameter provides the time constraint: daily ("P1D") between Aug. 1 and Sept. 1, 2006, inclusive. As before, the requested output grid is defined relative to WGS 84 UTM meters, Zone 18 North (EPSG 32618) – but rotated relative to the northing/easting lines: the two pairs of offsets in GridOffsets form 3-4-5 right triangles, for a 10x10 meter grid spacing and a counter-clockwise rotation of about 37 degrees. The output grid has its origin at (3000,4000) rather than (0,0).

This corresponds to the following XML request:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCoverage xmlns="http://www.opengis.net/wcs/1.1"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1 ../wcsGetCoverage.xsd"
service="WCS" version="1.1.2">
    <!-- Partial example. Primary editor: Arliss Whiteside. Last updated
2007-06-05-->
    <ows:Identifier>TBD</ows:Identifier>
    <DomainSubset>
        <ows:BoundingBox crs="urn:ogc:def:crs:OGC:2:84">
            <ows:LowerCorner>9999 9999</ows:LowerCorner>
            <ows:UpperCorner>9999 9999</ows:UpperCorner>
        </ows:BoundingBox>
    </DomainSubset>
    <RangeSubset>
        <FieldSubset>
            <ows:Identifier>TBD</ows:Identifier>
```

```
            </FieldSubset>
        </RangeSubset>
        <Output format="image/gif" store="false"/>
</GetCoverage>
```

10.4.4 Example responses

In response to such a request, the WCS server might generate a document that looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<Coverages xmlns="http://www.opengis.net/wcs/1.1"
    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
../owsCoverages.xsd http://www.opengis.net/ows/1.1
../../../ows/1.1.0/owsAll.xsd">
    <Coverage>
        <ows:Title>TBD</ows:Title>
        <ows:Abstract>Coverage created from GetCoverage operation request
to a WCS</ows:Abstract>
        <Identifier>TBD</Identifier>
        <Reference xlink:href="http://my.server.com/coverage/image.tiff"
xlink:role="urn:ogc:def:role:WCS:1.1:coverage"/>
        <Reference xlink:href="http://my.server.com/coverage/metadata.xml"
xlink:role="urn:ogc:def:role:WCS:1.1:metadata"/>
    </Coverage>
</Coverages>
```

Annex A
(normative)

Conformance

Specific conformance tests for a Web Coverage Service will be added in a future revision of this specification. At the moment, a WCS implementation must satisfy the following system characteristics to be minimally conformant with this specification:

a) *WCS Clients and servers* must support the *GetCapabilities*, *DescribeCoverage*, and *GetCoverage* operations.

b) *WCS clients* must issue *GetCapabilities* requests in Key-Value Pair (KVP), XML, or SOAP form. *GetCapabilities* KVP requests must conform to Subclause **8.2.2**. *GetCapabilities* XML requests must conform to Subclause **8.2.3**, and must be valid against the XML Schema definition in wcsGetCapabilities.xsd. *GetCapabilities* SOAP requests must conform to Subclause **8.2.4**.

c) *WCS servers* must respond to a *GetCapabilities* KVP or XML request with an XML document that conforms to Subclause **8.3**, and is valid against the XML Schema definition in wcsGetCapabilities.xsd and wcsContents.xsd. The response to a *GetCapabilities* SOAP request shall encapsulate this XML document in the body of a SOAP envelope, as specified in Annex E.

d) *WCS clients* must issue *DescribeCoverage* requests in Key-Value Pair (KVP) or XML form. *DescribeCoverage* KVP requests must conform to Subclause **9.2.2**. *DescribeCoverage* XML requests must conform to Subclause **9.2.3**, and must be valid against the XML Schema definition in wcsDescribeCoverage.xsd. *DescribeCoverage* SOAP requests must conform to Subclause **9.2.4**.

e) *WCS servers* must respond to a *DescribeCoverage* KVP or XML request with an XML document that conforms to Subclause **9.3**, and is valid against the XML Schema definition in wcsDescribeCoverage.xsd. The response to a *DescribeCoverage* SOAP request shall encapsulate this XML document in the body of a SOAP envelope, as specified in Annex E.

f) *WCS clients* must issue *GetCoverage* requests in Key-Value Pair (KVP) or XML form. *GetCoverage* KVP requests must conform to Subclause **10.2.2**. *GetCoverage* XML requests must conform to Subclause **10.2.3**, and must be valid against the XML Schema definition in wcsGetCoverage.xsd. *GetCoverage* SOAP requests must conform to Subclause **10.2.4**.

g) *WCS servers* must be able to respond to a *GetCoverage* KVP or XML operation with an XML document that conforms to Subclause **10.3.11**. The response to a *GetCoverage* SOAP request shall encapsulate this XML document in the body of a SOAP envelope, as specified in Subclause **10.3.11** and Annex E.

h) All paragraphs in the normative clauses of this specification that use the keywords "required", "shall", and "shall not" must be satisfied.

Annex B
(informative)

WCS XML Schemas

In addition to this document, this specification includes normative XML Schema Document files. The WCS-specific XML Schema Documents are posted online at the URL http://schemas.opengis.net/wcs/1.1. These XML Schema Documents are also bundled with the present document. In the event of a discrepancy between the bundled and online versions of the XML Schema files, the online files shall be considered authoritative.

The WCS abilities specified in this document use nine specified XML Schema Documents included in the zip file with this document, which are named:

a) wcsGetCapabilities.xsd

b) wcsDescribeCoverage.xsd

c) wcsGetCoverage.xsd

d) wcsContents.xsd

e) wcsCommon.xsd

f) wcsAll.xsd

g) wcs:GridCRS.xsd (specified in Annex F)

h) wcsCoverages.xsd (specified in Subclause H.2)

i) wcsInterpolationMethod.xsd (specified in Subclause H.3)

The wcsInterpolationMethod.xsd XML Schema Document references an XML document that specifies some values and meanings of the InterpolationType specified in Subclause H.3. That XML document is named interpolationMethods.xml, and is referenced at http://schemas.opengis.net/wcs/1.1/interpolationMethods.xml.

These XML Schema Documents use and build on the OWS Common XML Schema Documents listed in Annex B of [OGC 06-121r3], specified in the "ows" namespace.

All these XML Schemas contain documentation of the meaning of each element and attribute, and this documentation is considered normative as specified in Subclause 11.6.3 of [OGC 06-121r3].

Annex C
(informative)

XML document examples

C.1   Introduction

As an aid to understanding and a guide for implementation, this annex contains more example XML documents which are valid according to the XML schemas referenced in Annex B.

C.2   GetCapabilities operation XML encoding examples

To request a Capabilities document, a client could issue the following GetCapabilities minimal operation request encoded as KVP:

```
http://hostname:port/path&service=WCS&request=GetCapabilities
```

The corresponding minimum request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities xmlns="http://www.opengis.net/wcs/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
../wcsGetCapabilities.xsd"
service="WCS"/>
```

In response to such a GetCapabilities operation request, a WCS server might generate a document that looks like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities xmlns="http://www.opengis.net/wcs/1.1"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
../wcsGetCapabilities.xsd http://www.opengis.net/ows/1.1
../../../ows/1.1.0/owsAll.xsd"
version="1.1.2" updateSequence="ABC123">
   <ows:ServiceIdentification>
      <ows:Title>Acme Corp. Coverage Server</ows:Title>
      <ows:Abstract>Web Coverage Server maintained by Acme Corporation.
</ows:Abstract>
      <ows:Keywords>
         <ows:Keyword>USA</ows:Keyword>
      </ows:Keywords>
      <ows:ServiceType>OGC WCS</ows:ServiceType>
      <ows:ServiceTypeVersion>1.1.2</ows:ServiceTypeVersion>
      <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
      <ows:Fees>NONE</ows:Fees>
      <ows:AccessConstraints>NONE</ows:AccessConstraints>
   </ows:ServiceIdentification>
   <ows:ServiceProvider>
      <ows:ProviderName>Acme Corporation</ows:ProviderName>
```

```
        <ows:ProviderSite xlink:href="http://hostname/"/>
        <ows:ServiceContact>
            <ows:IndividualName>Jeff Smith, Server
Administrator</ows:IndividualName>
            <ows:PositionName>Computer Scientist</ows:PositionName>
            <ows:ContactInfo>
                <ows:Phone>
                    <ows:Voice>+1 301 555-1212</ows:Voice>
                    <ows:Facsimile>+1 301 555-1212</ows:Facsimile>
                </ows:Phone>
                <ows:Address>
                    <ows:DeliveryPoint>NASA Goddard Space Flight
Center</ows:DeliveryPoint>
                    <ows:City>Greenbelt</ows:City>
                    <ows:AdministrativeArea>MD</ows:AdministrativeArea>
                    <ows:PostalCode>20771</ows:PostalCode>
                    <ows:Country>USA</ows:Country>

    <ows:ElectronicMailAddress>user@host.com</ows:ElectronicMailAddress>
                </ows:Address>
            </ows:ContactInfo>
            <ows:Role>TBD</ows:Role>
        </ows:ServiceContact>
    </ows:ServiceProvider>
    <ows:OperationsMetadata>
        <ows:Operation name="GetCapabilities">
            <ows:DCP>
                <ows:HTTP>
                    <ows:Get xlink:href="http://hostname:port/path?"/>
                    <ows:Post xlink:href="http://hostname:port/path?"/>
                </ows:HTTP>
            </ows:DCP>
        </ows:Operation>
        <ows:Operation name="GetCoverage">
            <ows:DCP>
                <ows:HTTP>
                    <ows:Post xlink:href="http://hostname:port/path?"/>
                </ows:HTTP>
            </ows:DCP>
            <ows:Parameter name="Format">
                <ows:AllowedValues>
                    <ows:Value>image/gif</ows:Value>
                    <ows:Value>image/png</ows:Value>
                    <ows:Value>image/jpeg</ows:Value>
                </ows:AllowedValues>
            </ows:Parameter>
        </ows:Operation>
        <ows:Operation name="DescribeCoverage">
            <ows:DCP>
                <ows:HTTP>
                    <ows:Get xlink:href="http://hostname:port/path?"/>
                </ows:HTTP>
            </ows:DCP>
            <ows:Parameter name="Format">
                <ows:AllowedValues>
                    <ows:Value>text/xml</ows:Value>
                </ows:AllowedValues>
            </ows:Parameter>
        </ows:Operation>
    </ows:OperationsMetadata>
    <Contents>
        <CoverageSummary>
            <ows:Title>TBD</ows:Title>
```

```
            <ows:Abstract>TBD</ows:Abstract>
            <ows:WGS84BoundingBox>
                <ows:LowerCorner>-9999 9999</ows:LowerCorner>
                <ows:UpperCorner>9999 9999</ows:UpperCorner>
            </ows:WGS84BoundingBox>
            <Identifier>TBD</Identifier>
        </CoverageSummary>
    </Contents>
</Capabilities>
```

Here is another longer example response to a GetCapabilities operation request. It features fully detailed service metadata – Service Identification, Service Provider, OperationsMetadata, and Contents sections – the latter showing hierarchical descriptions afforded by CoverageSummary.

```
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities xmlns="http://www.opengis.net/wcs/1.1"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
../wcsGetCapabilities.xsd
http://www.opengis.net/ows/1.1 ../../../ows/1.1.0/owsAll.xsd"
version="1.1.2" updateSequence="1.0">
    <!-- *************************************************** -->
    <!-- * SERVICE IDENTIFICATION SECTION                * -->
    <!-- *************************************************** -->
    <ows:ServiceIdentification>
        <ows:Title>CubeWerx Demonstation WCS</ows:Title>
        <ows:Abstract>A demonstration server used to illustrate CubeWerx's
compilance with the Web Coverage Service 1.1.0 implementation
specification</ows:Abstract>
        <ows:Keywords>
            <ows:Keyword>Web Coverage Service</ows:Keyword>
            <ows:Keyword>06-083</ows:Keyword>
            <ows:Keyword>CubeWerx</ows:Keyword>
            <ows:Keyword>GeoTIFF</ows:Keyword>
            <ows:Keyword>Imagery</ows:Keyword>
        </ows:Keywords>
        <ows:ServiceType>WCS</ows:ServiceType>
        <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
        <ows:ServiceTypeVersion>1.1.2</ows:ServiceTypeVersion>
        <ows:Fees>NONE</ows:Fees>
        <ows:AccessConstraints>NONE</ows:AccessConstraints>
    </ows:ServiceIdentification>
    <!-- *************************************************** -->
    <!-- * SERVICE PROVIDER SECTION                      * -->
    <!-- *************************************************** -->
    <ows:ServiceProvider>
        <ows:ProviderName>CubeWerx Inc.</ows:ProviderName>
        <ows:ProviderSite xlink:href="http://www.cubewerx.com"/>
        <ows:ServiceContact>
            <ows:IndividualName>Panagiotis (Peter) A.
Vretanos</ows:IndividualName>
            <ows:PositionName>Senior Developer</ows:PositionName>
            <ows:ContactInfo>
                <ows:Phone>
                    <ows:Voice>123-456-7890</ows:Voice>
                    <ows:Facsimile>234-567-8901</ows:Facsimile>
                </ows:Phone>
                <ows:Address>
                    <ows:DeliveryPoint>15 rue Gamelin</ows:DeliveryPoint>
```

```
                    <ows:DeliveryPoint>Suite 506</ows:DeliveryPoint>
                    <ows:City>Gatineau</ows:City>
                    <ows:AdministrativeArea>Quebec</ows:AdministrativeArea>
                    <ows:PostalCode>J8Y 6N5</ows:PostalCode>
                    <ows:Country>Canada</ows:Country>

<ows:ElectronicMailAddress>pvretano[at]cubewerx[dot]com</ows:ElectronicMa
ilAddress>
                </ows:Address>
                <ows:OnlineResource
xlink:href="http://www.cubewerx.com/~pvretano"/>
                <ows:HoursOfService>24x7x365</ows:HoursOfService>
                <ows:ContactInstructions>email</ows:ContactInstructions>
            </ows:ContactInfo>
            <ows:Role>Developer</ows:Role>
        </ows:ServiceContact>
    </ows:ServiceProvider>
    <!-- ****************************************************** -->
    <!-- * OPERATIONS METADATA                              * -->
    <!-- ****************************************************** -->
    <ows:OperationsMetadata>
        <ows:Operation name="GetCapabilities">
            <ows:DCP>
                <ows:HTTP>
                    <ows:Get
xlink:href="http://demo.cubewerx.com/demo/cubeserv/cubeserv.cgi?service=W
MS&amp;"/>
                </ows:HTTP>
            </ows:DCP>
            <ows:Parameter name="service">
                <ows:AllowedValues>
                    <ows:Value>WCS</ows:Value>
                </ows:AllowedValues>
            </ows:Parameter>
            <ows:Parameter name="version">
                <ows:AllowedValues>
                    <ows:Value>1.0.0</ows:Value>
                    <ows:Value>1.1.2</ows:Value>
                </ows:AllowedValues>
            </ows:Parameter>
            <ows:Parameter name="AcceptVersions">
                <ows:AllowedValues>
                    <ows:Value>1.0.0</ows:Value>
                    <ows:Value>1.1.2</ows:Value>
                </ows:AllowedValues>
            </ows:Parameter>
        </ows:Operation>
        <ows:Operation name="DescribeCoverage">
            <ows:DCP>
                <ows:HTTP>
                    <ows:Get
xlink:href="http://demo.cubewerx.com/demo/cubeserv/cubeserv.cgi?service=W
MS&amp;"/>
                </ows:HTTP>
            </ows:DCP>
            <ows:Parameter name="service">
                <ows:AllowedValues>
                    <ows:Value>WCS</ows:Value>
                </ows:AllowedValues>
            </ows:Parameter>
            <ows:Parameter name="version">
                <ows:AllowedValues>
                    <ows:Value>1.0.0</ows:Value>
```

```
                <ows:Value>1.1.1</ows:Value>
            </ows:AllowedValues>
        </ows:Parameter>
        <ows:Parameter name="Identifier">
            <ows:AllowedValues>
                <ows:Value>ETOPO2</ows:Value>
                <ows:Value>GTOPO30</ows:Value>
                <ows:Value>RELIEF</ows:Value>
            </ows:AllowedValues>
        </ows:Parameter>
    </ows:Operation>
    <ows:Operation name="GetCoverage">
        <ows:DCP>
            <ows:HTTP>
                <ows:Get
xlink:href="http://demo.cubewerx.com/demo/cubeserv/cubeserv.cgi?service=W
MS&amp;"/>
            </ows:HTTP>
        </ows:DCP>
        <ows:Parameter name="service">
            <ows:AllowedValues>
                <ows:Value>WCS</ows:Value>
            </ows:AllowedValues>
        </ows:Parameter>
        <ows:Parameter name="version">
            <ows:AllowedValues>
                <ows:Value>1.0.0</ows:Value>
                <ows:Value>1.1.2</ows:Value>
            </ows:AllowedValues>
        </ows:Parameter>
        <ows:Parameter name="Identifier">
            <ows:AllowedValues>
                <ows:Value>ETOPO2</ows:Value>
                <ows:Value>GTOPO30</ows:Value>
                <ows:Value>RELIEF</ows:Value>
            </ows:AllowedValues>
        </ows:Parameter>
        <ows:Parameter name="InterpolationType">
            <ows:AllowedValues>
                <ows:Value>nearest</ows:Value>
                <ows:Value>linear</ows:Value>
            </ows:AllowedValues>
        </ows:Parameter>
        <ows:Parameter name="format">
            <ows:AllowedValues>
                <ows:Value>image/tiff</ows:Value>
                <ows:Value>image/tiff;
PhotometricInterpretation=RGB</ows:Value>
                <ows:Value>image/png</ows:Value>
                <ows:Value>image/png;
PhotometricInterpretation=PaletteColor</ows:Value>
                <ows:Value>image/png;
PhotometricInterpretation=RGB</ows:Value>
                <ows:Value>image/gif</ows:Value>
                <ows:Value>image/jpeg</ows:Value>
                <ows:Value>image/ppm</ows:Value>
            </ows:AllowedValues>
        </ows:Parameter>
    </ows:Operation>
</ows:OperationsMetadata>
<!-- ***************************************************** -->
<!-- * CONTENTS SECTION                                  * -->
<!-- ***************************************************** -->
```

```
    <Contents>
      <CoverageSummary>
        <ows:Title>Global 2 Minute Elevations</ows:Title>
        <ows:Abstract>ETOPO2 was generated from a digital data base of
land and sea-floor elevations on a 2-minute latitude/longitude grid. The
data sources used to create the ETOPO2 data set were: Smith/Sandwell,
GLOBE, DBDBV, IBCAO, and DBDB5</ows:Abstract>
        <ows:Metadata
xlink:href="http://www.ngdc.noaa.gov/mgg/image/2minrelief.html"/>
        <ows:WGS84BoundingBox>
          <ows:LowerCorner>-180.0 -90.0</ows:LowerCorner>
          <ows:UpperCorner>+180.0 +90.0</ows:UpperCorner>
        </ows:WGS84BoundingBox>
        <Identifier>ETOPO2</Identifier>
      </CoverageSummary>
      <CoverageSummary>
        <ows:Title>Global 30 Second Elevations</ows:Title>
        <ows:Abstract>GTOPO30 is a global digital elevation model (DEM)
resulting from a collaborative effort led by the staff at the U.S.
Geological Survey's EROS Data Center in Sioux Falls, South Dakota.
Elevations in GTOPO30 are regularly spaced at 30-arc seconds
(approximately 1 kilometer).  GTOPO30 was developed to meet the needs of
the geospatial data user community for regional and continental scale
topographic data.</ows:Abstract>
        <ows:Metadata
xlink:href="http://edcdaac.usgs.gov/gtopo30/README.html"/>
        <ows:WGS84BoundingBox>
          <ows:LowerCorner>-180.0 -90.0</ows:LowerCorner>
          <ows:UpperCorner>+180.0 +90.0</ows:UpperCorner>
        </ows:WGS84BoundingBox>
        <Identifier>GTOPO30</Identifier>
      </CoverageSummary>
      <CoverageSummary>
        <ows:Title>Two Minute Shaded Relief</ows:Title>
        <ows:Abstract>This image was generated from digital data bases
of seafloor and land elevations on a 2-minute latitude/longitude grid (1
minute of latitude = 1 nautical mile, or 1.852 km). Assumed illumination
is from the west; shading is computed as a function of the east-west
slope of the surface with a nonlinear exaggeration favoring low-relief
areas. A Cylindrical Equidistant projection was used for the world image,
which spans 360 degrees of of longitude from 180 West eastward to 180
East; latitude coverage is from 90 degrees North to 90 degrees South. The
resolution of the gridded data varies from true 2-minute for the
Atlantic, Pacific, andIndian Ocean floors and all land masses to 5
minutes for the Arctic Ocean floor. Clicking on a square above brings up
a 512 x 512 pixel color relief image of the 45 degree area selected,
clicking on the 512 x 512 image brings up the full-resolution 1350 x 1350
pixel (roughly 3 mb) color image of the area.</ows:Abstract>
        <ows:Metadata
xlink:href="http://www.ngdc.noaa.gov/mgg/image/2minrelief.html"/>
        <ows:WGS84BoundingBox>
          <ows:LowerCorner>-180.0 -90.0</ows:LowerCorner>
          <ows:UpperCorner>+180.0 +90.0</ows:UpperCorner>
        </ows:WGS84BoundingBox>
        <Identifier>RELIEF</Identifier>
      </CoverageSummary>
<SupportedCRS>urn:ogc:def:crs,crs:EPSG:6.3:4267</SupportedCRS>
      <SupportedCRS>urn:ogc:def:crs,crs:EPSG:6.3:4269</SupportedCRS>
      <SupportedCRS>urn:ogc:def:crs,crs:EPSG:6.3:4326</SupportedCRS>
      <SupportedFormat>image/tiff</SupportedFormat>
      <SupportedFormat>image/tiff;
PhotometricInterpretation=RGB</SupportedFormat>
      <SupportedFormat>image/png</SupportedFormat>
```

```
        <SupportedFormat>image/png;
PhotometricInterpretation=PaletteColor</SupportedFormat>
        <SupportedFormat>image/png;
PhotometricInterpretation=RGB</SupportedFormat>
        <SupportedFormat>image/gif</SupportedFormat>
        <SupportedFormat>image/jpeg</SupportedFormat>
        <SupportedFormat>image/ppm</SupportedFormat>
    </Contents>
</Capabilities>
```

C.3   DescribeCoverage operation XML encoding examples

These examples are provided in Subclause 9.5.

C.4   GetCoverage XML operation XML encoding examples

These examples are provided in Subclause 10.4.

Annex D
(informative)

UML model

D.1   Introduction

This annex provides a UML model of the WCS interface, using the OGC/ISO profile of UML summarized in Subclause 5.2 of the OWS Common [OGC 06-121r3].

Figure D.1 is a UML diagram summarizing the WCS interface. This class diagram shows that the WebCoverageService class inherits the getCapabilities operation from the abstract OGCWebService class, which is common to all OGC Web Services. The WebCoverageService class adds the getCoverage and describeCoverage operations. (The capitalization of class, operation, and data type names uses the OGC/ISO profile of UML.)
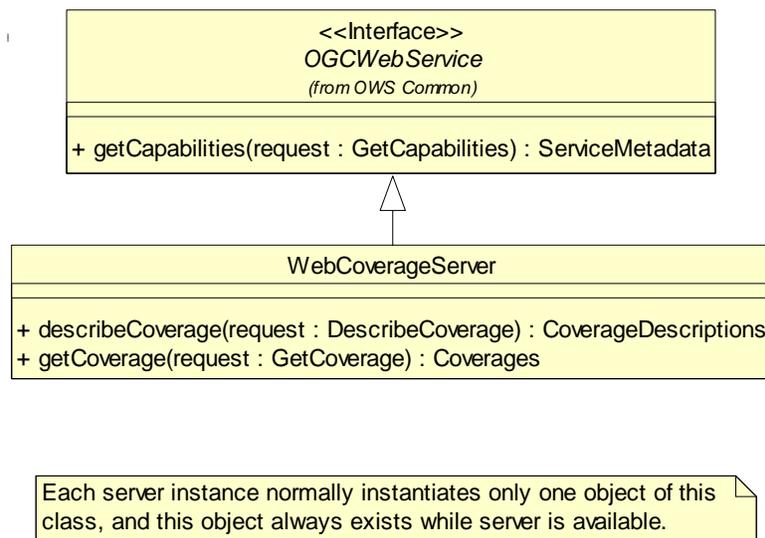
Figure D.1 — WCS interface UML class diagram

Each of the three operations uses a request and a response data type, each of which is defined by one or more additional UML classes. The following subclauses provide a more complete UML model of the WCS interface, adding UML classes defining the operation request and response data types.

D.2   UML packages

The WCS interface UML model is organized in nine packages, as shown in the package diagram in Figure D.2. These nine WCS-specific packages make use of most OGC Web Service Common (OWS Common) packages, some ISO 19111 packages and the ISO 19108 package. Only the packages used directly are shown, without listing the classes in the OWS

Common packages. This package diagram shows the dependencies among the various packages
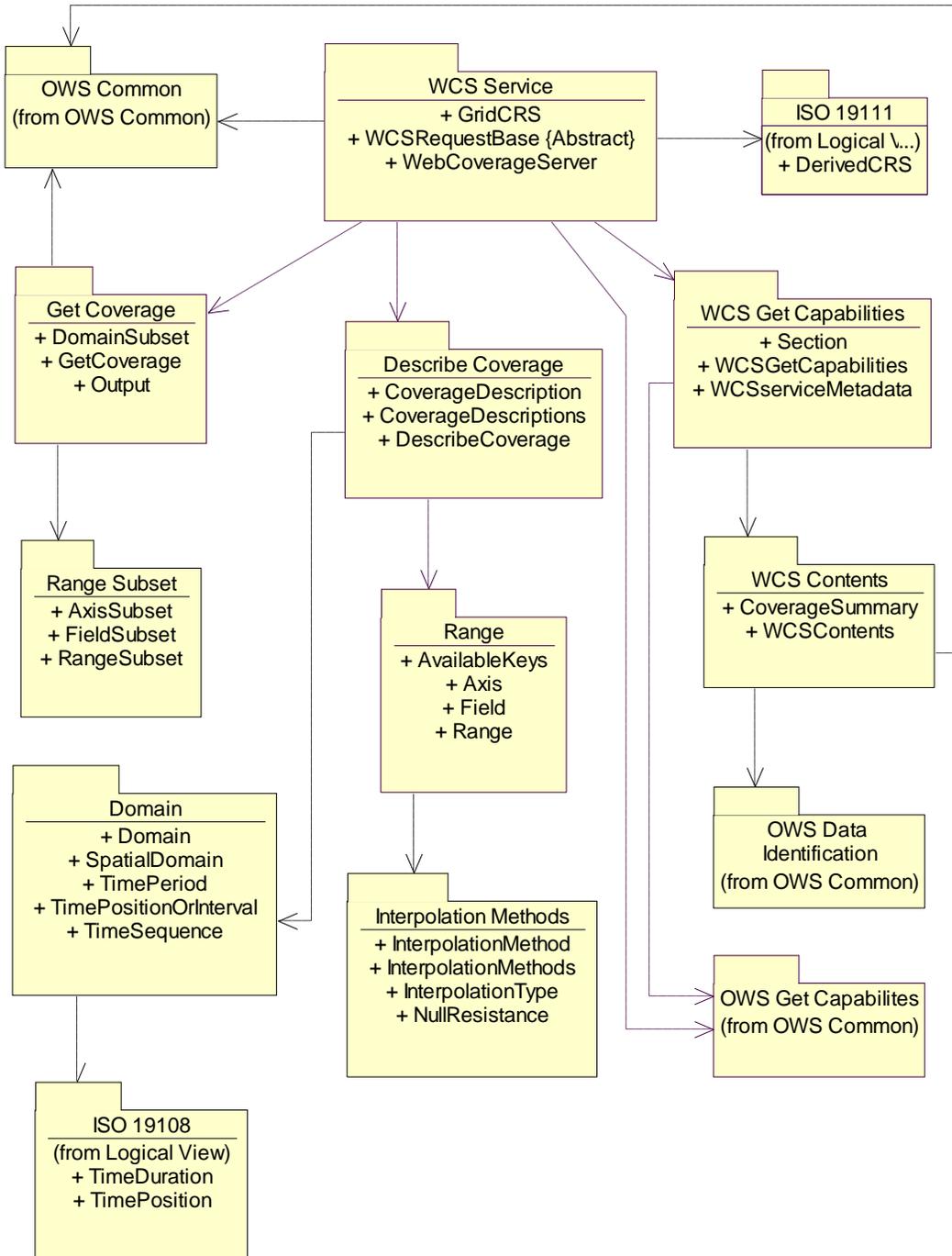
Figure D.2 — WCS interface package diagram

Eight of the WCS-specific packages shown in Figure D.2 are described in the following subclauses. The Coverages package is described in Subclause H.2.1, and the Interpolation Methods package is described in Subclause H.3.1.

D.3   WCS Service package

The WCS Service package is shown in the class diagram in Figure D.3. This diagram does not show the classes used by the three operation requests and responses, which are shown (with part of this package) in the Get Coverage, Describe Coverage, and WCS GetCapabilities packages. This diagram also shows three used classes from the OGC GetCapabilities, OGC Common, and ISO 19111 packages.
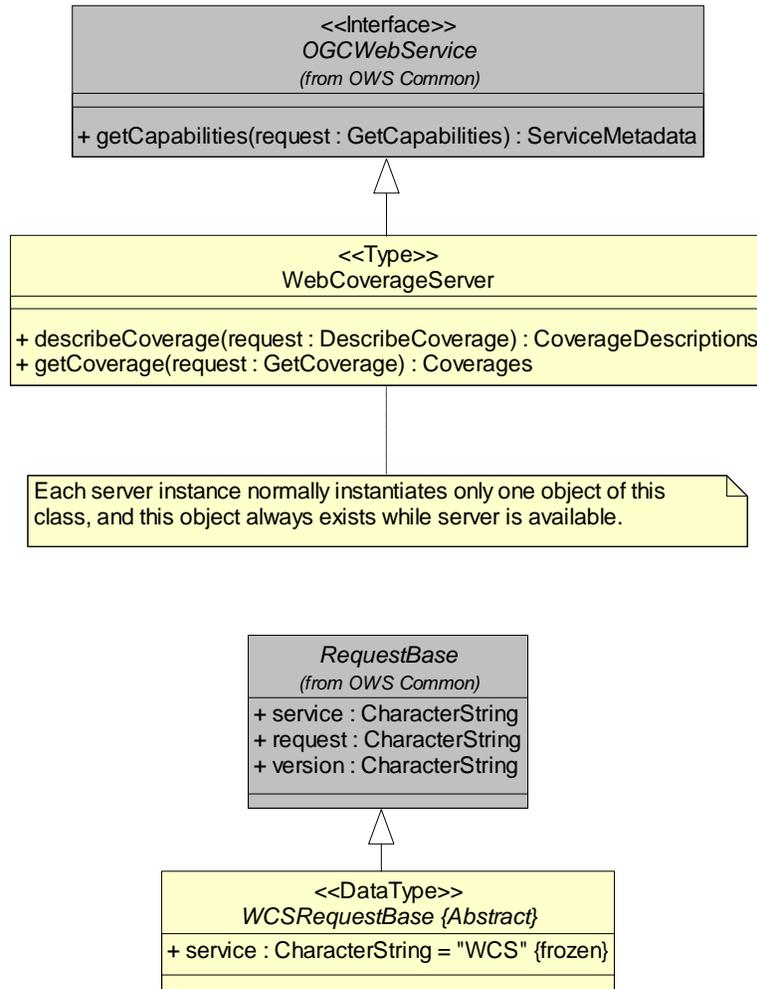
Figure D.3 — WCS Service package class diagram

D.4   Get Coverage package

The Get Coverage package is shown in the class diagram in Figure D.4. This diagram does not show details of the RangeSubset class, which is in the Range Subset package that is detailed in the following subclause. This diagram also shows several used classes from other packages. The three classes introduced by this package are further defined by Table 22 through Table 24 in this document.

73
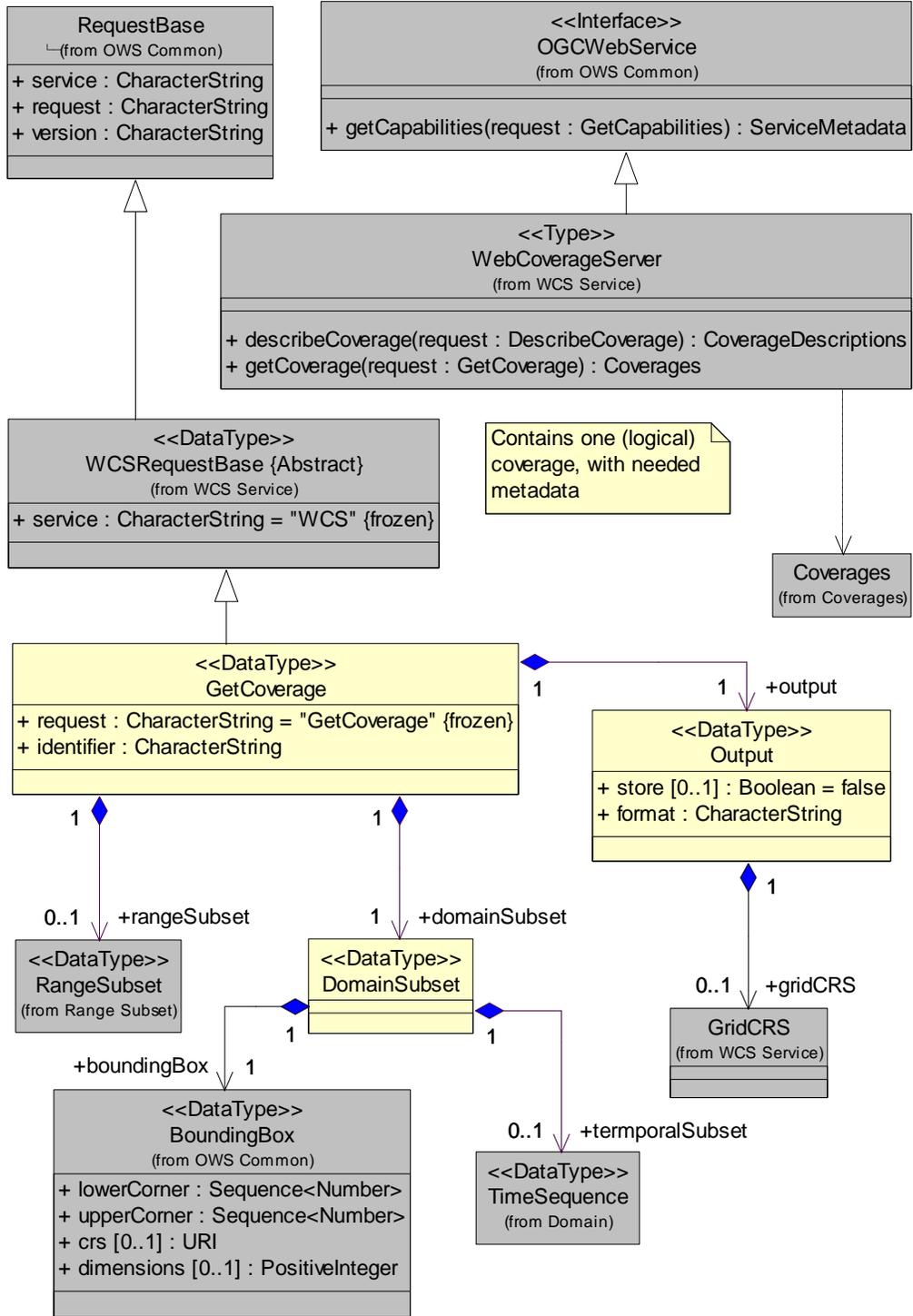
Figure D.4 — Get Coverage package class diagram

D.5   Range Subset package

The Range Subset package is shown in the class diagram in Figure D.5. This diagram also shows two used classes from other packages. The three classes introduced by this package are further defined by Table 25 through Table 27 in this document.
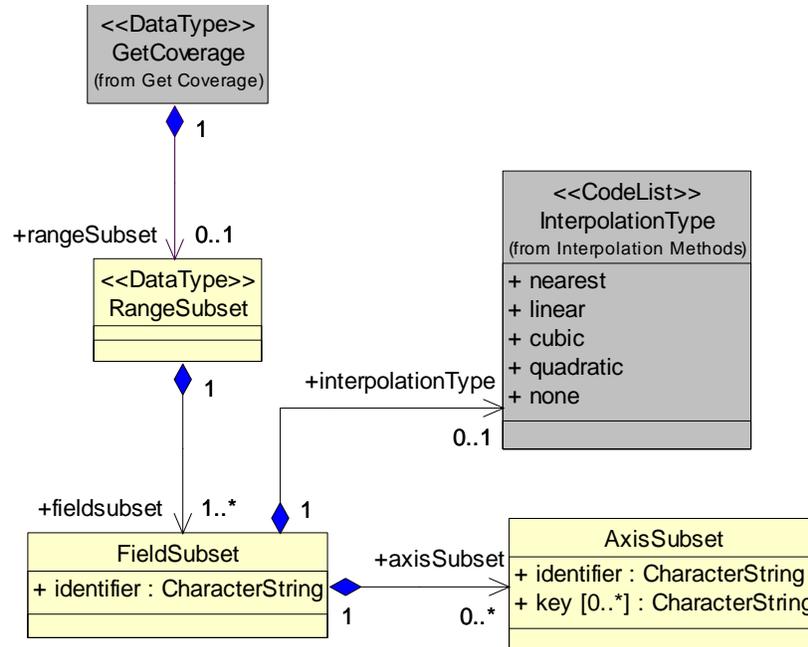
Figure D.5 — Range Subset package class diagram

D.6   Describe Coverage package

The Describe Coverage package is shown in the class diagram in Figure D.6. This diagram does not show details of the Domain and Range classes, which are in the Domain and Range packages that are detailed in following subclauses. This diagram also shows the some used classes from other packages. The three classes introduced by this package are further defined by Table 8 through Table 11 in this document.
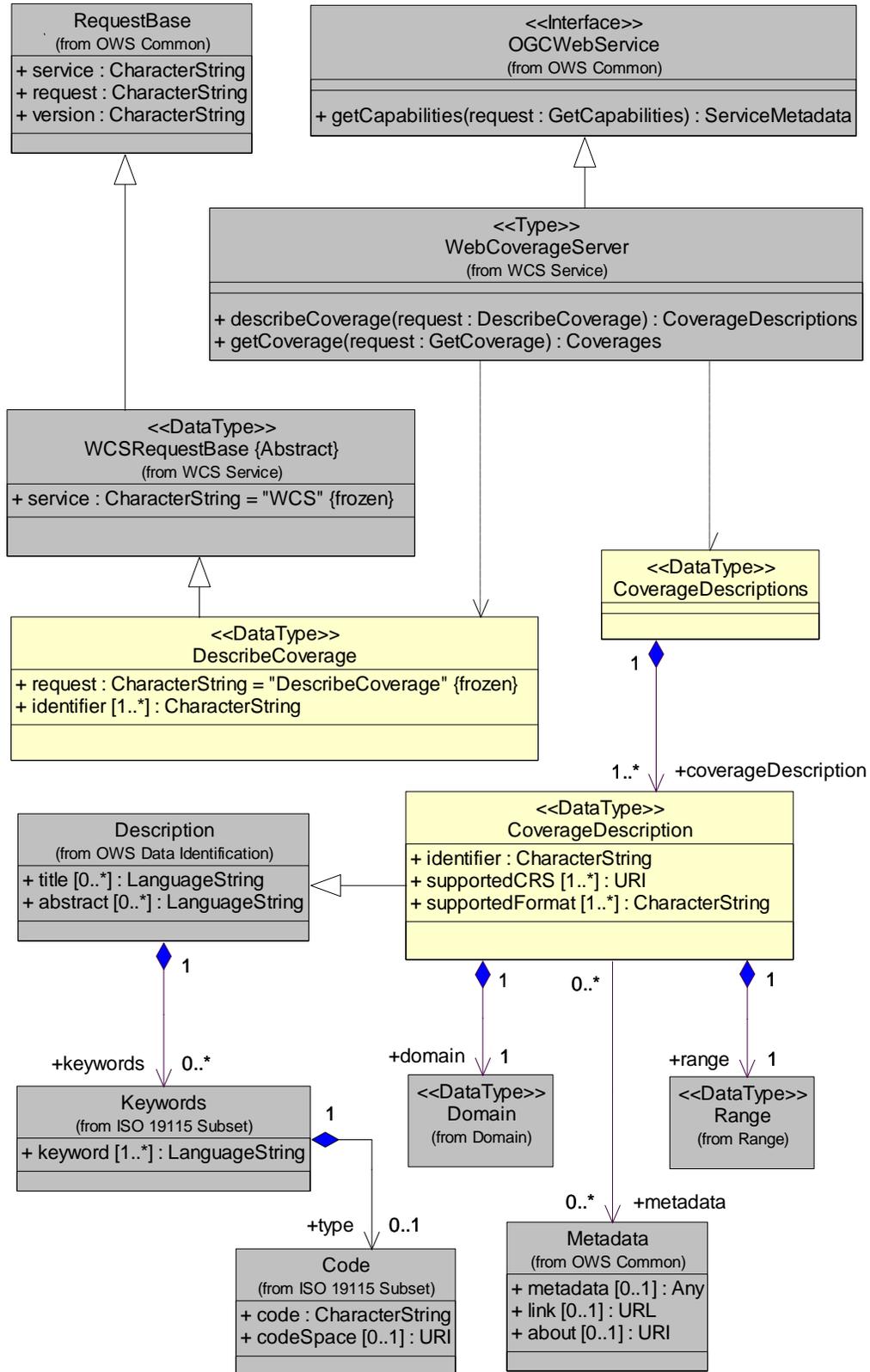
Figure D.6 — Describe Coverage package class diagram

D.7   Domain package

The Domain package is shown in the class diagram in Figure D.7. This diagram also shows one class from the Describe Coverage Package. The five classes introduced by this package are further defined by Table 12 through Table 16 in this document.
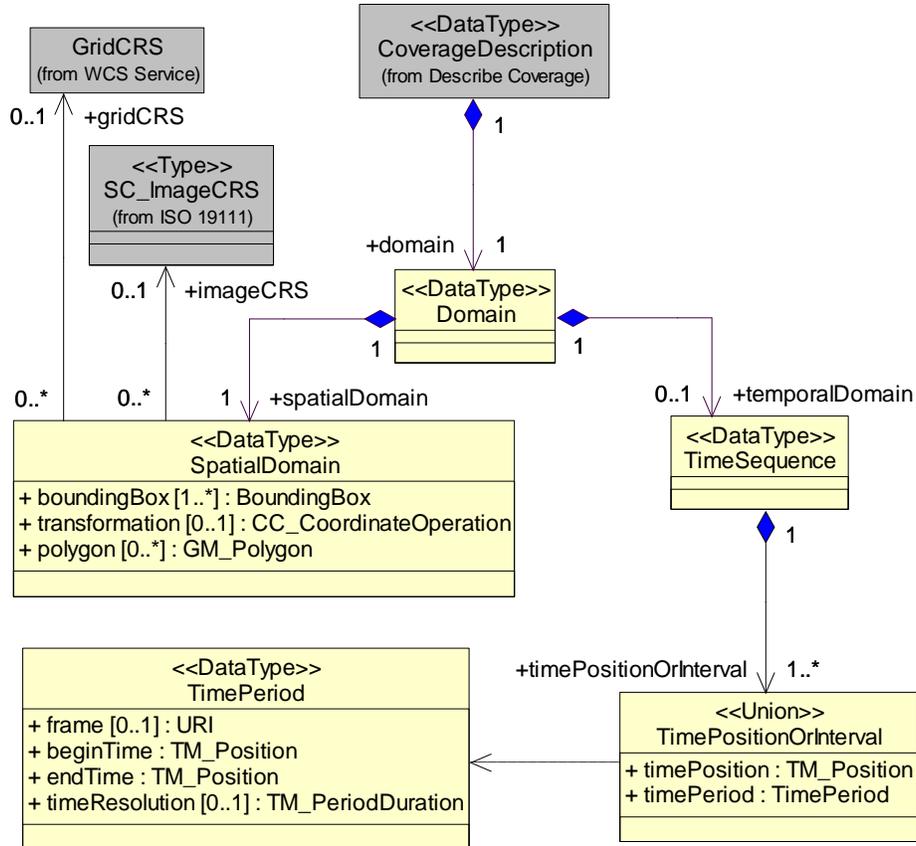
Figure D.7 — Domain package class diagram

D.8   Range package

The Range package is shown in the class diagram in Figures D.8 and D.9. This diagram also shows several used classes from other packages. The four classes introduced by this package are further defined by Table 17 through Table 20 in this document.

77
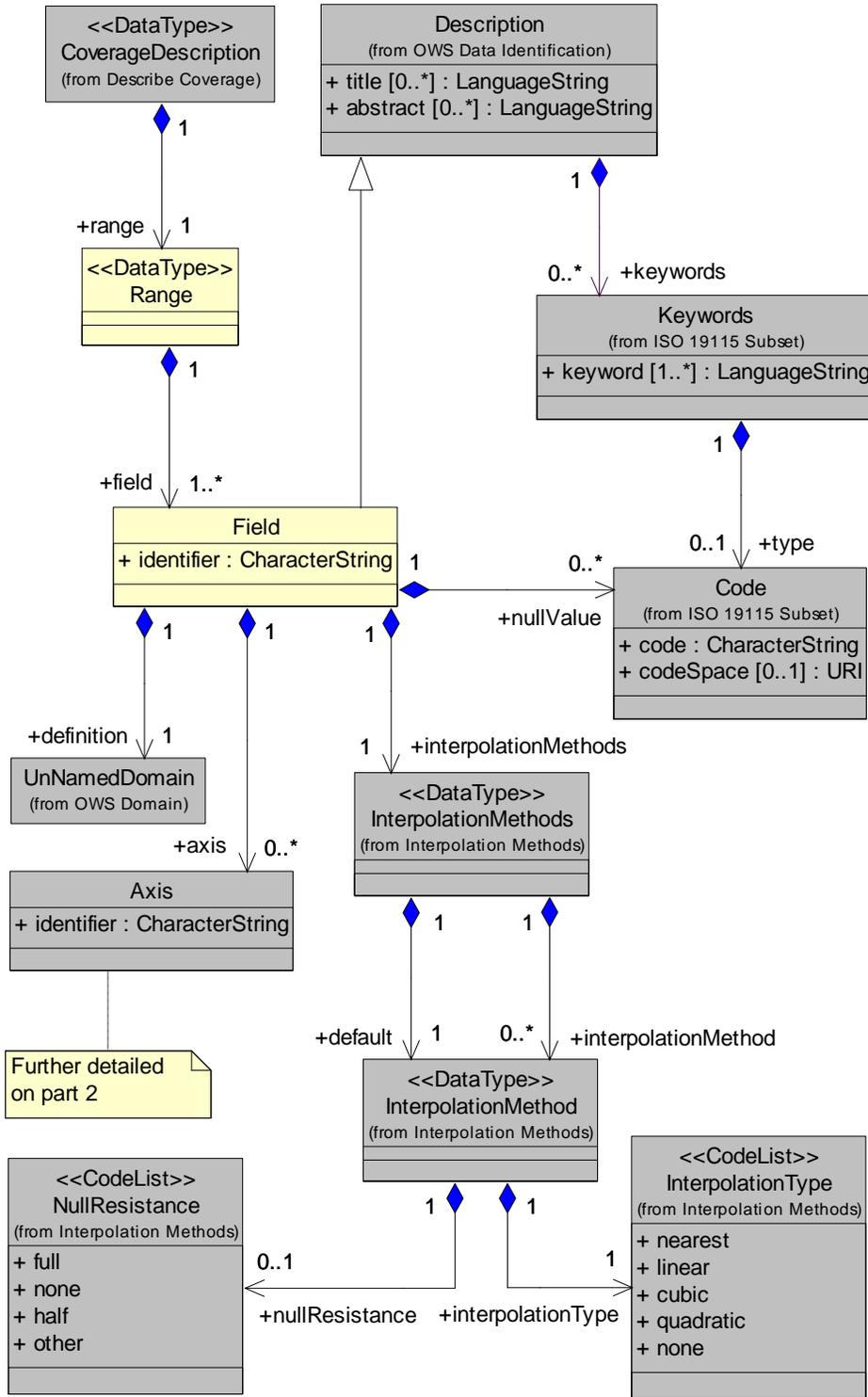
Figure D.8 — Range package class diagram, part 1
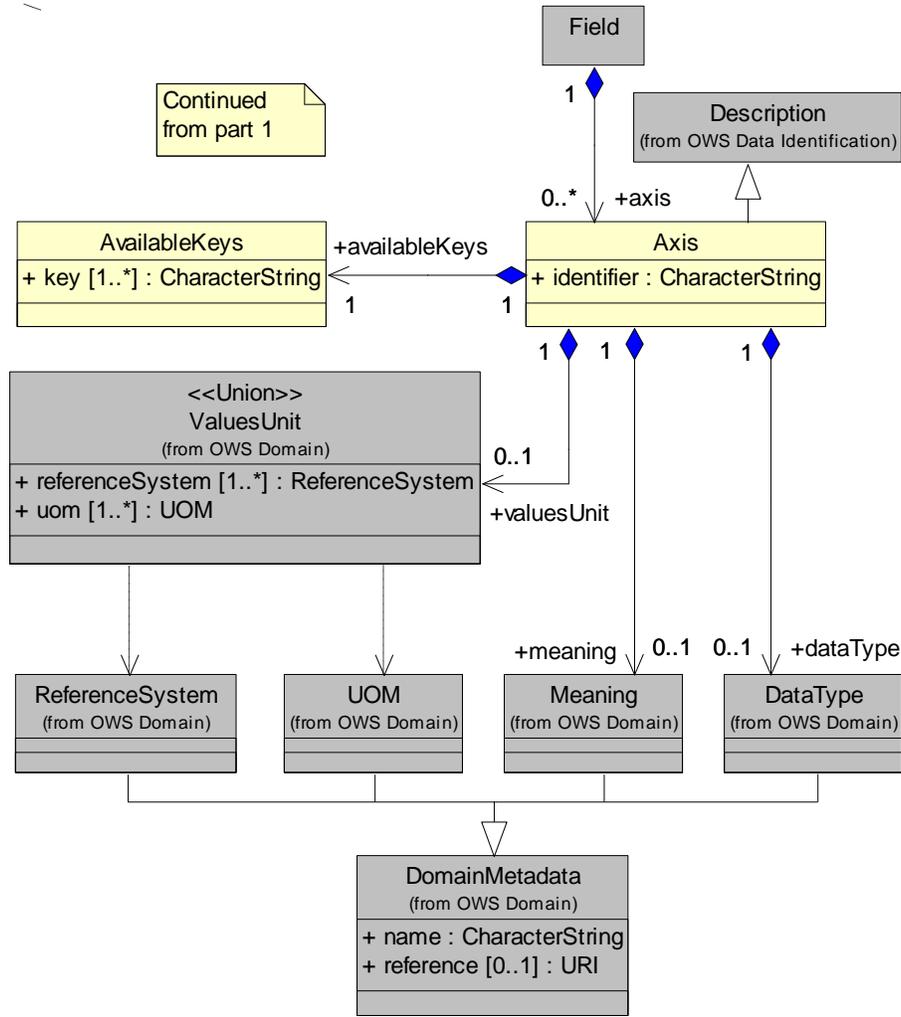
Figure D.9 — Range package class diagram, part 2

D.9 WCS Get Capabilities package

The WCS Get Capabilities package is shown in the class diagram in Figure D.10. This diagram does not show details of the WCSContents class, which is in the WCS contents package that is detailed in the following subclause. This diagram also shows several used classes from other packages.
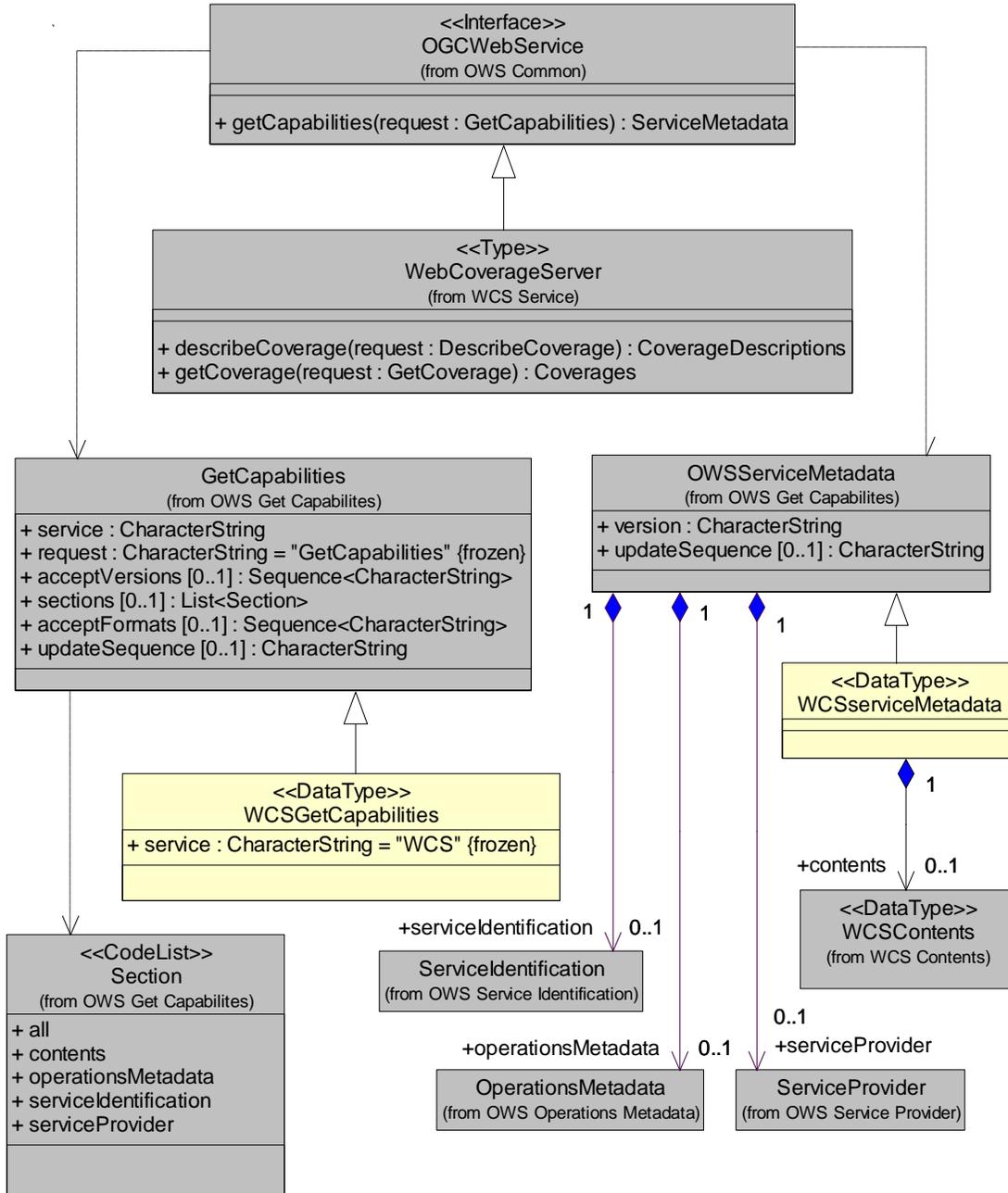
Figure D.10 — WCS Get Capabilities package class diagram

D.10   WCS Contents package

The WCS Contents package is shown in the class diagram in Figure D.11. This diagram also shows several used classes from other packages. The two classes introduced by this package are further defined by Table 5 and Table 6 in this document.
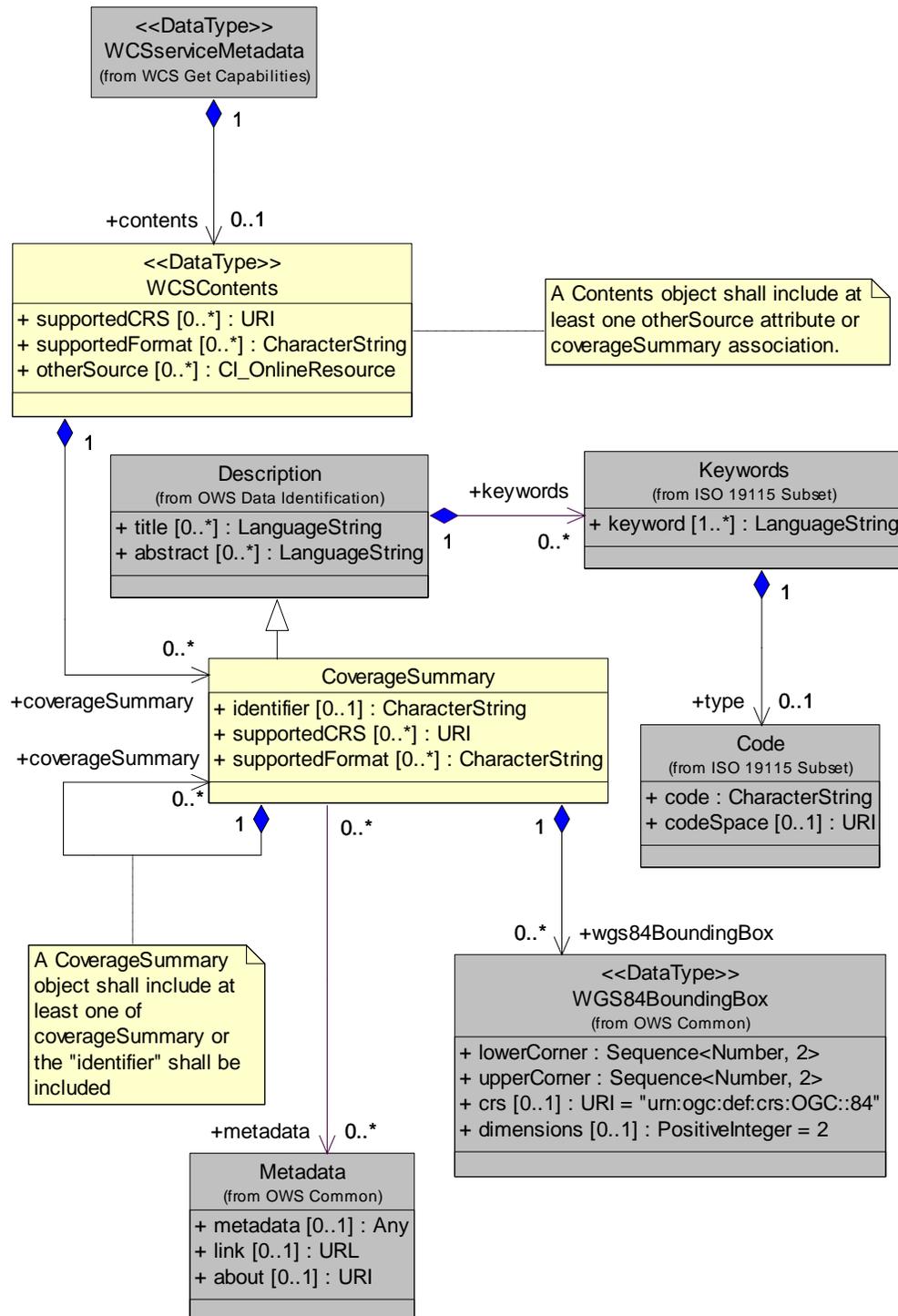

**Figure D.11 — WCS Contents package class diagram**

Annex E
(normative)

SOAP transfer

All compliant WCS servers may implement SOAP 1.2 transfer of all WCS operation requests and responses, using the XML encodings specified in the body of this document. When SOAP is implemented, the SOAP Request-Response message exchange pattern shall be used with the HTTP POST binding.

For SOAP transfer, each XML-encoded operation request shall be encapsulated in the body of a SOAP envelope, which shall contain only a body and only this request in that body. Similarly, each XML-encoded operation response shall be encapsulated in the body of a SOAP envelope, which shall contain only this response in that body. A WCS server shall return operation responses and error messages using only SOAP transfer when the operation request is sent using SOAP.

All compliant WCS servers shall specify the URLs to which SOAP operation requests may be sent, within the OperationsMetadata section of a service metadata (Capabilities) XML document, as specified in Subclause 8.3.2.

When an error is detected while processing an operation request encoded in a SOAP envelope, the WCS server shall generate a SOAP response message where the content of the Body element is a Fault element containing an ExceptionReport element. This shall be done using the following XML fragment:

```xml
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Body>
      <soap:Fault>
         <soap:Code>
            <soap:Value>soap:Server</soap:Value>
         </soap:Code>
         <soap:Reason>
            <soap:Text>A server exception was encountered.<soap:Text>
         </soap:Reason>
         <soap:Detail>
            <ows:ExceptionReport>
               ...
            </ows:ExceptionReport>
         </soap:Detail>
      </soap:Fault>
   </soap:Body>
</soap:Envelope>
```

The Code element shall have the Value "soap:server" indicating that this is a server exception. The Reason element shall have the Text "Server exception was encountered." This fixed string is used since the details of the exception shall be specified in the Detail element using an ows:ExceptionReport element as specified in OWS Common [OGC 06-121r3].

Annex F
(normative)

GridCRS

F.1   Introduction

This annex specifies a GridCRS data structure for encoding the definitions of rectangular grids defined within another CRS. This annex also specifies three instances of the gml:OperationMethod element that specify rectangular grids (more precisely, parallelogram grids in the other CRS). Such a gml:OperationMethod is referenced by the GridType that is included in the specified GridCRS data structure.

NOTE   This GridCRS data structure is specified here because it is now only used in WCS. However, this GridCRS is likely to be useful in other OWS specifications as well, such as the draft Web Coordinate Transformation Service (WCTS) [OGC 05-013]. When or before this GridCRS is used elsewhere, the GridCRS specification in this annex should be moved into a separate OGC Implementation Specification.

F.2   GridCRS

F.2.1   GridCRS contents

This subclause specifies a GridCRS data structure for encoding the definition of a rectangular grid that is specified within another CRS, called the GridBaseCRS. This GridCRS data structure is simplified from the DerivedCRS element that is defined in GML 3.1.1, to define such a rectangular grid in the GridBaseCRS.

NOTE 1   The GridCRS XML element can be referenced remotely as the source or target CRS of a CoordinateOperation element. However, it is NOT defined by a GML Application Schema, because such a definition would be too complex for use in WCS. This GridCRS element is thus NOT in the gml:_CRS substitutionGroup, and so it cannot be included as the GridBaseCRS of a DerivedCRS, or included in a GML dictionary. However, this GridCRS element does use a few parts of GML, which are included in the approved GML 3.1.1 CRS support profile [OGC 05-094r1].

The parameters contained in the GridCRS data structure are described in Figure F.1 and specified in Table F.1.

GridCRS
+ srsName [0..1] : URI
+ gridBaseCRS : URI
+ gridType [0..1] : URI = "urn:ogc:def:method: WCS:1.1:2dSimpleGrid"
+ gridOrigin [0..1] : Sequence<Double> = "0 0"
+ gridOffsets : Sequence<Double>
+ gridCS [0..1] : URI = "urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS
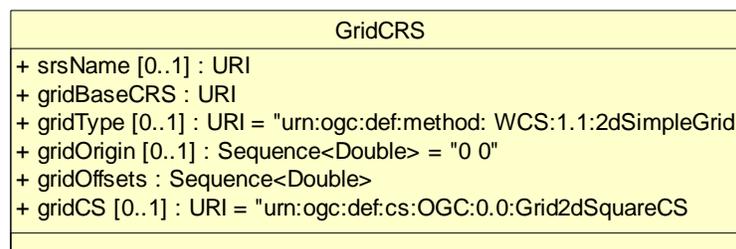
Figure F.1 — GridCRS class diagram

Table F.1 — Parameters in GridCRS data structure

Names [a]	Definition	Data type and value	Multiplicity and use
srsName srsName	Name or identifier of this GridCRS	URI	Zero or one (optional) Include when needed so this GridCRS can be referenced
gridBaseCRS GridBaseCRS	Reference to definition of base CRS used by this GridCRS	URI [b]	One (mandatory)
gridType GridType	Reference to definition of operation method that specifies quadrilateral grid in GridBaseCRS	URI Default is "**urn:ogc:def:method:WCS:1.1:2dSimpleGrid**"	Zero or one (optional) Include when not default GridType
gridOrigin GridOrigin	Position coordinates of origin of this GridCRS in the GridBaseCRS	Sequence<Double> Default is "**0.0 0.0**"	Zero or one (optional) Include when not default origin
gridOffsets GridOffsets	Position offsets between adjacent points in rectangular grid, in each dimension of GridCS in each dimension of GridBaseCRS	Sequence<Double>	One (mandatory)
gridCS GridCS	Reference to definition of 2D or 3D CartesianCS used by this GridCRS	URI Default is "**urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS**"	Zero or one (optional) Include when not default GridCS

a   See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.

b   The referenced GridBaseCRS can be a 2D or 3D spatial CRS, most frequently 2D. This GridBaseCRS can be a 2D gml:ProjectedCRS, a 2D or 3D gml:GeographicCRS, or a gml:CompoundCRS that combines a 2D gml:ProjectedCRS with a 1D gml:VerticalCRS, as specified in the GML 3.1.1 common CRSs profile [OGC 05-095r1]. This GridBaseCRS can alternately be a 2D or 3D gml:ImageCRS or another wcs:GridCRS. The two axes of a gml:GeographicCRS are usually named geodetic latitude and geodetic longitude, with coordinates listed in that order and with both latitude and longitude values given in decimal degrees. The two axes of a gml:ProjectedCRS are often named Easting and Northing, not always in that order, with values often given in metres.

In this GridCRS, the GridType remotely references one of several fixed XML documents that each specifies an OperationMethod object. There are many OperationMethod objects that might be referenced, three of which are specified in Subclause F.3.

F.2.2   GridCRS coordinates

Each coordinate value in a GridCRS is the index of a position in the grid coverage, with units of Grid Spacings, see Subclause 10.9 of [OGC 05-096r1]. (Coordinate values in a GridCRS are NOT in the GridBaseCRS.) To refer to a recorded pixel, each coordinate value will be an integer. However, to refer to a general position in a grid coverage, each coordinate value may include a fractional value to reference positions between recorded pixels.

The origin of a GridCRS may be at either of the following alternatives:

a) The origin may be the first pixel recorded for that grid coverage (similar to an ImageCRS). In this case, negative coordinate values shall not be used, since they would reference pixels that do not exist. Also, the GridCRS definition identifier

Copyright © 2008 Open Geospatial Consortium, Inc. All Rights Reserved.

should be changed when its first pixel is changed, from the origin of an offered coverage to the origin of an output coverage that is derived from that offered coverage. Specifically, the GridOrigin value(s) should be changed.

b) The origin may be located at any grid point in the specified grid. In this case, negative and positive coordinate index values may be used, since they reference pixel positions that are defined, but may not be recorded in the grid coverage. In this case, the GridCRS origin pixel (with 0 valued indices) may be anywhere in the pixels recorded, or may be anywhere outside the pixels recorded, and thus not recorded. In both cases, the same GridCRS definition may be used for grid coverages recorded with different first pixels, from the same or different original coverages or images. Also, the GridCRS definition identifier should be changed when any other part of the GridCRS definition is changed, either the GridOrigin, GridOffsets, or GridBaseCRS.

F.2.3   Default values

As indicated above, the definition of a GridCRS contains five primary parts. However, two of these parts will often have the same values. Those same values will be the values needed for the most common case, namely a 2D grid defined in a 2D CRS, without grid rotation or skewing. Furthermore, when included in the output part of a GetCoverage operation request, a third part will often have the same value, namely a "0.0 0.0" origin position. Default values are thus specified for these three parts, to allow omission of specific values for these parts in the most common cases. These default values are as specified in Table F.2.

Table F.2 — GridCRS default values

GridCRS part	Default value (XML encoded)
GridType	\<GridType\>urn:ogc:def:method:WCS:1.1:2dSimpleGrid\</GridType\> [a]
GridOrigin	\<GridOrigin\>0.0 0.0\</GridOrigin\>
GridCS	\<GridCS\>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS\</GridCS\> [b]
a This URN and its meaning are specified in Subclause F.3.4.	
b This URN and its meaning are specified in Subclause 10.4 of [OGC 05-096r1].	

F.2.4   XML encoding

The definition of a GridCRS will often be encoded in XML. This XML encoding shall use the XML Schema fragment specified in wcs:GridCRS.xsd.

EXAMPLE 1   An example XML fragment defining a 2D GridCRS without grid rotation or skewing is:

```
<GridCRS>
  <gml:srsName>2D Grid CRS Example 1</gml:srsName>
  <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:63266405</GridBaseCRS>
  <GridType>urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs</GridType >
  <GridOrigin>1.0 2.0</GridOrigin>
  <GridOffsets>0.1 0 0 0.2</GridOffsets>
  <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS >
</GridCRS>
```

NOTE 1   In the above example, the center two Offsets values are zero because there is no grid rotation or skewing relative to the underlying CRS axes (meridians and parallels). These offset values define a grid with a point spacing of 0.1 x 0.2 degree (latitude x longitude, as defined by EPSG 63266405)

EXAMPLE 2   An example XML fragment defining a 2D GridCRS with rotation (but no skewing) in the GridBaseCRS is:

```
<GridCRS>
    <gml:srsName>2D Grid CRS Example 2</gml:srsName>
    <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:32618</GridBaseCRS >
    <GridType>urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs</GridType>
    <GridOrigin>1.0 2.0</GridOrigin>
    <GridOffsets>0.0707 -0.0707 0.1414 0.1414</GridOffsets>
    <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS>
</GridCRS>
```

NOTE 2   In the above example, the center two Offsets values are non-zero because there is 45 degrees grid rotation clockwise relative to the easting/northing axes of the GridBaseCRS (WGS84 UTM meters, Zone 18N – denoted EPSG 32618). The grid defined by these offsets has a point spacing of 0.1x0.2 meter.

NOTE 3   These two example XML fragments reference the fixed XML document that specifies the 2dGridIn2dCrs given in Subclause F.3.2.

F.2.5   KVP encoding

This GridCRS may be KVP encoded in a GetCoverage operation request. That KVP encoding shall be as specified in Table F.3.

Table F.3 — GridCRS encoded as Key Value Pairs

Name and example	Optionality & use	Definition and format
GridBaseCRS=urn:ogc:def:crs: EPSG:6.6:32618	Optional, include all five GridCRS parameters, except for defaulted parameters, to request output in a CRS other than the ImageCRS or GridCRS of offered coverage	Reference to GridBaseCRS of desired output GridCRS, a URN or URL [a]
GridType=urn:ogc:def:method: WCS:1.1:2dGridIn2dCrs		Reference to grid type of desired output GridCRS, a URN or URL [a, b]
GridCS=urn:ogc:def:cs:OGC:0. 0:Grid2dSquareCS		Reference to coordinate system of desired output GridCRS, a URN or URL [a, c]
GridOrigin=1.0,2.0		Position coordinates of one possible grid origin, in GridBaseCRS of desired output GridCRS [a, d]
GridOffsets=0.0707, -0.0707,0.1414,0.1414&		Offsets between adjacent grid points, in GridBaseCRS of desired output GridCRS [a, e]

a The GridType parameter shall reference the known definition of a coordinate OperationMethod that defines a grid in the referenced GridBaseCRS. That GridType definition shall specify the meanings of the other "Grid…" parameters listed, including the number of dimensions in each other "Grid…" parameter.

b This GridType may be omitted when the GridBaseCRS is 2D and the grid is not rotated or skewed in the GridBaseCRS, in which case the default value shall be "urn:ogc:def:method:WCS:1.1:2dSimpleGrid".

c This GridCS may be omitted when the GridBaseCRS is 2D (in a 2D or 3D GridBaseCRS), in which case the default value shall be "urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS".

d This GridOrigin may be omitted when the GridBaseCRS is 2D, in which case the default value shall be "0.0 0.0".

e When the GridType is "urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs", there shall be four values in this GridOffsets. The center two of these offsets will be zero when the GridCRS is not rotated or skewed in the GridBaseCRS.

NOTE 1   This KVP encoding does not include the srsName of the GridCRS, since KVP encoding is only used in the Output part of a GetCoverage operation request, and any GridCRS defined there is never referenced.

EXAMPLE 1   Here is an example 2D GridCRS definition without grid rotation or skewing, defined using KVP encoding:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:63266405&
GridType=urn:ogc:def:method:WCS:1.1:2dSimpleGrid&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=0,0&
```

```
GridOffsets=0.1,0.2&
```

NOTE 2   This example and the next define a grid in the EPSG:63266405 CRS, that is, WGS84 latitude and longitude expressed in decimal degrees. The grid spacing is 0.1 degree of latitude and 0.2 degree of longitude.

EXAMPLE 2   Here is the same example GridCRS definition, taking advantage of all three applicable defaults:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:63266405&
GridOffsets=0.1,0.2&
```

EXAMPLE 3   Here is an example 2D GridCRS with 45 degrees clockwise rotation (but no skewing) in the GridBaseCRS, defined using KVP encoding:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=0,0
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

NOTE 3   This example and the next define a grid in the EPSG:32618 CRS, that is, WGS84 UTM Zone 18 North easting and northing expressed in meters. The grid spacing is 0.1m (NW-SE) and 0.2m (SW-NE)

EXAMPLE 4   Here is the same GridCRS definition, taking advantage of the two applicable defaults:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

F.3   Operation methods for rectangular grids

F.3.1   Introduction

This subclause specifies instances of the gml:OperationMethod element that specify rectangular grids in a manner similar to the gml:RectifiedGrid element. As previously stated, such a gml:OperationMethod is expected to normally be used by the GridType that is used by the specified GridCRS element. However, neither these gml:OperationMethods nor the GridCRSs using them define the extent of the grid (e.g., by extending gml:Grid as gml:RectifiedGrid does).

F.3.2   2D grid in 2D CRS (urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs)

The following OperationMethod specifies a 2D grid that can be rotated or skewed within a 2D CRS:

```
<?xml version="1.0" encoding="UTF-8"?>
<OperationMethod xmlns="http://www.opengis.net/gml"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/gml
../../../gml/3.1.1/Profiles/GridCRSs/1.0.0/gmlGridCRSsProfile.xsd"
gml:id="grid2dIn2dCrsMethod">
    <!-- Last updated 2007-06-06 -->
    <!-- ========================================= -->
    <methodName>2D Rectified Grid in 2D CRS Method</methodName>
    <methodID>
        <name>urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs</name>
    </methodID>
    <remarks>Method for use by a Coordinate Conversion defining a 2D
GridCRS in a 2D GridBaseCRS, based on CV_RectifiedGrid in ISO 19123 so
the grid can be rotated and skewed in this GridBaseCRS. </remarks>
    <methodFormula>The reverse direction (from the grid CRS to the
GridBaseCRS) shall be as specified by the equations:
```

```
    BaseX = origin(1) + offsets(1,1) * GridX + offsets(1,2) * GridY
    BaseY = origin(2) + offsets(2,1) * GridX + offsets(2,2) * GridY
where
    GridX, GridY are position coordinates in the grid 2D CRS
    BaseX, BaseY are position coordinates in the 2D base CRS
    origin(1), origin(2) are the two coordinates of the grid origin
position in the 2D base CRS
    offsets(1,1), offsets(1,2) are the grid point offsets of the first
grid axis in the 2D base CRS
    offsets(2,1), offsets(2,2) are the grid point offsets of the second
grid axis in the 2D base CRS

The forward direction (from the GridBaseCRS to the GridCRS) equations are
not included here. </methodFormula>
    <sourceDimensions>2</sourceDimensions>
    <targetDimensions>2</targetDimensions>
    <!-- ============================================ -->
    <usesParameter>
        <OperationParameter gml:id="origin2d">
            <parameterName>origin2d</parameterName>
            <parameterID>
                <name>urn:ogc:def:parameter:WCS:1.1:origin2d</name>
            </parameterID>
            <remarks>This parameter shall be encoded as a wcs:Origin
element, giving the two origin coordinates in the 2D base CRS of this
GridCRS. </remarks>
        </OperationParameter>
    </usesParameter>
    <!-- ============================================ -->
    <usesParameter>
        <OperationParameter gml:id="offsets2d">
            <parameterName>offsets2d</parameterName>
            <parameterID>
                <name>urn:ogc:def:parameter:WCS:1.1:offsets2d</name>
            </parameterID>
            <remarks>This parameter shall be encoded as a wcs:Offsets
element, giving the grid offsets in the two grid axes, each in the 2D
base CRS relative to the specified origin. This wcs:Offsets shall contain
four values, the first two values shall specify the grid offset for the
first grid axis, and the second pair of values shall specify the grid
offset for the second grid axis. </remarks>
        </OperationParameter>
    </usesParameter>
</OperationMethod>
```

This "2D grid in 2D CRS method" shall be referenced using the above-specified URN "**urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs**". Several examples in Subclauses F.2.2 and F.2.3 above reference this OperationMethod.

F.3.3   2D grid in 3D CRS (urn:ogc:def:method:WCS:1.1:2dGridIn3dCrs)

The following OperationMethod specifies a 2D grid within a 3D CRS:

```
<?xml version="1.0" encoding="UTF-8"?>
<OperationMethod xmlns="http://www.opengis.net/gml"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/gml
../../../gml/3.1.1/Profiles/GridCRSs/1.0.0/gmlGridCRSsProfile.xsd"
gml:id="grid2dIn3dMethod">
    <!-- Last updated 2006-02-28 -->
    <!-- ============================================ -->
```

```
    <methodName>2D Rectified Grid in 3D CRS Method</methodName>
    <methodID>
        <name>urn:ogc:def:method:WCS:1.1:2dGridIn3dCrs</name>
    </methodID>
    <remarks>Method for use by a Coordinate Conversion defining a 2D
GridCRS in a 3D GridBaseCRS, based on CV_RectifiedGrid in ISO 19123 so
the grid can be rotated and skewed in this GridBaseCRS. </remarks>
    <methodFormula>The reverse direction (from the GridCRS to the
GridBaseCRS) shall be as specified by the equations:
    BaseX = origin(1) + offsets(1,1) * GridX + offsets(2,1) * GridY
    BaseY = origin(2) + offsets(1,2) * GridX + offsets(2,2) * GridY
    BaseZ = origin(3) + offsets(1,3) * GridX + offsets(2,3) * GridY
where
    GridX, GridY are position coordinates in the 2D grid CRS
    BaseX, BaseY, BaseZ are position coordinates in the 3D base CRS
    origin(1), origin(2), origin(3) are the three coordinates of the grid
origin position in the 3D base CRS
    offsets(1,1),offsets(1,2), offsets(1,3) are the grid spacings of the
first grid axis in the 3D base CRS
    offsets(2,1), offsets(2,2), offsets(2,3) are the grid spacings of the
second grid axis in the 3D base CRS

The forward direction (from the GridBaseCRS to the GridCRS) equations are
not included here. </methodFormula>
    <sourceDimensions>3</sourceDimensions>
    <targetDimensions>2</targetDimensions>
    <!-- ============================================= -->
    <usesParameter>
        <OperationParameter gml:id="origin3d">
            <parameterName>origin3d</parameterName>
            <parameterID>
                <name>urn:ogc:def:parameter:WCS:1.1:origin3d</name>
            </parameterID>
            <remarks>This parameter shall be encoded as a wcs:Origin
element, giving the three origin coordinates in the 3D base CRS of this
GridCRS. </remarks>
        </OperationParameter>
    </usesParameter>
    <!-- ============================================= -->
    <usesParameter>
        <OperationParameter gml:id="offsets3d">
            <parameterName>offsets3d</parameterName>
            <parameterID>
                <name>urn:ogc:def:parameter:WCS:1.1:offsets3d</name>
            </parameterID>
            <remarks>This parameter shall be encoded as a wcs:Offsets
element, giving the grid offsets in the two grid axes, each in the 3D
base CRS relative to the specified origin. This wcs:Offsets shall contain
six values, the first three values shall specify the grid offset for the
first grid axis, and the second three values shall specify the grid
offset for the second grid axis. </remarks>
        </OperationParameter>
    </usesParameter>
</OperationMethod>
```

This "2D grid in 3D CRS method" shall be referenced using the above-specified URN
"**urn:ogc:def:method:WCS:1.1:2dGridIn3dCrs**".

EXAMPLE 1   Here is an example GridCRS that references this OperationMethod encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<GridCRS xmlns="http://www.opengis.net/wcs/1.1"
    xmlns:gml="http://www.opengis.net/gml"
```

```
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/wcs/1.1 ../wcsGridCRS.xsd"
    gml:id="example2dGridCrs2">
    <gml:srsName>2D Grid CRS Example 3</gml:srsName>
    <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:4326</GridBaseCRS>
    <GridType>urn:ogc:def:method:WCS:1.2:grid2dIn3dMethod</GridType>
    <GridOrigin>0 0 0</GridOrigin>
    <GridOffsets>0.0707 0.0707 0  -0.1414 0.1414 0</GridOffsets>
    <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS>
</GridCRS>
```

NOTE   This example and the next define a grid in the EPSG:7405 CRS (a 3D, Cartesian reference that uses east / north meters in the British National Grid, and vertical meters relative to the Ordnance Survey Datum). The six numbers in GridOffsets define a grid that is level to the ground plane but rotated at 45 degrees counterclockwise relative to the easting / northing axes, with a grid spacing of 0.1m (in the SW-NE direction) and 0.2m (NW-SE).

EXAMPLE 2   The same example GridCRS encoded in KVP is:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:7405&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn3dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=0,0,0&
GridOffsets=0.0707,0.0707,0,-0.1414,0.1414,0&
```

EXAMPLE 3   Here is the same example GridCRS encoded in KVP using the default GridCS and GridOrigin:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:7405&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn3dCrs&
GridOffsets=0.0707,0.0707,0,-0.1414,0.1414,0&
```

F.3.4   2D simple grid (urn:ogc:def:method:WCS:1.1:2dSimpleGrid)

The following OperationMethod specifies a 2D simple grid (with no rotation or skew) within a 2D CRS:

```
<?xml version="1.0" encoding="UTF-8"?>
<OperationMethod xmlns="http://www.opengis.net/gml"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/gml
../../../gml/3.1.1/Profiles/GridCRSs/1.0.0/gmlGridCRSsProfile.xsd"
gml:id="grid2dIn2dCrsMethod">
    <!-- Last updated 2007-06-06 -->
    <!-- ========================================= -->
    <methodName>2D Simple Grid Method</methodName>
    <methodID>
        <name>urn:ogc:def:method:WCS:1.1:2dSimpleGrid</name>
    </methodID>
    <remarks>Method for use by a Coordinate Conversion defining a simple
2D GridCRS in a 2D GridBaseCRS, where this grid is not rotated or skewed
in this GridBaseCRS. </remarks>
    <methodFormula>The reverse direction (from the grid CRS to the
GridBaseCRS) shall be as specified by the equations:
    BaseX = origin(1) + offsets(1) * GridX
    BaseY = origin(2) + offsets(2) * GridY
where
    GridX, GridY are position coordinates in the grid 2D CRS
    BaseX, BaseY are position coordinates in the 2D base CRS
    origin(1), origin(2) are the two coordinates of the grid origin
position in the 2D base CRS
    offsets(1) is the grid point offsets of the first grid axis in the 2D
base CRS
```

```
      offsets(2) is the grid point offsets of the second grid axis in the 2D
   base CRS

   The forward direction (from the GridBaseCRS to the GridCRS) equations are
   not included here. </methodFormula>
      <sourceDimensions>2</sourceDimensions>
      <targetDimensions>2</targetDimensions>
      <!-- =============================================== -->
      <usesParameter>
         <OperationParameter gml:id="origin2d">
            <parameterName>origin2d</parameterName>
            <parameterID>
               <name>urn:ogc:def:parameter:WCS:1.1:origin2d</name>
            </parameterID>
            <remarks>This parameter shall be encoded as a wcs:Origin
   element, giving the two origin coordinates in the 2D base CRS of this
   GridCRS. </remarks>
         </OperationParameter>
      </usesParameter>
      <!-- =============================================== -->
      <usesParameter>
         <OperationParameter gml:id="offsets2d">
            <parameterName>offsets2d</parameterName>
            <parameterID>
               <name>urn:ogc:def:parameter:WCS:1.1:offsets2d</name>
            </parameterID>
            <remarks>This parameter shall be encoded as a wcs:Offsets
   element, giving the grid offsets in the two grid axes, relative to the
   specified origin. This wcs:Offsets shall contain two values, the first
   value shall specify the grid offset for the first grid axis in the first
   GridBaseCRS axis, and the second value shall specify the grid offset for
   the second grid axis in the second GridBaseCRS axis. </remarks>
         </OperationParameter>
      </usesParameter>
   </OperationMethod>
```

This "2D simple grid method" shall be referenced using the above-specified URN
"**urn:ogc:def:method:WCS:1.1:2dSimpleGrid**".

EXAMPLE 1   Here is an example GridCRS that references this OperationMethod encoded in XML. (This
example and the next define a grid in the EPSG:63266405 CRS, that is, WGS84 latitude and longitude expressed
in decimal degrees.)

```
<?xml version="1.0" encoding="UTF-8"?>
<GridCRS xmlns="http://www.opengis.net/wcs/1.1"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/wcs/1.1 ../wcsGridCRS.xsd"
   gml:id="example2dGridCrs1">
   <gml:srsName>2D Grid CRS Example 1</gml:srsName>
   <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:4326</GridBaseCRS>
   <GridType>urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs</GridType>
   <GridOrigin>1.0 2.0</GridOrigin>
   <GridOffsets>0.1 0 0 0.2</GridOffsets>
   <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS>
</GridCRS>
```

NOTE   This 2D simple grid method uses a GridOffsets (KVP or XML) parameter containing only two
values: the grid spacing along each of its axes, expressed in the units and the order given by the underlying CRS
(in this case, degrees of latitude and longitude, in that order). Those two values correspond to the RESX and
RESY parameters used in the GetCoverage request KVP encoding in WCS 1.0.

EXAMPLE 2   Here is the same example GridCRS encoded in KVP:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:63266405&
```

```
GridType=urn:ogc:def:method:WCS:1.1:WCS:1.1:2dSimpleGrid&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=0,0&
GridOffsets=0.1,0.2&
```

EXAMPLE 2   Here is the same example GridCRS encoded in KVP, taking advantage of all three available
defaults:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:63266405&
GridOffsets=0.1,0.2&
```

Annex G
(informative)

GetCoverage use cases

G.1   Introduction

This annex describes a set of use cases for WCS GetCoverage operation requests and responses, followed by descriptions of how this WCS version can be used to satisfy these use cases. These use cases were developed and used by the WCS 1.1 RWG, and consider requesting spatial subsets of grid coverages, but do not discuss temporal subsets.

G.2   Use cases

G.2.1   Overview

These use cases are based on three conditions affecting what a server is required to do to respond to a GetCoverage operation request. These conditions may or may not be significant to the client. These three conditions are:

a) Is resampling of the offered coverage needed?

b) Is the offered coverage georectified or unrectified?

c) Is the desired spatial subset extent defined in ground (or object) or in grid (or image) coordinates?

These 10 use cases are summarized in Table G.1, assigning case numbers for all possible combinations of these three conditions.

Table G.1 — GetCoverage use case numbers for combinations of conditions

Offered coverage type Extent CRS type	Is resampling needed?	
	No resampling	Resampling

Georectified coverage
    Extent in ground CRS
    Extent in grid CRS
	1
	2
		6
		7

Unrectified and un-georeferenced coverage Extent in grid CRS [a, b]	3	8

Unrectified and georeferenced coverage
    Extent in grid CRS [b]
    Extent in ground CRS [b]
	4
	5
		9
		10

a   The extent cannot be defined in a ground CRS for an unrectified and un-georeferenced coverage. These use cases also apply when the georeferencing coordinate transformation(s) are not available to the server or cannot be used by the server,.

b   These use cases apply whether the offered unrectified coverage has an ImageCRS or a GridCRS with a GridBaseCRS that is an ImageCRS.

93

In order to allow a client to not know anything more than whether it is defining the extent in a ground or image CRS, the GetCoverage request design for these use cases should use the same request data structures for ALL supported use cases. For example, the GetCoverage request design should allow including either a ground or image CRS, of one subtype or another, in exactly the same places for ALL supported use cases.

In all these use cases, the server is required to be able to output metadata with the coverage, where all georeferencing information included within an image file is considered to be metadata.

The following subclauses further describe each use case.

G.2.2 Use case 1: No resampling of georectified coverage, defining extent in ground CRS

The GetCoverage request asks a server for an un-resampled subsection of a georectified image or other georectified grid coverage, defining the desired extent in a 2D ground CRS (GeographicCRS or ProjectedCRS). Notice that the specified extent could be skewed and with curved edges in the offered coverage. When such a skewed/curved extent is allowed in the offered coverage, the WCS server should interpret this extent as the minimum extent desired, and should extend that extent to the minimum rectangle surrounding that desired skewed/curved extent.

The needed response from the GetCoverage operation includes all:

a) Array of output coverage grid/pixel values

b) CRS of output coverage

c) Extent of output coverage

G.2.3 Use case 2: No resampling of georectified image, defining extent in grid CRS

The GetCoverage request asks a server for an un-resampled subsection of a georectified image, defining the desired extent in the grid CRS of the offered image.

The needed response from the GetCoverage operation includes all:

a) Array of output coverage grid/pixel values

b) CRS of output coverage

c) Extent of output coverage

G.2.4 Use case 3: No resampling of unrectified and un-georeferenced image

NOTE   This use case also applies when the georeferencing coordinate transformation(s) are not available to the server or cannot be used by the server.

The GetCoverage request asks a server for an un-resampled subsection of an unrectified and un-georeferenced image, defining the desired extent in the grid CRS of the unrectified offered image.

The needed response from the GetCoverage operation includes all:

a) Array of output coverage grid/pixel values

b) CRS of output image, at least identification of this CRS definition, for possible future use in georeferencing coordinate Transformation(s)

c) Extent of output image

G.2.5 Use case 4: No resampling of unrectified but georeferenced image, defining extent in grid CRS

The GetCoverage request asks a server for an un-resampled subsection of an unrectified but georeferenced image, defining the desired extent in the grid CRS of the unrectified offered image.

The needed response from the GetCoverage operation includes all:

a) Array of output coverage grid/pixel values

b) CRS of output image, at least identification of this CRS definition, for referencing in georeferencing coordinate Transformation(s)

c) Extent of output image

d) Georeferencing coordinate Transformation(s), between CRS of <u>output</u> image subsection and same ground CRS of georeferencing coordinate Transformation(s) for unrectified offered image

G.2.6 Use case 5: No resampling of unrectified but georeferenced image, defining extent in ground CRS

The GetCoverage request asks a server for an un-resampled subsection of an unrectified but georeferenced image, defining the desired extent in some 2D or 3D ground CRS (GeographicCRS, ProjectedCRS, or either compounded with a VerticalCRS). Notice that the specified extent could be skewed and with curved edges in the offered coverage. When such a skewed/curved extent is allowed in the offered coverage, the WCS server should interpret this extent as the minimum extent desired, and should extend that extent to the minimum rectangle surrounding that desired skewed/curved extent.

The needed response from the GetCoverage operation includes all:

a) Array of output coverage grid/pixel values

b) CRS of output image, at least identification of this CRS definition, for referencing in georeferencing coordinate Transformation(s)

c) Extent of output image

d) Georeferencing coordinate Transformation(s), between CRS of <u>output</u> image subsection and same Ground CRS of georeferencing coordinate Transformation(s) for unrectified offered image

G.2.7 Use case 6: Resampling of georectified coverage, defining extent in ground CRS

The GetCoverage request asks a server for a resampled (differently georectified) subsection of a georectified image or other georectified grid coverage, defining the desired extent in a 2D ground CRS (GeographicCRS or ProjectedCRS). See Addition A below.

The needed response from the GetCoverage operation includes all:

a) Array of output coverage grid/pixel values

b) CRS of output coverage

c) Extent of output coverage

G.2.8   Use case 7: Resampling of georectified image, defining extent in grid CRS

The GetCoverage request asks a server for a resampled (differently georectified) subsection of a georectified image, defining the desired extent in the grid CRS of the offered image. See Addition A below.

The needed response from the GetCoverage operation includes all:

a) Array of output coverage grid/pixel values

b) CRS of output coverage

c) Extent of output coverage

G.2.9   Use case 8: Resampling of unrectified and un-georeferenced image

NOTE   This use case also applies when the georeferencing coordinate transformation(s) are not available to the server or cannot be used by the server.

The GetCoverage request asks a server for a resampled subsection of an un-georeferenced (and unrectified) image, defining the desired extent in the grid CRS of the offered image. See Addition A below.

The needed response from the GetCoverage operation includes all:

a) Array of output coverage grid/pixel values

b) GridCRS of output image, with the GridBaseCRS being the CRS of the offered image

c) Extent of output image

G.2.10   Use case 9: Resampling of unrectified but georeferenced image, defining extent in grid CRS

The GetCoverage request asks a server for a resampled (possibly georectified) subsection of an unrectified but georeferenced image, defining the desired extent in the grid CRS of the offered coverage. See Addition A below.

The needed response from the GetCoverage operation includes all:

a) Array of output coverage grid/pixel values

b) CRS of output image

c) Extent of output image

G.2.11   Use case 10: Resampling of unrectified but georeferenced image, defining extent in ground CRS

The GetCoverage request asks a server for a resampled (possibly georectified) subsection of an unrectified but georeferenced image, defining the desired extent in a 2D or 3D Ground

CRS (GeographicCRS, ProjectedCRS, or either compounded with a VerticalCRS). See Addition A below.

The needed response from the GetCoverage operation includes all:

a) Array of output coverage grid/pixel values

b) CRS of output image

c) Extent of output image

G.2.12 Addition A

The following addition applies to uses cases 6 through 10 listed above

The "differently georectified" or "rectified" or "resampled" output grid requested must be able to be resampled into a client-defined grid that is rotated and skewed, with client-defined grid spacings and origin position, with respect to the ground base CRS for that grid. This client-defined grid in a 3D ground base CRS could be specified by a rectified grid as specified in Figure 14 and CV_RectifiedGrid in Clause 8 of ISO 19123. Figure 14 in ISO 19123 represents a quadrilateral (a parallelogram in general, not necessarily a square or rectangle) grid in a plane within the 3D XYZ space, where the axes of the grid can be at any angles in the 3D XYZ space. A 2D version of that Figure 14 is shown in Figure G.1.
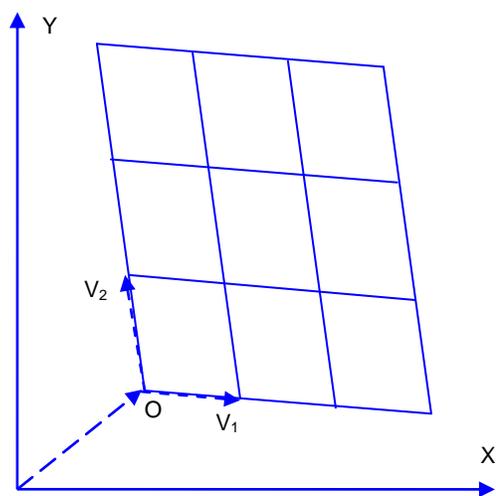
Figure G.1 — Parallelogram grid within a 2D X-Y space

NOTE   This client-defined grid in a 2D ground base CRS could be specified by EPSG coordinate operation method code 9623, which is specified in Subclause 2.4.2.2 and Figure 8 in EPSG Guidance Note 7.

Notice that the specified extent will often be skewed with curved edges in the desired resampled output coverage. A specified skewed and curved extent in the desired resampled output coverage should be interpreted as the minimum extent desired, and the server should extend this extent to the minimum rectangle surrounding that desired skewed/curved extent.

The squared-off or original rectangle can be transformed into the corresponding extent in the georectified offered coverage. Notice that the transformed extent will often be skewed with curved edges in the georectified offered coverage. When the transformed skewed/curved

extent does not lie wholly within the georectified offered coverage, padding with null values is required at all output grid points falling outside the georectified offered coverage.

G.3   Handling of use cases

G.3.1   Overview

This subclause describes how the 10 use cases described above can be handled by the GetCoverage operation design specified in Clause 10. For each use case, Table G.2 lists the type of CRS that can be referenced in the BoundingBox in the DomainSubset part of a GetCoverage operation request.

Table G.2 — GetCoverage request DomainSubset BoundingBox CRS for use cases

Is resampling needed? Offered coverage type Extent CRS type used	Use case number	CRS referenced in BoundingBox in DomainSubset part of GetCoverage request
No resampling		
Georectified coverage		
Extent in ground CRS	1	any ground CRS [c]
Extent in grid CRS	2	GridCRS of georectified offered coverage
Unrectified and un-georeferenced coverage		
Extent in grid CRS [a, b]	3	ImageCRS or GridCRS of unrectified offered coverage
Unrectified and georeferenced coverage		
Extent in grid CRS [b]	4	ImageCRS or GridCRS of unrectified offered coverage
Extent in ground CRS [b]	5	any ground CRS [c]
Resampling		
Georectified coverage		
Extent in ground CRS	6	any ground CRS [c]
Extent in grid CRS	7	GridCRS of georectified offered coverage
Unrectified and un-georeferenced coverage		
Extent in grid CRS [a, b]	8	ImageCRS or GridCRS of unrectified offered coverage
Unrectified and georeferenced coverage		
Extent in grid CRS [b]	9	ImageCRS or GridCRS of unrectified offered coverage
Extent in ground CRS [b]	10	any ground CRS [c]

a   The extent cannot be defined in a ground CRS for an unrectified and un-georeferenced coverage. These two use cases also apply when the georeferencing coordinate transformation(s) are not available to the server or cannot be used by the server,.

b   These use cases apply whether the offered unrectified coverage has an ImageCRS or a GridCRS with a GridBaseCRS that is an ImageCRS.

c   This DomainSubset BoundingBoxextent can be skewed with curved edges in the offered coverage.

For each use case, Table G.3 lists the type of CRS that can be included in the Output part of a GetCoverage request.

Table G.3 — GetCoverage request Output CRS for use cases

Is resampling needed? Offered coverage type Extent CRS type used	Use case number	CRS in Output part of GetCoverage request
No resampling		
Georectified coverage		
Extent in ground CRS	1	no CRS [c]
Extent in grid CRS	2	no CRS [c]
Unrectified and un-georeferenced coverage		
Extent in grid CRS [a]	3	no CRS
Unrectified and georeferenced coverage		
Extent in grid CRS	4	no CRS
Extent in ground CRS	5	no CRS
Resampling		
Georectified coverage		
Extent in ground CRS	6	any wcs:GridCRS that does not specify the same grid points as the wcs:GridCRS of georectified offered coverage [d]
Extent in grid CRS	7	any wcs:GridCRS that does not specify the same grid points as the wcs:GridCRS of georectified offered coverage [d]
Unrectified and un-georeferenced coverage		
Extent in grid CRS [a, b]	8	any wcs:GridCRS that is based on ImageCRS of unrectified offered coverage [d]
Unrectified and georeferenced coverage		
Extent in grid CRS [b]	9	any wcs:GridCRS
Extent in ground CRS [b]	10	any wcs:GridCRS

a   The extent cannot be defined in a ground CRS for an unrectified and un-georeferenced coverage. These two use cases also apply when the georeferencing coordinate transformation(s) are not available to the server or cannot be used by the server.

b   These use cases apply whether the offered unrectified coverage has an ImageCRS or a GridCRS with a GridBaseCRS that is an ImageCRS.

c   This use case will also be obtained when the GridCRS specified in the Output part of the GetCoverage request is the same as the GridCRS of the georectified offered coverage.

d   If only the origin in this output wcs:GridCRS is different from the GridCRS of the georectified offered coverage, but that origin is at a different grid point in the GridCRS of the georectified offered coverage, then this output GridCRS specifies the same grid positions and no resampling is needed.

For each use case, Table G.4 lists the type of CRS that can be referenced in the BoundingBox in the GetCoverage response.

Table G.4 — GetCoverage response BoundingBox CRS for use cases

Is resampling needed? Offered coverage type Extent CRS type used	Use case number	CRS referenced in BoundingBox in GetCoverage response [c]
No resampling		
Georectified coverage		
Extent in ground CRS	1	wcs:GridCRS for output coverage that references ground GridBaseCRS of georectified offered coverage
Extent in grid CRS	2	wcs:GridCRS for output coverage that references ground GridBaseCRS of georectified offered coverage
Unrectified and un-georeferenced coverage		
Extent in grid CRS [a]	3	ImageCRS or GridCRS for output image
Unrectified and georeferenced coverage		
Extent in grid CRS	4	ImageCRS or GridCRS for output image [d]
Extent in ground CRS	5	ImageCRS or GridCRS for output image [c d]
Resampling		
Georectified coverage		
Extent in ground CRS	6	wcs:GridCRS for output coverage based on same ground GridBaseCRS and defining same grid positions as requested output wcs:GridCRS
Extent in grid CRS	7	wcs:GridCRS for output coverage based on same ground GridBaseCRS and defining same grid positions as requested output wcs:GridCRS
Unrectified and un-georeferenced coverage		
Extent in grid CRS [a, b]	8	wcs:GridCRS for output coverage based on ImageCRS or GridCRS of offered unrectified coverage and defining same grid positions as requested output wcs:GridCRS
Unrectified and georeferenced coverage		
Extent in grid CRS [b]	9	wcs:GridCRS for output coverage based on same ground GridBaseCRS and defining same grid positions as requested output wcs:GridCRS [d]
Extent in ground CRS [b]	10	wcs:GridCRS for output coverage based on same ground GridBaseCRS and defining same grid positions as requested output wcs:GridCRS [d]

a   The extent cannot be defined in a ground CRS for an unrectified and un-georeferenced coverage. These two use cases also apply when the georeferencing coordinate transformation(s) are not available to the server or cannot be used by the server. .

b   These use cases apply whether the offered unrectified coverage has an ImageCRS or a GridCRS with a GridBaseCRS that is an ImageCRS.

c   Included unless all this information is included in the coverage format used.

d   For this use case, the GetCoverage response shall also include georeferencing coordinate Transformation(s) for the ImageCRS of the output images.

The following subclauses further describe how each use case can be handled by WCS 1.1. Many of these subclauses use the value "urn:ogc:def:crs:WCS:1.1:GridCRSTBD" to identify a Grid CRS in one or more examples. This subject is discussed in Subclause G.3.13.

G.3.2   Use case 1: No resampling of georectified coverage, defining extent in ground CRS

NOTE 1   This use case applies only when the offered georectified coverage has a GridCRS with a GridBaseCRS that is not an ImageCRS.

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing a ground CRS (not necessarily the GridBaseCRS of the georectified offered coverage). Any ground CRS can be used that the server can transform into the GridBaseCRS of the offered coverage and output coverage. Notice that this DomainSubset BoundingBox extent can be rotated and skewed, without curved edges, in the georectified offered coverage.

NOTE 2   The two specified corners of this BoundingBox are not required to fall exactly on offered coverage pixels in order to avoid resampling, since the Output part of the GetCoverage request specifies (or implies) the desired wcs:GridCRS of the output coverage.

The corners of this BoundingBox are first transformed into positions in the georectified offered coverage, which in this use case is the same as the wcs:GridCRS of the desired output coverage. When that extent is rotated and/or skewed (without curved edges) in the georectified offered coverage, a server interprets this extent as the minimum extent desired, and next extends that extent to the minimum rectangle surrounding that desired rotated/skewed extent. After that transformation and possible extension, the corners of this modified BoundingBox are extended to the next grid position in the georectified offered coverage. The actual output coverage uses the cell values from the source coverage (when they exist).

EXAMPLE 1   DomainSubset XML fragment (2D):

```
<DomainSubset>
  <ows:BoundingBox crs="urn:ogc:def:crs:EPSG:6.6:63266405" dimension="2">
    <ows:LowerCorner>30.25 110.15</ows:LowerCorner>
    <ows:UpperCorner>30.75 110.65</ows:UpperCorner>
  </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2   BoundingBox KVP fragment (2D):

```
BoundingBox=30.25,110.15,30.75,110.65,urn:ogc:def:crs:EPSG:6.6:63266405&
```

For this use case, the Output part of the GetCoverage request can specify either no CRS or the wcs:GridCRS of the georectified offered coverage (the same CRS as referenced in the DomainSubset BoundingBox). When no output grid is specified, this is interpreted by servers as requesting the grid of georectified offered coverage.

NOTE 3   The origin of this wcs:GridCRS is NOT required or expected to be the actual position of the origin in the GetCoverage operation output, since that origin position is indirectly specified by the OWS BoundingBox in the DomainSubset.

EXAMPLE 3   XML fragment (2D), in which GridCRS should be omitted:

```
<Output...>
    <GridCRS> <!-- See Addition B below for examples -->
    </GridCRS>
</Output>
```

EXAMPLE 4   KVP fragment (2D), which should be omitted:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

a) Array of output coverage grid/pixel values

b) ows:BoundingBox, referencing the wcs:GridCRS for the output coverage that references the ground CRS of the georectified offered coverage (required unless all this information is included in the coverage format used)

c) Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

EXAMPLE 5   XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="gridCRSofOutputCoverage" dimension="2">
    <ows:LowerCorner>0 0</ows:LowerCorner>
    <ows:UpperCorner>7 7</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS> <!-- See Addition B below for examples -->
</wcs:GridCRS>
```

NOTE 3   In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123. The modified wcs:GridCRS origin is at (589, 172).

G.3.3   Use case 2: No resampling of georectified image, defining extent in grid CRS

NOTE 1   This use case applies only when the offered georectified coverage has a GridCRS with a GridBaseCRS that is not an ImageCRS.

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing the GridCRS of the georectified offered coverage. Notice that this extent cannot be skewed or with curved edges in the georectified offered coverage.

EXAMPLE 1   DomainSubset XML fragment (2D):

```
<DomainSubset>
    <ows:BoundingBox crs="urn:ogc:def:crs:WCS:1.1:GridCRSTBD" dimension="2">
        <ows:LowerCorner>1000 2000</ows:LowerCorner>
        <ows:UpperCorner>1511 2511</ows:UpperCorner>
    </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2   BoundingBox KVP fragment (2D):

```
BoundingBox=1000,2000,1511,2511,urn:ogc:def:crs:WCS:1.1: GridCRSTBD
```

For this use case, the Output part of the GetCoverage request can specify either no CRS or the wcs:GridCRS of the georectified offered coverage (not the same CRS as referenced in the DomainSubset BoundingBox). When no output grid GRS is specified, this is interpreted by servers as requesting the grid of the georectified offered coverage.

EXAMPLE 3   XML fragment (2D) , in which GridCRS should be omitted:

```
<Output...>
    <GridCRS> <!-- See Addition B below for examples -->
    </GridCRS>
</Output>
```

EXAMPLE 4   KVP fragment (2D), which should be omitted:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

a)   Array of output coverage grid/pixel values

b)   ows:BoundingBox, referencing the wcs:GridCRS for the output coverage that references the ground CRS of the georectified offered coverage (required unless all this information is included in the coverage format used)

c)   Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

EXAMPLE 5   XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="urn:ogc:def:crs:WCS:1.1:GridCRSTBD" dimension="2">
  <ows:LowerCorner>0 0</ows:LowerCorner>
  <ows:UpperCorner>511 511</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS> <!-- See Addition B below for examples -->
</wcs:GridCRS>
```

NOTE 2   In this example, the LowerCorner is the (0, 0) origin of this output grid.

G.3.4   Use case 3: No resampling of unrectified and un-georeferenced image

NOTE 1   This use case applies whether the offered unrectified coverage has an ImageCRS or a GridCRS with a GridBaseCRS that is an ImageCRS

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing the ImageCRS or GridCRS of the unrectified offered coverage. Notice that this extent cannot be skewed or with curved edges in the unrectified offered coverage.

EXAMPLE 1   DomainSubset XML fragment (2D):

```
<DomainSubset>
    <ows:BoundingBox crs="urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:ImageCrsID"
dimension="2">
        <ows:LowerCorner>1000 2000</ows:LowerCorner>
        <ows:UpperCorner>1511 2511</ows:UpperCorner>
    </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2   BoundingBox KVP fragment (2D):

```
BoundingBox=1000,2000,1511,2511,urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:Imag
eCrsID&
```

For this use case, the Output part of the GetCoverage request does not specify any wcs:GridCRS. The CRS used by the actual output coverage will be the ImageCRS or GridCRS of the offered unrectified image usually with a different origin point or pixel.

For this use case, the GetCoverage response contains:

a) Array of output coverage grid/pixel values

b) ows:BoundingBox (XML encoded), referencing the GridCRS for the output image (required unless all this information is included in the coverage format used)

c) Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

EXAMPLE 5   XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:ImageCrsID"
dimension="2">
   <ows:LowerCorner>0 0</ows:LowerCorner>
   <ows:UpperCorner>511 511</ows:UpperCorner>
</ows:BoundingBox>
```

NOTE 2   In this example, the LowerCorner is the (0, 0) origin of this output grid. The wcs:GridCRS origin is at (1000, 2000).

G.3.5   Use case 4: No resampling of unrectified but georeferenced image, defining extent in grid CRS

NOTE 1   This use case applies whether the offered unrectified coverage has an ImageCRS or a GridCRS with a GridBaseCRS that is an ImageCRS

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing the ImageCRS or GridCRS of the unrectified offered coverage. Notice that this extent cannot be skewed or with curved edges in the unrectified offered image.

EXAMPLE 1   DomainSubset XML fragment (2D):

```
<DomainSubset>
   <ows:BoundingBox crs="urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:ImageCrsID"
dimension="2">
      <ows:LowerCorner>1000 2000</ows:LowerCorner>
      <ows:UpperCorner>1511 2511</ows:UpperCorner>
   </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2   BoundingBox KVP fragment (2D):

```
BoundingBox=1000,2000,1511,2511,urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:Imag
eCrsID&
```

For this use case, the Output part of the GetCoverage request does not specify any wcs:GridCRS. The CRS used by the actual output coverage will be the ImageCRS of the offered image, usually with different origin point or pixel.

For this use case, the GetCoverage response contains:

a) Array of output coverage grid/pixel values

b) ows:BoundingBox (XML encoded), referencing the ImageCRS or GridCRS of output image (required unless all this information is included in the coverage format used)

c) Mandatory georeferencing coordinate Transformation(s), referencing the ImageCRS or GridCRS of the output image (this Transformation(s) might be included in the coverage format used)

EXAMPLE 3   XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:ImageCrsID"
dimension="2">
   <ows:LowerCorner>0 0</ows:LowerCorner>
   <ows:UpperCorner>511 511</ows:UpperCorner>
</ows:BoundingBox>
```

NOTE 2   In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123.

G.3.6 Use case 5: No resampling of unrectified but georeferenced image, defining extent in ground CRS

NOTE 1   This use case applies whether the offered unrectified coverage has an ImageCRS or a GridCRS with a GridBaseCRS that is an ImageCRS

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing a ground CRS (not necessarily the GridBaseCRS of the georeferencing Transformations). Any ground CRS can be used that the server can transform into the GridBaseCRS of both the offered coverage and the desired re-rectified output coverage. Notice that this extent can be skewed with curved edges in the offered coverage.

NOTE 2   The two specified corners of this BoundingBox are not required to fall exactly on offered coverage pixels in order to avoid resampling, since the Output part of the GetCoverage request implies the ImageCRS of the offered image.

EXAMPLE 1   DomainSubset XML fragment (2D):

```
<DomainSubset>
   <ows:BoundingBox crs="urn:ogc:def:crs:EPSG:6.6:63266405" dimension="2">
      <ows:LowerCorner>30.25 110.15</ows:LowerCorner>
      <ows:UpperCorner>30.75 110.65</ows:UpperCorner>
   </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2   BoundingBox KVP fragment (2D):

```
BoundingBox=30.25,110.15,30.75,110.65,urn:ogc:def:crs:EPSG:6.6:63266405&
```

For this use case, the Output part of the GetCoverage request does not specify any wcs:GridCRS. The CRS used by the actual output coverage will be the ImageCRS or GridCRS of the offered unrectified image, usually with different origin point or pixel.

For this use case, the GetCoverage response contains:

a) Array of output coverage grid/pixel values

b) ows:BoundingBox (XML encoded), referencing the GridCRS of output image (required unless all this information is included in the coverage format used)

c) Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

d) Mandatory georeferencing coordinate Transformation(s), referencing the ImageCRS of the output image (this Transformation(s) might be included in the coverage format used)

EXAMPLE 3   XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="urn:ogc:def:crs:WCS:1.1:GridCRSTBD" dimension="2">
   <ows:LowerCorner>0 0</ows:LowerCorner>
   <ows:UpperCorner>7 7</ows:UpperCorner>
</ows:BoundingBox>
```

NOTE 3   In this example, the LowerCorner is the (0, 0) origin of this output grid. The wcs:GridCRS origin is at (589, 172).

G.3.7   Use case 6: Resampling of georectified coverage, defining extent in ground CRS

NOTE 1   This use case applies only when the offered georectified coverage has a GridCRS with a GridBaseCRS that is not an ImageCRS.

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing a ground CRS (not necessarily the base of desired re-rectified output coverage). Any ground CRS can be used that the server can transform into the GridBaseCRS of both the offered coverage and the desired re-rectified output coverage. Notice that this extent can be skewed with curved edges in the georectified offered coverage and in the desired output image.

EXAMPLE 1   DomainSubset XML fragment (2D):

```
<DomainSubset>
   <ows:BoundingBox crs="urn:ogc:def:crs:EPSG:6.6:63266405" dimension="2">
      <ows:LowerCorner>30.25 110.15</ows:LowerCorner>
      <ows:UpperCorner>30.75 110.65</ows:UpperCorner>
   </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2   BoundingBox KVP fragment (2D):

```
BoundingBox=30.25,110.15,30.75,110.65,urn:ogc:def:crs:EPSG:6.6:63266405&
```

For this use case, the Output part of the GetCoverage request can specify any wcs:GridCRS. This wcs:GridCRS is the desired CRS of the output coverage. The GridBaseCRS of this wcs:GridCRS (roughly) corresponds to the WCS 1.0 ResponseCRS. This wcs:GridCRS contains additional information, including the information corresponding to the WCS 1.0 RESX, RESY, and RESZ. The GridBaseCRS of this wcs:GridCRS can be any ground CRS, either the same as or different from the georectified offered coverage.

NOTE 2   If only the origin in this output wcs:GridCRS is different from the GridCRS of the georectified offered coverage, but that origin is at a different grid point in the GridCRS of the georectified offered coverage, then this output GridCRS specifies the same grid positions and no resampling is needed, see use case 1

EXAMPLE 3   XML fragment (2D):

```
<Output...>
 <GridCRS> <!-- See Addition B below for examples -->
    </GridCRS>
</Output>
```

EXAMPLE 4   KVP fragment (2D):

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

a) Array of output coverage grid/pixel values

b) ows:BoundingBox (XML encoded), referencing the wcs:GridCRS for the output coverage that is based on the same ground CRS and defines same grid positions as requested output wcs:GridCRS (required unless all this information is included in the coverage format used)

c) Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

EXAMPLE 5   XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="urn:ogc:def:crs:WCS:1.1:GridCRSTBD" dimension="2">
   <ows:LowerCorner>0 0</ows:LowerCorner>
   <ows:UpperCorner>987 987</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS gml:id="TBD"> See Addition B below
```
NOTE 2   In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123.

G.3.8   Use case 7: Resampling of georectified image, defining extent in grid CRS

NOTE 1   This use case applies only when the offered georectified image has a GridCRS with a GridBaseCRS that is not an ImageCRS.

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing the GridCRS of the georectified offered coverage. Notice that this extent cannot be skewed or with curved edges in the unrectified offered coverage. However, this extent can be skewed with curved edges in the desired output image.

EXAMPLE 1   DomainSubset XML fragment (2D):

```
<DomainSubset>
   <ows:BoundingBox crs="urn:ogc:def:crs:WCS:1.1:GridCRSTBD" dimension="2">
      <ows:LowerCorner>1000 2000</ows:LowerCorner>
      <ows:UpperCorner>1511 2511</ows:UpperCorner>
   </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2   BoundingBox KVP fragment (2D):

```
BoundingBox=1000,2000,1511,2511,urn:ogc:def:crs:WCS:1.1:GridCRSTBD&
```

For this use case, the Output part of the GetCoverage request can specify any wcs:GridCRS. This wcs:GridCRS is the desired CRS of the output coverage. The GtidBaseCRS of this wcs:GridCRS (roughly) corresponds to the WCS 1.0 ResponseCRS. This wcs:GridCRS contains additional information, including the information corresponding to the WCS 1.0 RESX, RESY, and RESZ. The GridBaseCRS of this wcs:GridCRS can be any ground CRS, either the same as or different from the georectified offered coverage.

NOTE 3   If only the origin in this output wcs:GridCRS is different from the GridCRS of the georectified offered coverage, but that origin is at a different grid point in the GridCRS of the georectified offered coverage, then this output GridCRS specifies the same grid positions and no resampling is needed, see use case 2

EXAMPLE 3   XML fragment (2D):

```
<Output...>
   <GridCRS> <!-- See Addition B below for examples -->
   </GridCRS>
</Output>
```

EXAMPLE 4   KVP fragment (2D):

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

a)   Array of output coverage grid/pixel values

b)   ows:BoundingBox (XML encoded), referencing the wcs:GridCRS for the output coverage that is based on the same ground CRS and defines same grid positions as requested output wcs:GridCRS

c)   Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

EXAMPLE 5   XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="urn:ogc:def:crs:WCS:1.1:GridCRSTBD" dimension="2">
   <ows:LowerCorner>0 0</ows:LowerCorner>
   <ows:UpperCorner>765 765</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS> <!-- See Addition B below for examples -->
</wcs:GridCRS>
```

NOTE 4   In this example, the LowerCorner is the (0, 0) origin of this output grid.

G.3.9   Use case 8: Resampling of unrectified and un-georeferenced image

NOTE 1   This use case applies whether the offered unrectified coverage has an ImageCRS or a GridCRS with a GridBaseCRS that is an ImageCRS.

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing the ImageCRS or GridCRS of the unrectified offered coverage. Notice that this extent cannot be skewed or with curved edges in the unrectified offered coverage. However, this extent can be skewed with straight edges in the desired output image.

EXAMPLE 1   DomainSubset XML fragment (2D):

```
<DomainSubset>
   <ows:BoundingBox crs="urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:ImageCrsID"
dimension="2">
      <ows:LowerCorner>1000 2000</ows:LowerCorner>
      <ows:UpperCorner>1511 2511</ows:UpperCorner>
   </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2   BoundingBox KVP fragment (2D):

```
BoundingBox=1000,2000,1511,2511,urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:Imag
eCrsID
```

For this use case, the Output part of the GetCoverage request can specify any wcs:GridCRS. This wcs:GridCRS is the desired CRS of the output coverage. The GridBaseCRS of this wcs:GridCRS (roughly) corresponds to the WCS 1.0.0 ResponseCRS. This wcs:GridCRS contains additional information, including the information corresponding to the WCS 1.0.0 RESX, RESY, and RESZ.

NOTE 2   No resampling is needed when this wcs:GridCRS (that is based on the ImageCRS of the georectified offered coverage) has parameters that define grid positions which exactly match the grid positions in that ImageCRS, but this should be rare.

EXAMPLE 3   XML fragment (2D):

```
<Output...>
   <GridCRS> <!-- See Addition B below for examples -->
   </GridCRS>
</Output>
```

EXAMPLE 4   KVP fragment (2D):

```
GridBaseCRS=urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:ImageCrsID&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

a)   Array of output coverage grid/pixel values

b)   ows:BoundingBox (XML encoded), referencing the wcs:GridCRS for the output coverage based on the ImageCRS or GridCRS of the unrectified offered coverage and defines same grid positions as the requested output wcs:GridCRS (required unless all this information is included in the coverage format used)

c)   Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

EXAMPLE 5   XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="urn:ogc:def:crs:WCS:1.1:GridCRSTBD" dimension="2">
   <ows:LowerCorner>0 0</ows:LowerCorner>
   <ows:UpperCorner>765 765</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS> <!-- See Addition B below for examples -->
</wcs:GridCRS>
```

NOTE 3   In this example, the LowerCorner is the (0, 0) origin of this output grid.

G.3.10   Use case 9: Resampling of unrectified but georeferenced image, defining extent in grid CRS

NOTE 1   This use case applies whether the offered unrectified coverage has an ImageCRS or a GridCRS with a GridBaseCRS that is an ImageCRS.

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing the ImageCRS or GridCRS of the unrectified offered coverage. Notice that this extent cannot be skewed or with curved edges in the unrectified offered coverage. However, this extent can be skewed with curved edges in the desired output coverage.

EXAMPLE 1   DomainSubset XML fragment (2D):

```
<DomainSubset>
   <ows:BoundingBox crs="urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:ImageCrsID"
dimension="2">
      <ows:LowerCorner>1000 2000</ows:LowerCorner>
      <ows:UpperCorner>1511 2511</ows:UpperCorner>
   </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2   BoundingBox KVP fragment (2D):

```
BoundingBox=1000,2000,1511,2511,urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:Imag
eCrsID&
```

For this use case, the Output part of the GetCoverage request can specify any wcs:GridCRS. This wcs:GridCRS is the desired CRS of the output coverage. The GridBaseCRS of this wcs:GridCRS (roughly) corresponds to the WCS 1.0 ResponseCRS. This wcs:GridCRS contains additional information, including the information corresponding to the WCS 1.0 RESX, RESY, and RESZ. The GridBaseCRS of this wcs:GridCRS can be any ground CRS, either the same as or different from the unrectified offered coverage.

EXAMPLE 3   XML fragment (2D):

```
<Output...>
   <GridCRS> <!-- See Addition B below for examples -->
   </GridCRS>
</Output>
```

EXAMPLE 4   KVP fragment (2D):

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

a) Array of output coverage grid/pixel values

b) ows:BoundingBox (XML encoded), referencing the wcs:GridCRS for the output coverage that is based on the same ground CRS and defines same grid positions as requested output wcs:GridCRS (required unless all this information is included in the coverage format used)

c) Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

d) Mandatory georeferencing coordinate Transformation(s), referencing the GridCRS of the output image (this Transformation(s) might be included in the coverage format used)

EXAMPLE 5   XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="urn:ogc:def:crs:WCS:1.1:GridCRSTBD" dimension="2">
   <ows:LowerCorner>0 0</ows:LowerCorner>
   <ows:UpperCorner>765 765</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS> <!-- See Addition B below for examples -->
</wcs:GridCRS>
```

NOTE 2   In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123.

G.3.11 Use case 10: Resampling of unrectified but georeferenced image, defining extent in ground CRS

NOTE 1   This use case applies whether the offered unrectified coverage has an ImageCRS or a GridCRS with a GridBaseCRS that is an ImageCRS.

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing a ground CRS (not necessarily the base CRS of desired rectified output coverage). Any ground CRS can be used that the server

can transform into the GridBaseCRS of both the offered coverage and the desired resampled output coverage. Notice that this extent can be skewed with curved edges in the unrectified offered coverage and in the desired output coverage.

EXAMPLE 1   DomainSubset XML fragment (2D):

```
<DomainSubset>
   <ows:BoundingBox crs="urn:ogc:def:crs:EPSG:6.6:63266405" dimension="2">
      <ows:LowerCorner>30.25 110.15</ows:LowerCorner>
      <ows:UpperCorner>30.75 110.65</ows:UpperCorner>
   </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2   BoundingBox KVP fragment (2D):

```
BoundingBox=30.25,110.15,30.75,110.65,urn:ogc:def:crs:EPSG:6.6:63266405
```

For this use case, the Output part of the GetCoverage request can specify any wcs:GridCRS. This wcs:GridCRS is the desired CRS of the output coverage. The GridBaseCRS of this wcs:GridCRS (roughly) corresponds to the WCS 1.0 ResponseCRS. This wcs:GridCRS contains additional information, including the information corresponding to the WCS 1.0 RESX, RESY, and RESZ. The GridBaseCRS of this wcs:GridCRS can be any ground CRS.

EXAMPLE 3 XML fragment (2D):

```
<Output...>
    <GridCRS> <!-- See Addition B below for examples -->
    </GridCRS>
</Output>
```

EXAMPLE 4   KVP fragment (2D):

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,10.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

a) Array of output coverage grid/pixel values

b) ows:BoundingBox (XML encoded), referencing the wcs:GridCRS for the output coverage that is based on the same ground CRS and defines the same grid positions as requested output wcs:GridCRS (required unless all this information is included in the coverage format used)

c) Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

d) Mandatory georeferencing coordinate Transformation(s), referencing the GridCRS of the output image (this Transformation(s) might be included in the coverage format used)

EXAMPLE 5   XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="urn:ogc:def:crs:WCS:1.1:GridCRSTBD" dimension="2">
   <ows:LowerCorner>0 0</ows:LowerCorner>
   <ows:UpperCorner>987 987</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS> <!-- See Addition B below for examples -->
</wcs:GridCRS>
```

NOTE 2   In this example, the LowerCorner is the (0, 0) origin of this output grid.

111

G.3.12 Addition B

EXAMPLE 1 An XML document fragment for a 2D wcs:GridCRS without rotation or skewing, using the 2dGridIn2dCrs method is:

```
<GridCRS gml:id="example2dGridCrs1">
   <gml:srsName>2D Grid CRS Example 1</gml:srsName>
   <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:63266405</GridBaseCRS>
   <GridType>urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs</GridType>
   <GridOrigin>1.0 2.0</GridOrigin>
   <GridOffsets>0.1 0 0 0.2</GridOffsets>
   <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS>
</GridCRS>
```

EXAMPLE 2 An XML document fragment for a 2D wcs:GridCRS with rotation but no skewing using the 2dGridIn2dCrs is:

```
<GridCRS gml:id="example2dGridCrs2">
   <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:32618</GridBaseCRS>
   <GridType>urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs</GridType>
   <GridOrigin>1.0 2.0</GridOrigin>
   <GridOffsets>0.0707 -0.0707 0.1414 0.1414</GridOffsets>
   <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS>
</GridCRS>
```

G.3.13 GridCRS identifiers

Many of these subclauses use the value "urn:ogc:def:crs:WCS:1.1:GridCRSTBD" to identify a Grid CRS in one or more of the examples. This example value is used for the "crs" attribute in an ows:BoundingBox. This value is used since no URNs in the OGC URN namespace have been specified to date for GridCRSs.

OWS Common now specifies that a value be included for the "crs" in an ows:BoundingBox, unless specified in an XML element which includes this BoundingBox. However in this WCS standard, when a GridCRS is used by an ows:BoundingBox, that GridCRS is sometimes specified in the same XML element which includes that ows:BoundingBox.

EXAMPLE 1   In Table 13, the first BoundingBox in the SpatialDomain specifies the BoundingBox in the one CRS of the offered coverage. That CRS is either the GridCRS defined within that SpatialDomain, or the ImageCRS referenced within that SpatialDomain. In this case, the "crs" of this first BoundingBox seems clear, even if that GridCRS is not identified.

EXAMPLE 2   In Table 23, the BoundingBox in the DomainSubset may use many different CRSs, including the one CRS of the offered coverage. That offered coverage CRS is either the GridCRS defined in or the ImageCRS referenced in the SpatialDomain of the CoverageDescription. (This BoundingBox may NOT use the GridCRS defined in the Output part of the GetCoverage request.) If the GridCRS of the offered coverage includes an identifier value, that value can be used to reference that CRS in this BoundingBox. Otherwise, the "crs" of this BoundingBox seems unclear, when it is the GridCRS of the offered coverage.

EXAMPLE 3   In Table 30, the BoundingBox in the required output metadata will often use the GridCRS that is also specified in that required output metadata. (That GridCRS may be the GridCRS defined in the Output part of the GetCoverage request.) In this case, the "crs" of this BoundingBox seems clear, even if that GridCRS is not identified.

EDITOR'S NOTE   I think this issue should be clarified in WCS 1.2.

Annex H
(normative)

Other data structures

H.1   Overview

This annex specifies the Coverages data structure that is used by this WCS, based on the Manifest data structure specified in OWS Common 1.1 [OGC 06-121r3]. This annex also specifies the InterpolationMethods data structure.

NOTE   Although discussed here as parts of WCS, the Coverages and InterpolationMethods might be placed in a separate document specifying aspects specific to coverages that are used by multiple Implementation specifications.

H.2   Coverages data structure

H.2.1   Introduction

In some cases, an operation response can consist of a group of coverages where each coverage uses essentially the same contents as the ReferenceGroup specified in Subclause 13.3 of OWS Common [OGC 06-121r3]. Similarly, part of an operation request can contain a group of coverages using essentially the same contents. That is, this "Coverages" data structure is similar to a Manifest data straucture which can contain multiple coverages, where each coverage can be a group of related data items. In this use, all of these data items might be contained in one package that is transferred as the operation response or in the operation request mesage, or some of these data items might be stored elsewhere for subsequent retrieval.

EXAMPLE 1   Some responses to a WCS GetCoverage operation can contain multiple groups of related data items, where each group contains an output coverage with its metadata.

EXAMPLE 2   The request for a WCS Transaction operation will contain multiple groups of related data items, where each group is an input coverage with its metadata.

This subclause thus defines a Coverages class, and corresponding element, which may be used by specific OWSs. This Coverages class uses the ReferenceGroup class renamed Coverage. These classes are included in the Coverages package UML class diagram in Figure H.1.

For Coverages used as an operation input, each Coverage may include data items that are accessible from another OGC or other web service using an operation. If this operation request is KVP encoded, that complete operation request shall be encoded as a URL value, in the xlink:href attribute in the AbstractReferenceBaseType. However, if this operation request is XML encoded, a XML encoded operation request must be sent to the other web server whose operation URL is referenced by the xlink:href attribute. Each Coverage may thus use the ServiceReference (class and element) as specified in Subclause 13.5 of [OGC 06-121r3]. This class is a subclass of the Reference class, which

113

adds a UML attribute that either contains or references the XML encoded operation request needed to retrieve this input data from the other web server.
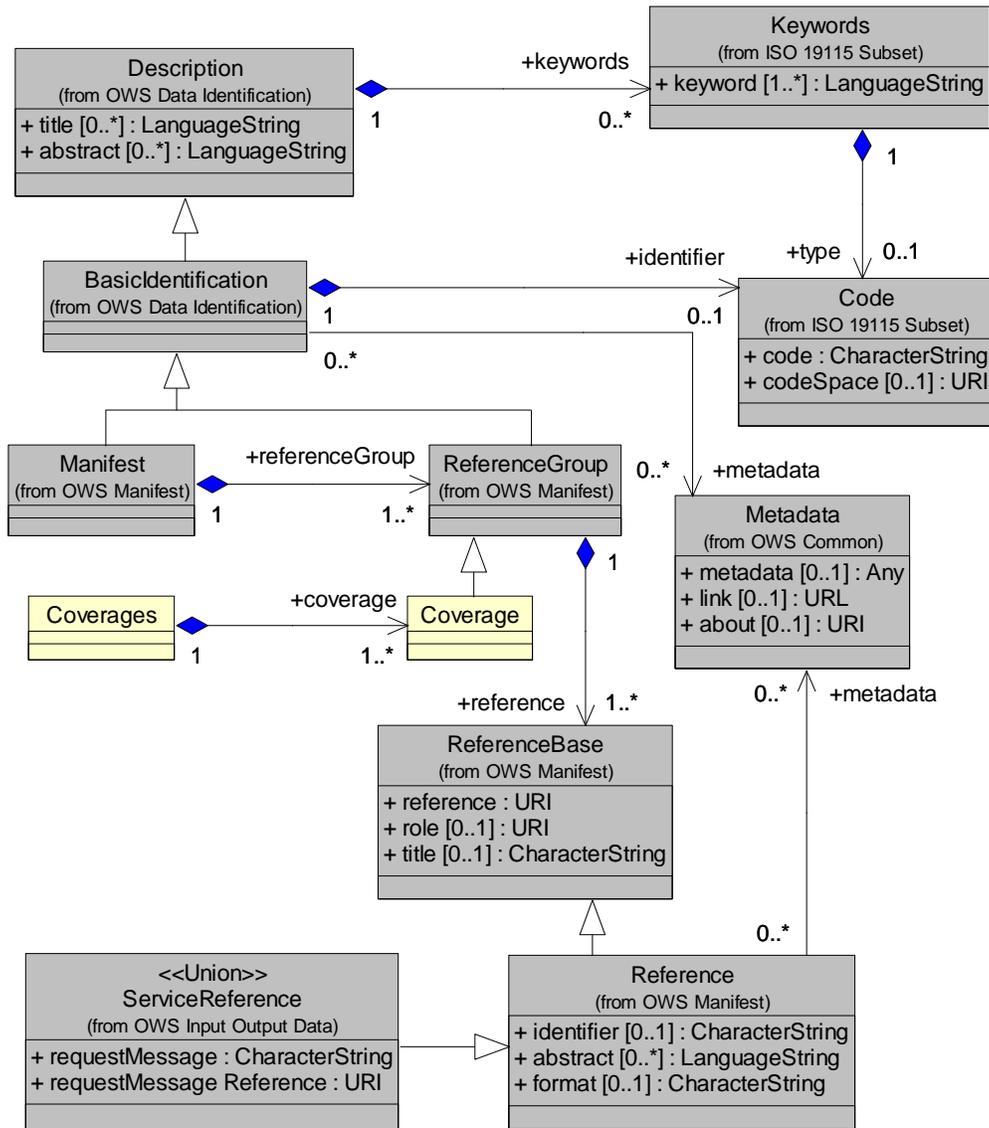

Figure H.1 — Coverages package UML diagram

H.2.2   XML encoding

A Coverages data structure shall be encoded as specified in the attached wcsCoverages.xsd file.

EXAMPLE   Here is an example XML document using the Coverages element:

```
<?xml version="1.0" encoding="UTF-8"?>
<Coverages xmlns="http://www.opengis.net/wcs/1.1.1"
    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/1.1.1 ../owsCoverages.xsd
http://www.opengis.net/ows/1.1 ../../../ows/1.1.0/owsAll.xsd">
```

```
    <Coverage>
        <ows:Title>TBD</ows:Title>
        <ows:Abstract>Coverage created from GetCoverage operation request to a
WCS</ows:Abstract>
        <Identifier>TBD</Identifier>
        <Reference xlink:href="http://my.server.com/coverage/image.tiff"
xlink:role="urn:ogc:def:role:WCS:1.1:coverage"/>
        <Reference xlink:href="http://my.server.com/coverage/metadata.xml"
xlink:role="urn:ogc:def:role:WCS:1.1:metadata"/>
    </Coverage>
</Coverages>
```

H.3   Interpolation methods

H.3.1   Interpolation methods contents

The InterpolationMethods data structure lists one or more spatial interpolation methods, which can each be used for interpolation of a range record field in a continuous grid coverage. The InterpolationMethods shall use the data structure described in Figure H.2 and specified in Tables H.1 through Table H.4.
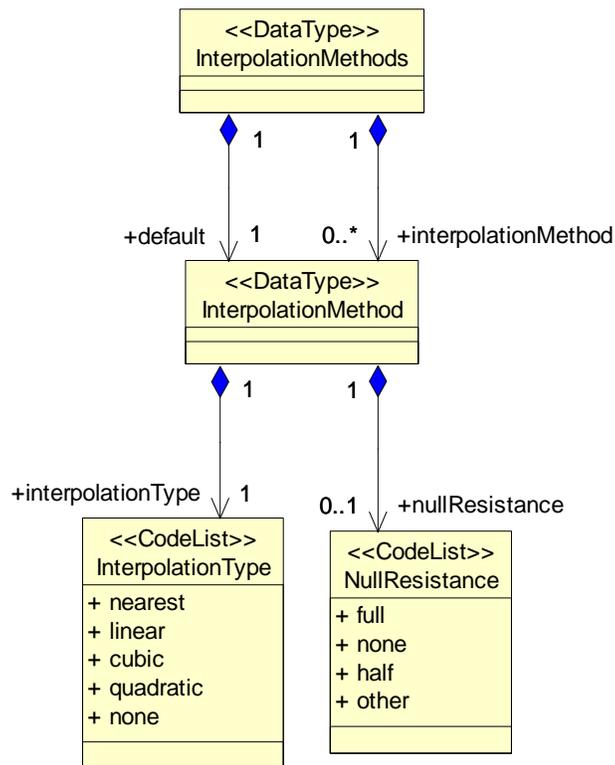
Figure H.2 — Interpolation Methods class diagram

Table H.1 — Parts of the InterpolationMethods data structure

Names [a]	Definition	Data type	Multiplicity and use
default Default	Spatial interpolation method used for this range field when client doesn't specify one [b]	InterpolationMethod data structure, see Table H.2	One (mandatory)
interpolationMethod InterpolationMethod	Another spatial Interpolation method that server can apply to this range field	InterpolationMethod data structure, see Table H.2	Zero or more (optional) Include for each other method supported [c]
a	See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.		
b	This default interpolation method should be the recommended or normal method for this coverage range field.		
c	When the default interpolation method is "none", no other methods should be listed.		

InterpolationMethods has a mandatory **Default** parameter that identifies the spatial interpolation method used for GetCoverage requests that don't specify one. **InterpolationMethods** may also have zero or more **InterpolationMethod** parameters whose values identify other spatial interpolation methods that the server can use for this coverage range field. However, when the default interpolation method is "none", no other methods should be listed. When the only listed (default) interpolation method is "none", clients may only retrieve subsets of this offered coverage in its offered CRS and with its offered resolution and grid positions. That is, the requested output grid points shall lie exactly at some offered grid points. In this case, no interpolation is required to be performed by a WCS server. Alternately, the "nearest" interpolation method type may be applied, to output grid points which are always at the same locations as offered grid points.

Table H.2 — Parts of the InterpolationMethod data structure

Names [a]	Definition	Data type and value	Multiplicity and use
Interpolation MethodType *(anonymous)*	Identifier of spatial interpolation method.	CodeList type One of the strings defined in Table H.3	One (mandatory)
nullResistance nullResistance	How server handles null values when spatially interpolating values in this field	CodeList type One of the strings listed in Table H.4.	Zero or one (optional) Include when known
a	See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.		

For a continuous grid coverage, the normal spatial interpolation types are listed in Table H.3 and in the attached default InterpolationMethods.xml dictionary. Compared to ISO 19123, WCS does not distinguish linear from bilinear, quadratic from biquadratic, and cubic from bicubic interpolation. For WCS, the linear, quadratic, and cubic interpolation methods shall be applicable to coverages of any number of spatial dimensions. Servers may implement and list other interpolation methods, defined in dictionaries other than the default.

NOTE 1   The interpolation methods "lost area" and "barycentric" are not applicable to grid coverages, hence they are not included in this version of WCS.

Table H.3 — Normal spatial interpolation types

Interpolation type identifier	Interpolation method	Description
nearest	nearest neighbour	These are defined in ISO 19123 (Schema for Coverage Geometry and Functions), Annex B.
linear	bilinear [a]	
quadratic	biquadratic [a]	
cubic	bicubic [a]	
none	none	No interpolation is available; requests must be for locations that are among the original domain locations.
a ISO 19123 specifies two-dimensional bilinear, biquadratic, and bicubic interpolation types. When also interpolating in the elevation dimension of a 3D CRS, the 3D extensions of these interpolation types shall be used.		

Each **InterpolationMethod** element has an optional **nullResistance** summarizing how the server treats null-valued points in interpolation. The normal values for this parameter are listed in Table H.4 and in the default InterpolationMethods.xml dictionary.

NOTE 2   These nullResistance values are sent from a server to a client, to summarize how the server treats null-valued points. These nullResistance values are NOT sent from a client to a server (to indicate how the client wants null-valued points treated).

Table H.4 — Effect of null values in spatial interpolation

nullResistance	Description
full	Nulls in the input grid(s) have no effect on interpolated value; resampling uses only non-null valued cells.
none	Interpolated value is Null when any one input value is Null.
half	Interpolated value is Null when half or more input values are Null.
other	Resampling treats nulls differently than any method described above

H.3.2   Interpolation methods XML encoding

A set of interpolation methods can be XML encoded, in each Field in the Range part of each CoverageDescription. The XML encoding of InterpolationMethods shall use the XML Schema fragment in the attached wcsInterpolationMethod.xsd file. The InterpolationType and nullResistance are UML <CodeList> stereotyped classes, which are thus converted into XML Schema following the GML pattern for XML encoding of <CodeList> classes.

That is, in wcsInterpolationMethod.xsd, the normal values of InterpolationType and nullResistance are NOT encoded in the XML Schema, but in the default-referenced GML Dictionary. That Dictionary document is named interpolationMethods.xml and also attached to this document. This approach is used since not all useful values of these quantities are currently known, and this approach allows additional values to be more easily specified and used.

EXAMPLE   Here is an example InterpolationMethods data structure encoded in XML. It lists four interpolation methods: the first three ("cubic," "linear," "quadratic") are defined in the default codespace dictionary; the fourth ("sinusoidal") is defined at the URL given in the codeSpace attribute.

```
<?xml version="1.0" encoding="UTF-8"?>
<InterpolationMethods xmlns="http://www.opengis.net/wcs/1.1"
```

```
xmlns:ows="http://www.opengis.net/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
../wcsInterpolationMethod.xsd">
    <InterpolationMethod>cubic</InterpolationMethod>
    <InterpolationMethod>linear</InterpolationMethod>
    <InterpolationMethod>quadratic</InterpolationMethod>
    <InterpolationMethod
codeSpace="http://my.server.org/OtherInterpolationMethods.xml">sinusoidal</Inter
polationMethod>
    <Default>nearest</Default>
</InterpolationMethods>
```

H.3.3   Interpolation method KVP encoding

One selected interpolation method can be KVP encoded for each range field in a
GetCoverage operation request as specified in Subclause 10.2.2.

Bibliography

[1]   ISO 8601:2000, Data elements and interchange formats — Information interchange — Representation of dates and times

[2]   ISO 19103, Geographic Information – Conceptual schema language

[3]   OGC 05-007r4, Web Processing Service (WPS) Implementation Specification

[4]   OGC 06-094, Web Feature service, version 1.1

[5]   OGC 07-066, Corrigendum 1 for OpenGIS Implementation Specification 06-083r8 (WCS 1.1.0)

[6]   OGC 07-092r2, Definition identifier URNs in OGC namespace

[7]   OMG Unified Modeling Language Specification (UML), Version 1.5, March 2003, http://www.omg.org/docs/formal/03-03-01.pdf