Open Geospatial Consortium Inc.

Date: 2007-10-08

Reference number of this document: OGC 07-055r1

Version: 0.4.0

Category: OpenGIS® Discussion Paper

Editors: Arliss Whiteside, Markus Müller, Stephane Fellah, Frank Warmerdam

Web Coordinate Transformation Service (WCTS) Interface Engineering Report

Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Document type:	OpenGIS® Discussion Paper
Document subtype:	
Document stage:	Revised Discussion Paper
Document language:	English

Contents

Page

Figures

Page

Tables

Page

i. Preface

This Discussion Paper describes an interface specification for a web coordinate transformation service that now builds on version 1.1 of the OWS Common Specification [OGC 06-121r3]. All versions of this document specify an "OGC Web Service" type of interface to a service that performs coordinate transformations. Such transformations include all the types of coordinate operations defined in OGC Abstract Specification Topic 2 [OGC 05-103], including both coordinate "Transformations" and "Conversions".

Suggested additions, changes, and comments on this draft specification are welcome and encouraged. Such suggestions may be submitted to the editor by email message. Extensive and/or multiple changes can be suggested by making changes in an edited copy of this document.

ii. Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 06-121r3]. In particular, the word "shall" (not "must") is the verb form used to indicate a requirement to be strictly followed to conform to this specification

iii. Submitting organizations

The following organizations submitted this proposed Implementation Specification to the Open Geospatial Consortium Inc. as a part of the WCTS Request For Comment (RFC): (TBR)

a) Galdos Systems, Inc.

b) BAE Systems

c) lat/lon

d) DM Solutions Group Inc.

e) CadCorp

f) PCI Geomatics Inc.

g) George Mason University

h) US Census (?)

iv. Document contributor contact points

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

Contact	Organization
Arliss Whiteside	BAE Systems E&IS
Markus U. Müller	lat/lon
Andreas Poth	lat/lon
Frank Warmerdam	DM Solutions Group
Paul Daisey	U.S. Census
George Percivall	NASA / GST, Inc.

v. Revision history

Date	Release	Editor	Primary clauses modified	Description
2002-07-26	0.0.1	Andreas Poth		initial version
2002-08-01	0.0.2	Markus Müller	7-11	added paragraphs 7-11
2002-08-14	0.0.3	Andreas Poth	6-11	chapter content slightly modified; schema definitions changed; example requests and responses added
2002-09-15	0.0.4	Markus Müller	i, 3	included changes requested by CRS SIG for public discussion paper status
2002-11-25	0.1.0	Markus Müller	All	Adapted spec to GML3 and included some changes responding to NASA comments on discussion paper. Dropped support of WKT.
2003-03-17	0.1.1	Markus Müller	All	Minor changes (support of Xlink, renamed some XML elements, grammatical changes, formatting)
2003-03-30	0.1.2	Markus Müller	7-11, Annex A	Changes according to Frank's and Arliss' comments and questions (added example transformation, corrected some minor mistakes).
2003-04-08	0.1.3	Arliss Whiteside	I to 6, Annex C	Edit front of document to improve it and to better match OGC/ISO document template
2003-04-26	0.1.4	Arliss Whiteside	vi, 5.3, 6, Annex D	Add UML diagrams and list of open questions
2003-06-24	0.1.5	Arliss Whiteside	vi, Annex F	Edited based on recent design decisions, added Annex F
2003-09-16	0.1.6	Arliss Whiteside	Annex B, E	Edited UML model and XML Schemas
2003-12-01	0.1.7	Arliss Whiteside	1, 3 to 11, Bibliography	Edited to build on OWS Common Implementation Specification 03-088r2
2004-02-06	0.1.8	Arliss Whiteside	3, 7.4, 8-11, B, E	Edited to build on OWS Common Implementation Specification 04-016
2004-05-27	0.1.9	Arliss Whiteside	vi, 3, 5.1, 7, 8.2, 8.3, B	Further edited to build on OWS Common Implementation Specification 04-016r4

Date	Release	Editor	Primary clauses modified	Description
2004-10-11	0.2.0	Stephane Fellah, Arliss Whiteside	All	Update to latest schema version after OWS-2 implementation and adding more explanation
2004-11-09	0.2.1	Arliss Whiteside	All	Many small corrections and improvements
2004-12-21	0.2.2	Arliss Whiteside	A, All	Added draft of Annex A, Many small corrections and improvements
2005-01-07	0.2.2	Arliss Whiteside	i, 2, 3, 6, 9.2.1, A, E.12	Some small improvements and corrections, many suggested by Charles Roswell
2005-01-31	0.3.0	Arliss Whiteside	Cover, i	Approved Recommendation Paper
2007-05-16	0.4.0	Arliss Whiteside	Most	Draft Revised Recommendation Paper, updated to use OWS Common 1.1
2007-10-08	0.4.1	Arliss Whiteside	cover, vii, B	Approved Revised Recommendation Paper

vi. Changes to the OGC Abstract Specification

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

vii. Future work

NOTE Generalization of the service interface now specified in this Discussion Paper has not yet been considered by the OGC. Such generalization of this interface will include considering combining the functions supported here with functions currently supported by other OGC (approved and draft) Web Services.

EDITOR'S NOTE This clause needs to be updated based on recent changes to this document.

This document should be updated to use GML 3.2. This document now uses the CRS Schemas for GML 3.1.1, in document [OGC 04-092r4].

The design and documentation questions that need to be resolved for the initial version of the WCTS specification, with the decisions made then being incorporated in this draft specification, include: (in no particular order)

a) What standardized CRS Application Schemas should be specified for "well-known" coordinate operations? (I assume most of these Application Schemas and XML documents will be specified in the CRS Schemas application profile.)

b) What standardized XML documents should be specified for "well-known" coordinate operations, operation methods, CRSs, and units? (I assume most of these XML documents will be specified in the CRS Schemas application profile.)

c) What standard srsName contents should be defined (in the CRS Schemas Application Profile or WCTS)?

d) How will the Transform operation compute a transformed "boundedBy" gml:Envelope in an enclosing feature or feature collection? (Notice that the name gml:Box was deprecated in GML 3.0, and gml:Envelope was re-defined in GML 3.1.1.)

e) What are the GML geometric primitive types that should be transformed by a WCTS server?

f) When the CRS Schemas are used to transfer actual definitions of user defined coordinate operations and/or CRSs, how will the required gml:id attributes be used? How will they be used in server inputs? How will they be used in server outputs? Specifically, what values will they be used when the transferred GML object is never expected to be the target of a remote reference?

g) What abilities will be included in the “limited” WCTS? (The questions in this list that are answered with non-minimum abilities might have lesser abilities defined as required in the limited WCTS profile.)

h) What standard GML Application Schema(s) should be specified for use in transforming one or more point positions?

i) Include position in inputs to IsTransformable operation?

A later version of this draft specification is expected to specify a limited set of abilities required to be implemented by a minimal conformant server implementation. Abilities that might be excluded from or limited in that minimal set include:

a) Not support transforming coverages, or not support transforming ordinary features

b) Reduce set of GML geometric primitive types handled

c) Reduce sets of coordinate operations and operation methods handled

d) Not support any Concatenated Operation abilities

e) Not use any specified CRS Application Schemas for well-known coordinate operations

f) Limit GML Application Schemas which can be used to define feature and feature collection types, in inputs to the Transform operation, more than just restricting the geometric primitive types included in the input features and feature collections.

g) Not compute a transformed "boundedBy" gml:Envelope in any enclosing feature or feature collection, by the Transform operation.

h) Not implement the optional "Sections" and “updateSequence” parameters in the GetCapabilities operation request

Future versions of this WCTS specification are expected to consider various possible expansions of the abilities specified herein, including:

a) Expand the set of GML geometric primitive types that can be handled by the Transform operation

b) Allow combinations of Feature, File, and Remote OWS (e.g., WFS) inputData in one Transform operation request. If so, specify what flexibility should be and can be supported

c) Allow transforming geometry elements separate from containing features

d) Provide special abilities for transforming a single point or few points, not otherwise contained in a GML geometry or feature. If so, specify what special abilities.

e) Support returning information on the quality of the available transformation(s), from the IsTransformable operation

f) Support returning information on the quality of the transformation performed, from the Transform operation

g) Support updating metadata to reflect the transformation performed, from the Transform operation

h) Require supporting transformations that can change the type of a geometry element. (For example, North Pole point becomes a line.)

i) Check the contents of the input srsName attributes, in geometry elements being transformed. If so, specify what form(s) of srsName contents are supported within the geometries included in, or referenced by, Transform operation requests.

j) When appropriate, change the locations of srsName attributes in geometries transformed, relative to input locations. (In GML 3.1.1, the srsName must either be included in each geometry element or inherited from an enclosing geometry element. The GML 3.1.1 the srsName can be inherited from the "boundedBy" gml:Envelope element in an enclosing feature or feature collection.)

k) Support use of two-way coordinate operations and operation methods. If so, specify required type(s) of support. (A coordinate operation and its operations method are generally two-way, as defined in Abstract Specification Topic 2.)

l) Support user-defined operation methods. If so, specify how operation methods can be user specified any limitations on such user-defined operation methods.

Foreword

This version of this future specification supersedes all previous versions, including OGC documents 05-013, 04-072, and 02-061r1, titled Web Coordinate Transformation Service draft Implementation Specification. This engineering report supplements, and does not supersede, OGC document 00-009 titled OpenGIS Implementation Specification: Coordinate Transformation Services.

This WCTS interface uses profiles of the Geography Markup Language (GML) Version 3.1.1 [OGC 04-092r4], which partially-specifies XML-encoding of geospatial data. The geometries in GML are largely based on ISO 19107:2003.

This specification includes eight annexes; Annexes A, B, C, and G are normative and the other four 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. To date, no such patent rights have been claimed or identified.

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

This document specifies the interface to a Web Coordinate Transformation Service (WCTS), which can be used by geospatial applications and other services. Such transformation of geospatial data from one coordinate reference system (CRS) to another is frequently required when using data from different sources in one application. That is, geospatial data are often stored in different coordinate reference systems (CRSs). To use together data stored in different CRSs, such data must be transformed into the same CRS. Not all applications or services are capable of performing such transformations.

This document specifies an "OGC Web Service" type of interface to a service that performs coordinate transformations. Such transformations include all the types of coordinate operations, including both "transformations" and "conversions". This service inputs digital features or coverages in one CRS and outputs the same features in a different CRS. The service inputs include identifications of the input and output CRSs, and optionally the coordinate transformation between these CRSs.

This WCTS interface specifies five operations that can be requested by a client and performed by a WCTS server. Those operations are:

a) Transform (required implementation) – This operation allows a client to request transformation of the coordinates in a specified set of features, and to receive back the transformed features. This operation allows the client to identify the CRS of the input features and the desired CRS of the output features. Each feature can be a single feature, a feature collection containing multiple features, or a coverage.

b) GetCapabilities (required implementation) – This operation allows a client to request and receive back a service metadata (or Capabilities) document that describes the abilities of the specific server implementation.

c) IsTransformable (optional implementation) – This operation allows a client to ask a server if it can perform any coordinate transformation from one identified CRS into another. This operation also allows the client to ask if this transformation can be performed on specified geometric primitive or coverage types. The response back is binary, plus problem information when the answer is false.

d) GetTransfomation (optional implementation) – This operation allows a client to request and receive back the XML-encoded definition of the server-known coordinate transformation(s) from one identified CRS into another.

e) GetResourceByID (optional implementation) – This operation allows a client to request and receive back XML-encoded definitions of one or more identified abilities provided by the WCTS server. Those abilities can be coordinate transformations, coordinate reference systems (CRSs), or operation methods.

Web Coordinate Transformation Service (WCTS) draft Implementation Specification

1 Scope

This document specifies the interface to a Web Coordinate Transformation Service (WCTS), which can be used by geospatial applications and other services. More specifically, this service transforms geospatial coordinates from one coordinate reference system (CRS) into another, as requested by clients of this service. This WCTS is an OGC Web Service, with appropriate similarities to other OGC Web Services.

The central ability of the specified WCTS is to transform digital feature and coverage data from one CRS to another. The supported abilities include allowing a client to:

a) Provide the feature or coverage data to be transformed.

b) Identify the current CRS of the data to be transformed.

c) Identify the needed CRS of the transformed data.

d) Identify the desired coordinate transformation between the two CRSs, when desired.

e) Check if a coordinate transformation between two identified CRSs is supported by a WCTS server.

f) Determine the source and target CRSs that are supported by a WCTS server.

g) Obtain definitions of the CRSs that are supported by a WCTS server.

h) Determine the coordinate transformations that are supported by a WCTS server.

i) Obtain definitions of the coordinate transformations that are supported by a WCTS server.

j) Determine the geometric primitive types that are supported by a WCTS server.

k) Determine the coverage types that are supported by a WCTS server.

l) Determine if user-defined coordinate transformations are supported by a WCTS server.

m) Determine the operation methods that are supported by a WCTS server which can be used in user-defined coordinate transformations.

n) Obtain definitions of the operation methods that are supported by a WCTS server.

o) Provide user-defined coordinate transformation definitions to a WCTS server.

p) Determine if user-defined CRSs are supported by a WCTS server.

q) Provide user-defined CRS definitions to a WCTS server.

NOTE The above list can be considered to be a list of requirements on the design of this WCTS interface.

This specification is applicable to geospatial data that are XML encoded using GML 3.1.1 [OGC 04-092r4], and profiles thereof. The geometries in GML 3.1.1 are largely based on ISO 19107:2003. This specification is applicable to geospatial data that use CRSs and coordinate transformations which are or can be defined using the XML encoding of coordinate reference system definitions specified in GML 3.1.1. Those coordinate reference system definitions are largely based on OGC Abstract Specification Topic 2 [OGC 05-103], which is the proposed revision of ISO 19111.

This specification largely uses the terminology used in Abstract Specification Topic 2. However, all the types of coordinate operations defined in that document are usually referred to here as transformations or coordinate transformations. (That document separately defines ConcatenatedOperations, PassThroughOperations, Conversions, and Transformations, as the concrete subtypes of the abstract CoordinateOperation.)

2 Conformance

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

In order to conform to this specification, a software implementation shall choose to implement one or more of the conformance classes specified in Annex B (normative).

3 Normative references

The following normative documents contain provisions which, through reference in this document, 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.

European Petroleum Survey Group database: *EPSG Geodesy Parameters*, available through http://www.epsg.org/

ISO 19123:2005, *Geographic information - Schema for coverage geometry and functions*

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 05-099r2, GML 3.1.1 simple dictionary profile

OGC 05-103, *The OpenGIS® Abstract Specification Topic 2: Spatial referencing by coordinates*

OGC 06-023r1, *URNs of definitions in ogc namespace*

OGC 06-047r1, *GML simple features profile*

OGC 07-067r2, *Web Coverage Service (WCS),* Version 1.1.1

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

NOTE This OWS Common Specification contains a list of normative references that are also applicable to this Implementation Specification.

4 Terms and definitions

For the purposes of this specification, the definitions given in Abstract Specification Topic 2: Spatial referencing by coordinates [OGC 05-103] and in OGC Web Services Common Specification [OGC 06-121r3] 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 spatial, temporal, or spatiotemporal domain

EXAMPLE Examples include a raster image, polygon overlay, or digital elevation matrix.

4.2
feature
abstraction of real world phenomena [ISO 19101]

4.3
geometric primitive
a geometric object that is not decomposed further into other primitives

NOTE All geometric primitives are oriented in the direction implied by the sequence of their coordinate tuples.

5 Conventions

5.1 Abbreviated terms

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

OWS-2 OGC Web Services interoperability initiative, phase 2

RFC Request for Comments

WCTS Web Coordinate Transformation Service

5.2 UML notation

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 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 light grey background (15%).

5.4 Platform-neutral and platform-specific specifications

As specified in Subclause 5.4 of the OWS Common Specification [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, 8.3.2, 9.2.1. 9.3.2, 10.2.1, 10.3.2, 11.2.1, 11.3.2, 12.2.1and 12.3.2.

EXAMPLES 2 Platform-specific specifications for KVP encoding are contained in Subclauses 8.2.4, 9.2.3, 10.2.3, 11.2.3 and 12.2.4.

EXAMPLES 3 Platform-specific specifications for XML encoding are contained in Subclauses 8.2.5, 8.3.4, 9.2.4, 9.3.6, 10.2.3, 10.3.3, 11.2.4, 11.3.4, and 12.2.5.

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 Table 1 of [OGC 06-121r3].

6 WCTS overview

A Web Coordinate Transformation Service (WCTS) transforms digital geospatial data from one Coordinate Reference System (CRS) to another. The geospatial data transformed is digital feature data, including digital coverages. Such transformations include all the types of coordinate operations, including both coordinate "transformations" and "conversions". This service inputs digital features in one CRS and outputs the same features in a different CRS. The service inputs include the identifications of the input and output CRSs, and optionally the coordinate transformation between these CRSs.

A Web Coordinate Transformation Service (WCTS) can be used by many different geospatial applications and other services. Transformation of geospatial data from one Coordinate Reference System (CRS) to another is a frequent requirement when using data from different sources in one application. That is, geospatial data are often stored in different coordinate reference systems (CRSs). To integrate or otherwise use together data stored in different CRSs, such data must be transformed or converted into the same CRS. Not all applications or services are capable of directly performing such transformations. More information on expected uses of a WCTS is provided in Annex C: Use cases (informative).

The WCTS is an "OGC Web Service" type of interface, where client and server software interact using HTTP messages. This WCTS interface specifies five operations that can be requested by a client and performed by a WCTS server. Those operations are:

a) Transform (required implementation by servers) – This operation allows a client to request transformation of the coordinates in a specified set of features, and to receive

back the transformed features. This operation allows the client to identify the CRS of the input features and the desired CRS of the output features. Each feature can be a single feature, a feature collection containing multiple features, or a coverage.

b) GetCapabilities (required implementation by servers) – This operation allows a client to request and receive back a service metadata (or Capabilities) document that describes the abilities of the specific server implementation. This operation also supports negotiation of the specification version being used for client-server interactions.

c) IsTransformable (optional implementation by servers) – This operation allows a client to ask a server if it can perform any coordinate transformation from one identified CRS into another. This operation also allows the client to ask if this transformation can be performed on specified geometric primitive or coverage types. This transformation can be a single transformation, or a ConcatenatedOperation which combines multiple single transformations. The response back is binary, plus problem information when the answer is false.

d) GetTransfomation (optional implementation by servers) – This operation allows a client to request and receive back the definition of the server-known coordinate transformation(s) from one identified CRS into another. Each transformation can be a single transformation, or a ConcatenatedOperation that combines multiple single transformations. Each transformation definition is encoded in XML using the GML 3.1.1 Grid CRSs profile [OGC 05-096r1].

e) GetResourceByID (optional implementation by servers) – This operation allows a client to request and receive back definitions of one or more identified abilities provided by the WCTS server. Those abilities can be coordinate transformations, coordinate reference systems (CRSs), or operation methods. Each transformation, CRS, or operation method definition is encoded in XML using the GML 3.1.1 Grid CRSs profile [OGC 05-096r1].

The GetCapabilities operation has many similarities to the GetCapabilities operation in other OGC Web Services, including the WMS, WFS, and WCS. The other four WCTS operations also have similarities to other operations in other OWSs. Many of the WCTS interface aspects that are common with other OWSs are thus specified in the OWS Common Specification [OGC 06-121r3]. Many of these common aspects are normatively referenced herein, instead of being repeated in this specification.

Figure 1 is a simple UML diagram summarizing the WCTS interface. This class diagram shows that the WCTService interface class inherits the GetCapabilities and GetResourceByID operations from the OGCWebService interface class, and adds the "transform", isTransformable, and getTransformation operations. (This capitalization of names uses the OGC/ISO profile of UML.) A more complete UML model of the WCTS is provided in Annex E (informative).

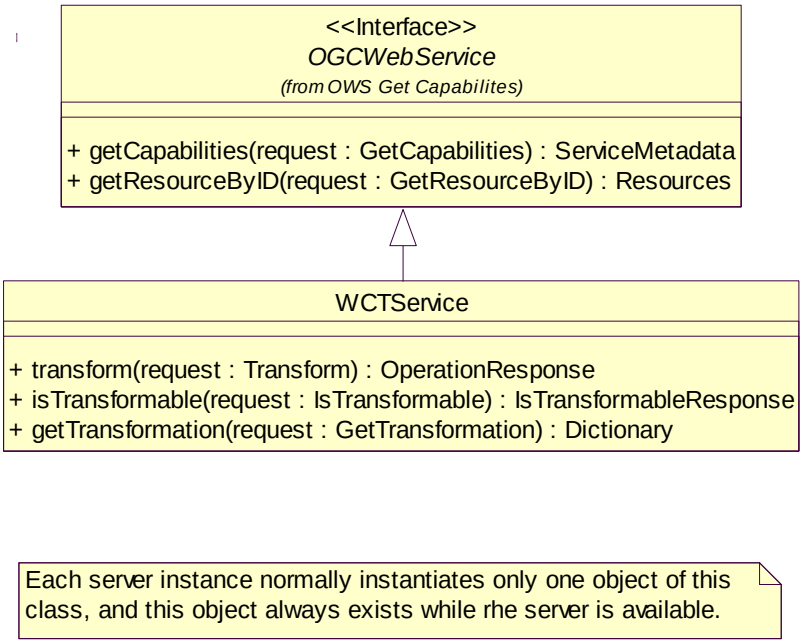

Figure 1 — WCTS interface UML diagram

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 modelling these data structures are included in the complete UML model in Annex E.

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

7 Shared aspects

7.1 Introduction

This clause specifies aspects of Web Coordinate Transformation Service behaviour that are shared by several operations.

7.2 Shared data structures

This clause specifies four data structures used by multiple operation requests and responses specified in the following clauses. The parameter names, meanings, and formats shall be as specified below. These data structures are shown graphically in the UML class diagram in Figure 2

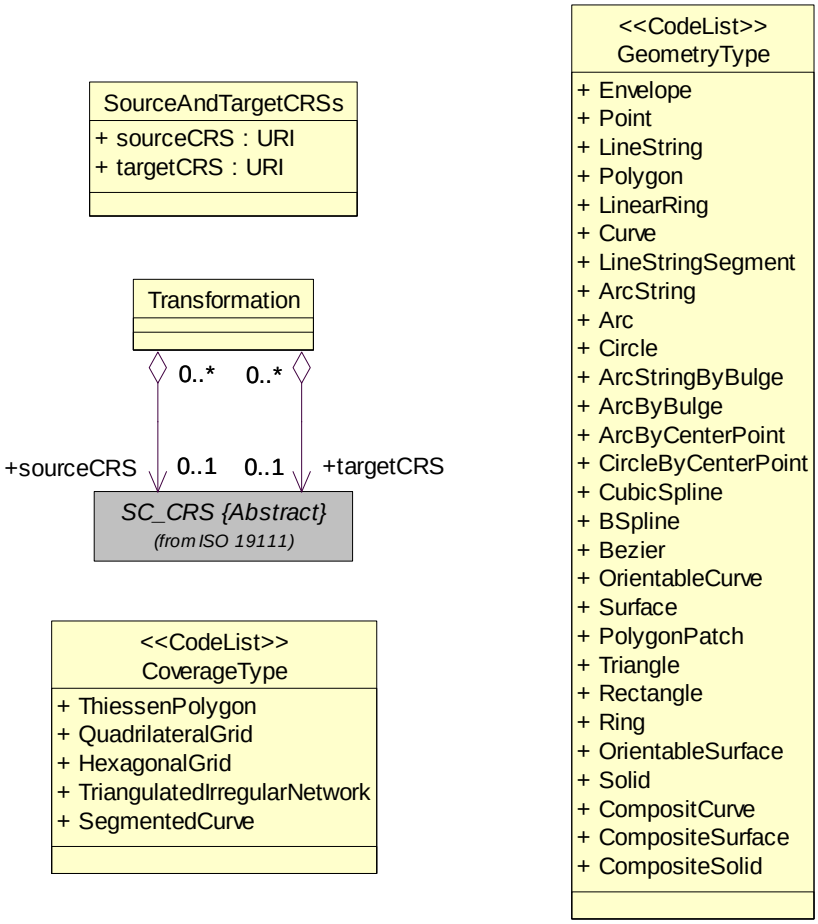

Figure 2 — WCTS shared operation data UML class diagram

Table 1 specifies the class (or data structure) named SourceAndTargetCRSs, which references the input and desired output coordinate reference systems (CRSs), and is used by multiple operation requests.

Table 1 — SourceAndTargetCRSs data structure

Names	Definition	Data type and value	Multiplicity
sourceCRS SourceCRS	Reference to input coordinate reference system	URI [a] Values defined in service metadata or in data known to client	One (mandatory)
targetCRS TargetCRS	Reference to desired output coordinate reference system	URI [a] Values defined in service metadata or in data known to client	One (mandatory)
a A URI shall reference a CRS as specified in Subclause 10.3 of [OGC 06-121r3].			

Table 2 specifies the class (or data structure) named Transformation, which references or defines a coordinate Transformation, and is used by multiple operation requests and responses.

Table 2 — Transformation data structure

Names	Definition	Data type and value	Multiplicity and use
coordinate Operation (none)	Reference to or definition of coordinate operation	URI [a] or CC_CoordinateOperation [b] Values defined in service metadata or known to client	One (mandatory)
sourceCRS SourceCRS	Reference to or definition of input coordinate reference system	URI [a] or SC_CRS [c] Values defined in service metadata or known to client	Zero or one (optional) Include when operation is Conversion
targetCRS TargetCRS	Reference to or definition of output coordinate reference system	URI [a] or SC_CRS [c] Values defined in service metadata or known to client	Zero or one (optional) Include when operation is Conversion

a A URI shall reference a CRS or coordinate operation as specified in Subclause 10.3 of [OGC 06-121r3].

b The CC_CoordinateOperation UML class defines a coordinate operation and is specified in Subclause 12.4 of OGC Abstract Specification Topic 2. GML 3.1.1 [OGC 04-092r4] specifies how to XML encode the CC_CoordinateOperation class. A KVP encoding of the CC_CoordinateOperation class is not considered practical.

c The SC_CRS UML class defines a CRS and is specified in Subclause 9.4 of OGC Abstract Specification Topic 2. GML 3.1.1 [OGC 04-092r4] specify how to XML encode the SC_CRS class. A KVP encoding of the SC_CRS class is not considered practical.

Table 3 specifies the CodeList class named CoverageType, which lists various possible coverage types, and is used by multiple operation requests and responses. These types are based on ISO 19123.

Table 3 — CoverageType values

CoverageType value	Definition
ThiessenPolygon	2D Coverage with domain consisting of set of polygons that each enclose one of a set of points in a plane so as to include all positions that are closer to that point than to any other point in the set
QuadrilateralGrid	2D Coverage with domain consisting of set of points defined by the intersections of two sets of straight lines in one plane, in which the lines of each set intersect lines of other set in a systematic way
GRID	Abbreviation for QuadrilateralGrid described above
HexagonalGrid	2D Coverage with domain consisting of network of hexagons centred on a set of grid points in a plane, using interpolation between the centres of the hexagons surrounding the grid points
TriangulatedIrregular Network	2D Coverage with domain consisting of set of triangles which partition the convex hull of a set of points in a plane into a computationally unique set of non-overlapping triangles, with each triangle formed by three of the points
TIN	Abbreviation for triangulated irregular network described above
SegmentedCurve	1D Coverage with domain consisting of set of curves that includes all positions in all curves in a set of curves

Table 4 specifies the CodeList class named GeometryType, which lists possible geometry primitive types, and is used by multiple operation requests and responses. These types are based on GML 3.1.1, using the XML Schema type name without the "Type" suffix.

Table 4 — GeometryType values

GeometryType value
Envelope
Point
LineString
Polygon
LinearRing
Curve
LineStringSegment
ArcString
Arc
Circle
ArcStringByBulge
ArcByBulge
ArcByCenterPoint
CircleByCenterPoint
CubicSpline
BSpline
Bezier
OrientableCurve
Surface
PolygonPatch
Triangle
Rectangle
Ring
OrientableSurface
Solid
CompositeCurve
CompositeSurface
CompositeSolid

These shared UML classes are XML encoded in the attached normative XML Schema Document named wctsCommon.xsd. The CoverageType and GeometryType 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 wctsCommon.xsd, the normal values of CoverageType and GeometryType are NOT encoded in the XML Schema, but in the default-referenced GML Dictionaries named coverageType.xml and geometryType.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 or different values to be more easily specified and used.

7.3 Operation request encoding

The encoding of operation requests may use HTTP GET with KVP encoding and/or HTTP POST with XML encoding as specified in Clause 11 of [OGC 06-121r3]. SOAP encapsulation of XML encoded operation requests may also be used, as specified in Annex G of this document. Table 5 summarizes the WCTS operations and their encoding methods defined in this specification.

Table 5 — Operation request encoding

Operation	Request encoding
Transform (mandatory)	XML and optional SOAP, KVP
GetCapabilities (mandatory)	KVP and optional XML, SOAP
IsTransformable (optional)	XML, SOAP, and/or KVP
GetTransformation (optional)	XML, SOAP, and/or KVP
GetResourceByID (optional)	XML, SOAP, and/or KVP

The KVP values may include XML elements. For example, this is the case when encoding a Transform operation request with an in-line feature. That feature shall be encoded using GML 3.1.1 encoded to use it the HTTP GET request.

7.4 Standard operation exceptions

When a WCTS server encounters an error while performing an operation, it shall return an exception report message as specified in Subclause 7.4 of [OGC 06-121r3]. For most WCTS operations, the allowed standard exception codes shall include those listed in Table 6. For each listed exceptionCode, the contents of the "locator" parameter value shall be as specified in the right column of Table 6.

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 6 — Exception codes for WCTS optional operations

exceptionCode value	Meaning of code	"locator" value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
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
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit "locator" parameter

8 Transform operation (mandatory)

8.1 Introduction

The mandatory Transform operation allows WCTS clients to request transformation of coordinates from one coordinate reference system (CRS) into another. Input data is provided in the operation request as one or more references to a remote resource or local payload (attachment to the message), in a supported format. A remote resource is referenced using a URL, which can be a request to a specific server providing the data. For example, this remote server may be a WFS or WCS. The WCTS may optionally support user-defined coordinate transformations, in which case it is necessary for the client to provide the parameter values for the transformation.

This clause first discusses the Transform operation request and response for digital features, largely ignoring grid coverages. The additional Transform operation aspects for grid coverages are then discussed in Subclause 8.5.

8.2 Operation request

8.2.1 Transform request parameters

A request to perform the Transform operation shall include the parameters and data structures shown graphically in the UML diagram in Figure 3, and specified in Table 7 and Table 8.

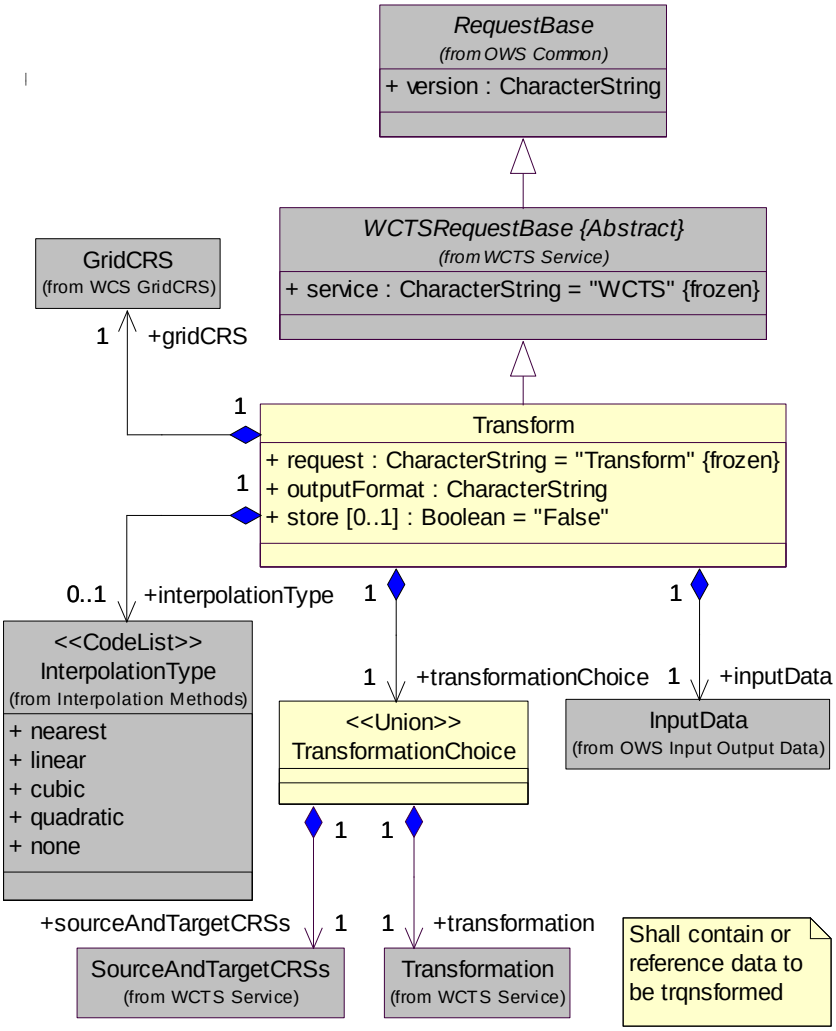

Figure 3 — WCTS Transform request UML class diagram

NOTE 1 The first three parameters listed below are largely copied from Table 26 in Subclause 9.2.1 of [OGC 06-121r3]. The InterpolationType parameter listed below is largely copied from Subclause I.4.1 of WCS 1.1 [OGC 07-067r2]. The OutputFormat parameter listed below is largely copied from Table 32 in Subclause 10.6.1 of [OGC 06-121r3].

Table 7 — Parameters in Transform operation request

Names	Definition	Data type and value	Multiplicity and use
service service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, namely "WCTS"	One (mandatory)
request request	Operation name	Character String type, not empty Value is operation name, namely "Transform"	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)
transformation Choice (none)	Identification or definition of desired coordinate operation	TransformationChoice data structure, see Table 8	One (mandatory)
gridCRS GridCRS	Definition of output GridCRS referenced by transformation	GridCRS data structure, see Annex G of WCS 1.1 [OGC 07-067r2]	Zero or one (optional) Include when transforming grid coverage
inputData InputData	Data to be transformed	InputData data structure, see 13.5 of [OGC 06-121r3]	One (mandatory)
interpolation Type [a] Interpolation Type	Identifier of spatial interpolation type which should be used to transform coverage	InterpolationType code list, see I.4.1 of [OGC 07-067r2] Values defined in service metadata	Zero or one (optional) Include when transforming coverage(s) using spatial interpolation
outputFormat OutputFormat	Identifier of output format to be used for the transformed features or coverage(s) [b]	MIME type, see Subclause 10.5 of [OGC 06-121r3] Allowed values defined in service metadata	Zero or one (optional) Include when output data not in input format
store store	Specifies if transformed data to be stored as remote resource(s)	Boolean Values are: true or false	Zero or one (optional) Include when not default (false, return in response)

a Transforming a grid coverage usually requires resampling of coverage values to obtain the values at different grid points. For example, a coverage can be transformed from an unrectified grid to a georectified grid. When transforming a grid coverage, the SourceCRS and TargetCRS parameters shall completely specify the grids of the input and output coverages. These SourceCRS and TargetCRS parameters can be included in either the Transformation or the SourceAndTargetCRSs data structures.

b Allows WCTS servers to perform reformatting. The output formats supported by a WCTS are listed in the Contents section of the Capabilities document.

EDITOR'S NOTE This draft does not support different interpolation types for different components of a coverage range (as allowed in WCS 1.1).

EDITOR'S NOTE The Transformations (or coordinate operations) implemented by WCTS servers are currently limited to transformations in one direction only (not including the reverse direction). The group working on the WCTS specification has not yet considered what would be required to implement two-way transformations.

EDITOR'S NOTE It is not clear whether an optional bounding box would be useful in the Transform operation request.

Table 8 — Parameters in TransformationChoice data structure

Names	Definition	Data type and value	Multiplicity and use
sourceAnd TargetCRSs SourceAnd TargetCRSs	References to SourceCRS and desired TargetCRS	SourceAnd TargetCRSs data structure, see Table 1	Zero or one (Mutually exclusive) [a]
transformation Transformation	Identification or definition of desired coordinate operation	Transformation data structure, see Table 2	Zero or one (Mutually exclusive) [b]

a Included when the server is permitted to select any coordinate operation which it knows about that can transform coordinates from the identified sourceCRS to the targetCRS.

b Included when client is specifying a specific coordinate operation, possibly a user-defined coordinate operation. In this use, this element shall either:

* Reference a well-known coordinate operation, whose definition is known to the WCTS server

* Contain a URL from which that transformation definition can be retrieved, using GML encoding

* Contain the coordinate operation definition object, using GML encoding, and a xlink:href value containing the URN that references this definition

8.2.2 Use of InputData structure

Each ReferenceGroup in the InputData structure shall reference separate data to be transformed, each in one contained Reference data structure. The data referenced may be attached to the Transform request or referenced from an external source. In addition to the data to be transformed, each ReferenceGroup may also reference associated metadata, in separate References. For example, this metadata can include a human readable title and/or abstract, plus a link to more metadata.

NOTE The associated referenced metadata will be either passed unchanged to the Transform output OperationResponse, or modified to reflect the coordinate transformation performed, as specified in Subclause 8.3.1. For example, attached data quality metadata might be updated to reflect the coordinate transformation applied.

For digital feature data, the data to be transformed can be encoded as either a gml:_Geometry or a gml:_Feature that contains a gml:_Geometry. For grid coverage data, the grid point values can be encoded in any suitable format that is supported by the WCTS server. The coverage metadata may be encoded in the same coverage file and/or in separate metadata files that are encoded in XML.

The xlink:role in the Reference to data to be transformed shall be included and used to indicate the type of the data to be transformed, as specified in Table 9.

Table 9 — Data types to be transformed

Data type	Value of xlink:role
Feature collection	FeatureCollection
Single feature	Feature
Geometry	Geometry
Grid coverage	Coverage

8.2.3 Transform request implementation requirements

The "Multiplicity and use" columns in Table 7 and Table 8 specify the optionality of each listed parameter in the Transform operation request. Table 10 specifies the implementation of those parameters by servers and clients that implement the WCTS.

Table 10 — Implementation of parts of Transform operation request

Name	Multiplicity	Server implementation	Client implementation
service	One (mandatory)	Each shall be implemented by all servers [a]	Each shall be implemented by all clients, using allowed values
request	One (mandatory)		
version	One (mandatory)		
InputData	One (mandatory)		
SourceAnd TargetCRSs	Zero or one (Mutually exclusive)	Each shall be implemented by all servers [a]	Each should be implemented by all clients
Transformation			
GridCRS	Zero or one (optional)	Each shall be implemented by all servers that can transform coverages [a]	Each should be implemented by all clients that can transform coverages
Interpolation Type	Zero or one (optional)		
OutputFormat	Zero or one (optional)	Shall be implemented by all servers that offer different output formats [a, b]	Should be implemented by a client [c]
store	Zero or one (optional)	Shall be implemented by all servers that can transform coverages [a, b, d]	May be implemented by a client [c]

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

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

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

d The "store" parameter may also be implemented by servers that cannot transform coverages.

8.2.4 Transform request KVP encoding (optional)

WCTS servers may implement HTTP GET transfer of the Transform operation request, using KVP encoding. The KVP encoding of the Transform operation request is limited to transforming one gml:_Feature or gml:_Geometry, and shall use the parameters specified

in Table 11. The parameters listed in Table 11 shall be as specified in Table 7 and Table 8 above.

Table 11 — Transform operation request URL parameters

Name and example [a]	Optionality	Definition and format
service=WCTS	Mandatory	Service type identifier
request= Transform	Mandatory	Operation name
version=0.0.0	Mandatory	Specification and schema version for this operation
sourceCRS=urn:ogc:def:crs:EPSG:6.3:4326	Optional [b]	Identifier URI of input coordinate reference system
targetCRS=urn:ogc:def:crs:EPSG:6.3:32611	Optional [b]	Identifier URI of desired output coordinate reference system
transformation=urn:ogc:def:coordinateOperation:EPSG:6.6:TBD	Optional [b]	Identifier URI of desired coordinate operation
inputData=TBD	Mandatory	feature data to be transformed, either encoded inline as GML or referenced using a URL
interpolation Type=linear	Optional	Identifier of interpolation type that should be used to transform a coverage
outputFormat=TBD	Optional [c]	Identifier of output format to be used for the transformed features or coverage [c]
store=true	Optional	Boolean (true and false values) used to indicate if the transformed data needs to be stored on a remote resource or returned directly in the response. By default, it is set to false. (return directly in response)

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

b Either the SourceCRS and TargetCRS shall be included, or the Transformation shall be included. If the SourceCRS and TargetCRS are included, the server is permitted to select any coordinate operation which it knows about that can transform coordinates from the sourceCRS to the targetCRS identified. When the Transformation is included, this "transformation" shall not be a Conversion and shall either:

* Reference a coordinate operation listed in the Contents part of the Capabilities XML document
* Contain a URL from which the definition of a coordinate operation can be retrieved using GML encoding

c Allows advanced WCTS servers to perform reformatting. The output formats supported by a WCTS are listed in the Contents section of the Capabilities document.

8.2.5 Transform request XML encoding (required)

All WCTS servers shall implement HTTP POST transfer of the Transform operation request, using XML encoding only. The XML schema fragment for the Transform request is specified by the Transform element in the attached normative wctsTransform.xsd file. This fragment uses the gmlGridCRSsProfile of GML 3.1.1, as specified in [OGC 05-096r1], which is referenced in the attached normative wctsCommon.xsd file. This fragment also uses the interpolationMethods.xsd part of the WCS 1.2 XML Schema.

8.2.6 Transform request SOAP encoding

WCTS servers may implement SOAP version 1.2 transfer of the Transform operation request as specified in Annex G, using the XML encoding referenced above.

8.3 Operation response

8.3.1 Transform operation exceptions

When a web coordinate transformation server encounters an error while performing a Transform operation, it shall return an exception report message as specified in Subclause 7.4 of [OGC 06-121r3]. The allowed exception codes shall include those listed in Table 12. For each listed exceptionCode, the contents of the "locator" parameter value shall be as specified in the right column of Table 12.

NOTE The first three values listed below are copied from Table 25 in Subclause 8.3 of [OGC 06-121r3].

Table 12 — Exception codes for Transform operation

exceptionCode value	Meaning of code	"locator" value
MissingParameterValue	Request does not include a parameter value, and this server did not declare a default value for that parameter [a]	Name of missing parameter
InvalidParameterValue	Request contains an invalid parameter value	Name of parameter with invalid value
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit "locator" parameter
NoInputData	No input data was available from a specified source for input data [b]	Value of xlink:href in InputData
InvalidArea	One or more points in InputData are outside the domainOfValidity of the transformation	Value of xlink:href in InputData
TransformException	Used where a computation error occurs while performing the transformation	None, omit "locator" parameter
UnsupportedCombination	Operation request contains output CRS that can not be used within output format	Name of Format or name of CRS
NotEnoughStorage	Operation request specifies "store" result, but not enough storage is available to do this	None, omit "locator" parameter

a This code shall be used when the optional OuputFormat parameter is omitted, but the server does not support as an output format the same format as used by the input data.

b The specified source for input data could be the URL of the file containing a gml:_Feature or an OWS operation request to get a gml:_Feature. If remote access of input data was requested, and an exception report or other error response was returned by that resource, then the first ExceptionText element shall contain the received exception report or other error response.

8.3.2 Transform normal response

The normal response to a Transform operation request is at least one GML 3.1.1 feature, feature collection, geometry, or (usually not GML) grid coverage. More precisely, a response from the Transform operation shall use the OperationResponse data structure specified in Subclause 13.4 of [OGC 06-121r3]. Each ReferenceGroup in the

OperationResponse shall reference separate data that was transformed, either attached or referenced by URL.

In addition to the data transformed, each ReferenceGroup may also reference associated metadata. For example, this metadata can include a human readable title and/or abstract, plus a link to more metadata. The associated referenced metadata shall be either passed unchanged to the Transform output OperationResponse, or modified to reflect the coordinate transformation performed, as may be specified in a WCTS Profile.

For digital feature data, the data transformed will be GML 3.1.1 encoded as it was in the InputData, either a gml:_Geometry or a gml:_Feature that contains a gml:Geometry. For grid coverage data, the grid point values can be encoded in any suitable format that is supported by the WCTS server. The coverage metadata can be encoded in the same coverage file and/or in separate metadata files that are encoded in XML.

The xlink:role in the Reference to data transformed shall be included and used to indicate the type of the data transformed, as specified in Table 9.

8.3.3 Transform response encodings

The encoding of the Transform operation response consists of an OperationResponse XML document (see 8.3.4 below), packaged according to the request encoding and the value of the "store" parameter (specified in Table 7). WCTS servers shall implement Transform response encodings according to Table 13.

Table 13 — Transform response encodings

Request encoding	store=true [a]	store=false *(default)*
KVP or XML	OperationResponse alone (see 8.3.4)	OperationResponse in MIME Multipart message (see 8.3.5)
SOAP	OperationResponse in SOAP envelope (see 8.3.6)	OperationResponse in SOAP envelope with Attachments (see 8.3.5)
a Applies only when the server supports the "store" parameter (see Subclause 8.2.3).		

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 OperationResponse data structure (within a SOAP envelope in the case of a SOAP request) with references to the other files as indicated in 8.3.4 and 8.3.6 below.

When the "store" parameter is absent or has the value "false", the server shall transfer the complete OperationResponse 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 OperationResponse shall reference the other parts of the MIME multi-part message (or SOAP attachments) as indicated in 8.3.5 below.

8.3.4 Transform response XML encoding (required)

The XML Schema fragment for the OperationResponse is specified in the normative owsInputOutputData.xsd file attached to [OGC 06-121r3]. If multiple data sets are input,

the transformed data shall be returned in the same order as the input (possibly in attachments in a multipart MIME message).

8.3.5 MIME multipart response

Transform operation responses transferred using MIME multipart shall be encoded according to [IETF RFC 2387], with the first part containing the XML encoded OperationResponse element. Every part of the multipart message shall have a content-id. The part containing the OperationResponse element shall have a content-id equal to "urn:ogc:wcts:1.1:transformResponse"; other parts may have any content-id values. The Reference elements in the OperationResponse 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 OperationResponse 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 OperationResponse can (but normally should not) reference resources external to the MIME multipart message.

8.3.6 Transform response SOAP encoding (optional)

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

When the "store" parameter is absent or has the value "false", responses to Transform 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 8.3.5, with the OperationResponse element contained within a SOAP 1.2 envelope, encoded as specified in Annex G.

8.4 Feature transformation examples

8.4.1 Transform request example

If a client wants to transform a GML Feature Collection from EPSG:4326 to EPSG:32611 using the default transformation, it may issue the following XML encoded Transform operation request:

```
<?xml version="1.0" encoding="UTF-8"?>
<Transform xmlns="http://www.opengis.net/wcts/0.0"
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/wcts/0.0
../../wctsTransform.xsd http://www.opengis.net/ows/1.1
../../../../ows/1.1.0/owsAll.xsd" ="WCTS" version="0.0.0">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-05-01-->
   <SourceCRS>urn:ogc:def:crs:EPSG:6.0:4326</SourceCRS>
   <TargetCRS>urn:ogc:def:crs:EPSG:6.0:32611</TargetCRS>
```



```
   <ows:InputData>
      <ows:ReferenceGroup>
         <ows:Title>The Capital feature "Aguascalientes"</ows:Title>
         <ows:Reference xlink:href="Aquascalientes4326.xml"
xlink:role="FeatureCollection"/>
      </ows:ReferenceGroup>
   </ows:InputData>
   <OutputFormat>text/xml; gmlVersion=3.1.1</OutputFormat>
</Transform>
```

The above example requests coordinate transformation between two well-known CRSs, each referenced using a URN. Alternately, a well-known coordinate transformation can be requested, by providing its URI. For a user-defined coordinate transformation, the desired transformation must be specified in a concrete element that substitutes for the gml:_CoordinateOperation abstract element.

The above example references a feature collection that can be XML encoded as:

```
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns="http://www.opengis.net/examples"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/examples
exampleCapital.xsd">
   <!-- Primary editor: Arliss Whiteside. Last updated 2006-10-10-->
   <gml:boundedBy>
      <gml:Envelope srsName="urn:ogc:def:crs:EPSG:6.0:4326">
         <gml:lowerCorner>-115.467123 16.743654</gml:lowerCorner>
         <gml:upperCorner>-88.291157 32.654688</gml:upperCorner>
      </gml:Envelope>
   </gml:boundedBy>
   <gml:featureMember>
      <Capital gml:id="AC">
         <cityName>Aguascalientes</cityName>
         <stateName>AGUASCALIENTES</stateName>
         <gml:pointProperty>
            <gml:Point srsName="urn:ogc:def:crs:EPSG:6.0:4326">
               <gml:pos>-102.28969800 21.88751600</gml:pos>
            </gml:Point>
         </gml:pointProperty>
      </Capital>
   </gml:featureMember>
</gml:FeatureCollection>
```

This XML document could be either bundled with the Transform operation request or accessed remotely. As indicated, this feature collection is XML encoded using an example GML 3.1.1 Application Schema, named exampleCapital.xsd and listed in Subclause F.2.

8.4.2 Transform response example

The response to a request for transforming a point from EPSG:4326 to EPSG:32611 could be, if no exception occurs:

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



```
<OperationResponse xmlns="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/ows/1.1
../../../../ows/1.1.0/owsAll.xsd">
   <!-- Primary editor: Arliss Whiteside. Last updated 2006-10-10-->
   <ReferenceGroup>
         <Title>The Capital feature "Aguascalientes"</Title>
         <Reference xlink:href="Aquascalientes32611.xml"
xlink:role="FeatureCollection"/>
   </ReferenceGroup>
</OperationResponse>
```

The above example references a feature collection that can be XML encoded as:

```
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns="http://www.opengis.net/examples"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/examples
exampleCapital.xsd">
   <!-- Primary editor: Arliss Whiteside. Last updated 2006-10-10-->
   <gml:boundedBy>
      <gml:Envelope srsName="urn:ogc:def:crs:EPSG:6.0:32611">
         <gml:lowerCorner>-115.467123 16.743654</gml:lowerCorner>
         <gml:upperCorner>-88.291157 32.654688</gml:upperCorner>
      </gml:Envelope>
   </gml:boundedBy>
   <gml:featureMember>
      <Capital gml:id="AC">
         <cityName>Aguascalientes</cityName>
         <stateName>AGUASCALIENTES</stateName>
         <gml:pointProperty>
            <gml:Point srsName="urn:ogc:def:crs:EPSG:6.0:32611">
               <gml:pos>-102.28969800 21.88751600</gml:pos>
            </gml:Point>
         </gml:pointProperty>
      </Capital>
   </gml:featureMember>
</gml:FeatureCollection>
```

That XML document could be either bundled with the Transform output OperationResponse or accessed remotely. Again, this feature collection is XML encoded using an example GML 3.1.1 Application Schema, named exampleCapital.xsd and listed in Subclause F.2.

8.5 Grid coverage inputs and outputs

8.5.1 Introduction

The preceding subclauses discussed the Transform operation request and response primarily for digital features, largely ignoring grid coverages. The additional Transform operation aspects for grid coverages are now discussed.

8.5.2 Input and output coverage formats

A WCTS server shall input and output coverages in any of the formats identified as an input or output CoverageFormat in CoverageAbilities in the Contents section of the Capabilities document. For most possible formats, more information is required to specify how a coverage shall (or can) be encoded using that format. That specific information should be specified in a WCTS or WCS Application Profile document (see Subclause 9.3.2.2 of WCS 1.1, OGC 07-067r2).

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 colour values; hence a GIF converter will approximate colours while determining a best-match colour 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 Transform operation response shall be encoded as an OperationResponse XML document that references one or more files. Similarly, the Transform operation request shall be encoded using an InputData XML structure that references one or more files. In either case, the referenced files may be bundled with it or accessible separately via URL(s), as detailed in Subclause 8.3.4 above.

8.5.3 Input and output coverage metadata

The metadata listed in Table 14, or the equivalent information, shall be included with each WCTS input and output coverage. Such equivalent metadata may be encoded with the coverage in some of the possible coverage formats. When not encoded with the coverage data, this metadata shall be encoded using GML 3.1.1.

Table 14 — Required coverage metadata

Names	Definition	Data type	Multiplicity and use
transformation Transformation	Definition of georeferencing coordinate transformation	CC_Transformation or CC_ConcatenatedOperation	Zero or one (optional) Should include when coverage is not georectified but is georeferenced
gridCRS GridCRS	Definition of GridCRS used by coverage	GridCRS data structure, see Annex G of WCS 1.1 [OGC 07-067r2]	Zero or one (optional) Include when coverage is georectified
imageCRS ImageCRS	Association to ImageCRS of this coverage	Association to ImageCRS	Zero or one (optional) Include when coverage is an image
boundingBox BoundingBox	Minimum bounding box for this coverage [a]	BoundingBox data structure, see Subclause 10.2 in OWS Common [OGC 06-121r3]	One (mandatory)
a This bounding box shall use the ImageCRS or GridCRS of this coverage. Grid coverages use of this BoundingBox is further specified in Subclause 7.7 of WCS 1.1 [OGC 07-067r2].			

8.5.4 Output coverage CRS

Each output coverage from a Transform operation shall be in the GridCRS that is specified in the Transform operation request. This GridCRS specifies the desired grid for the output coverage(s), including the desired grid spacings, one of many possible origin locations, optional rotation, and optional skewing (angle between the axes) in a base CRS. However, this GridCRS is NOT required or expected to specify the actual position of the origin in the Transform operation output. The position of the origin in the Transform operation output shall be as directly or indirectly specified by the input coverage in the Transform operation request.

8.5.5 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 input coverage. The BoundingBox of this spatial extent shall be computed, in the CRS of that output coverage, from the BoundingBox of the input coverage. This extent may be skewed with curved edges in the desired output coverage CRS, although it is rectangular in the CRS of the input coverage. Such a skewed and curved extent in the desired resampled output coverage shall be interpreted as the minimum extent desired.

NOTE 1 Grid coverages use of BoundingBoxes is further specified in Subclause 7.7 of WCS 1.1 [OGC 07-067r2].

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

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 centre points in each side of that bounding box. The positions of these eight or more polygon vertices may be transformed from the input coverage GridCRS (not the baseCRS 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 of column of grid point positions.

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

8.5.6 SupportedFormat and SupportedCRS dependencies

Some coverage formats require that the coverage data be stored in a specific CRS, or the CRS must be chosen from a small list of CRSs. The dependency between CRSs and formats can not be expressed in this specification. When a WCTS server receives a request where the requested output coverage format and the requested output CRS are incompatible, the server shall return an UnsupportedCombination exception, see Table 12.

8.5.7 Output coverage range and temporal extent

The range and temporal extent of a Transform operation output coverage shall be the same as for that input coverage.

8.5.8 Image transformation examples

8.5.8.1 Transform request example

If a client wants to transform an image from OGC:XXXX to OGC:YYYY using the default transformation, it may issue the following XML encoded Transform operation request:

```
<?xml version="1.0" encoding="UTF-8"?>
<Transform xmlns="http://www.opengis.net/wcts/0.0"
xmlns:wcs="http://www.opengis.net/wcs/1.2"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcts/0.0
../../wctsTransform.xsd
http://www.opengis.net/wcs/1.2 ../../../../wcs/1.2.0/wcsAll.xsd
http://www.opengis.net/gml ../../../../wcs/1.2.0/gml4wcs.xsd"
service="WCTS" version="0.0.0" store="true">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28-->
   <Transformation xlink:href="http://TBD"/>
   <wcs:GridCRS gml:id="GridCRSabcd">
      <wcs:GridBaseCRS></wcs:GridBaseCRS>
      <wcs:GridOffsets></wcs:GridOffsets>
   </wcs:GridCRS>
   <ows:InputData>
      <ows:ReferenceGroup>
         <ows:Reference
xlink:href="http://baesystems.com/images/ABC123.pix"
xlink:role="wcsCoverage"/>
      </ows:ReferenceGroup>
   </ows:InputData>
   <InterpolationType>linear</InterpolationType>
</Transform>
```

The input image is referenced at the URL http://baesystems.com/images/ABC123.pix. The sourceCRS is the ImageCRS of that image, using a URN specified in Subclause 10.2 of [OGC 05-096r1]. The targetCRS is a GridCRS

NOTE The sourceCRS and targetCRS cannot be any CRS defined by the EPSG, because the EPSG has defined no CRSs that (completely) specify an image CRS. In addition, each image will normally have a different CRS.

8.5.8.2 Transform response example

The response to a request for transforming an image from the server could be, if no exception occurs:

```
<?xml version="1.0" encoding="UTF-8"?>
<OperationResponse xmlns="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/wcts/1.0
../../wctsTransform.xsd">
   <ReferenceGroup>
      <Reference xlink:href="http://baesystems.com/images/DEF456.pix"
xlink:role="Coverage"/>
   </ReferenceGroup>
</OperationResponse>
```

In this example, the output image is assumed to be referenced at http://baesystems.com/images/DEF456.pix.

9 GetCapabilities operation (mandatory)

9.1 Introduction

The mandatory GetCapabilities operation allows WCTS clients to retrieve service metadata from a server. The response to a GetCapabilities request shall be a XML document containing service metadata about the server, including information about its coordinate transformation abilities. This clause specifies the XML document that a WCTS server shall return to describe its capabilities.

9.2 Operation request

9.2.1 GetCapabilities request parameters

The GetCapabilities operation request shall be as specified in Subclauses 7.2.2 of [OGC 06-121r3]. The value of the "service" parameter shall be "WCTS". The allowed set of service metadata (or Capabilities) XML document section names and meanings shall be as specified in Table 3 in Subclause 7.3.3 of [OGC 06-121r3].

9.2.2 GetCapabilities request implementation requirements

The "Multiplicity and use" column in Table 1 of [OGC 06-121r3] specifies the optionality of each listed parameter in the GetCapabilities operation request. Table 15 specifies the implementation of those parameters by WCTS clients and servers.

Table 15 — Implementation of parameters in GetCapabilities operation request

Names	Multiplicity	WCTS server implementation	WCTS 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 If this parameter is not implemented or not received, server shall provide the specified default response. If it is implemented and received, the server shall provide the specified response.

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

9.2.3 GetCapabilities request KVP encoding (required)

All WCTS servers shall implement HTTP GET transfer of the GetCapabilities operation request, using KVP encoding as specified in Subclause 7.2.3 of [OGC 06-121r3].

EXAMPLE To request a WCTS capabilities document, a client could issue the following KVP encoded GetCapabilities operation request with near-minimum contents:

```
www.baesystems.com/webservice/igs&service=WCTS&request=getCapabilities
```

9.2.4 GetCapabilities request XML encoding (optional)

Servers may also implement HTTP POST transfer of the GetCapabilities operation request, using XML encoding only. The XML Schema fragment for encoding a WCTS GetCapabilities operation request extends the ows:GetCapabilitiesType in owsCommon.xsd of [OGC 06-121r3], and is the GetCapabilities element in the attached normative wctsGetCapabilities.xsd file. The GetCapabilities operation request examples are very similar to those listed in Subclauses 7.2.2 and 7.2.3 of [OGC 06-121r3].

9.2.5 GetTransformation request SOAP encoding (optional)

WCTS servers may implement SOAP version 1.2 transfer of the GetTransformation operation request as specified in Annex G, using the XML encoding referenced above.

9.3 Operation response

9.3.1 GetCapabilities operation exceptions

When a web coordinate transformation 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 5 of [OGC 06-121r3], if the updateSequence parameter is implemented by the server.

9.3.2 GetCapabilities normal response

The complete service metadata document shall contain the four sections specified in Table 16. Depending on the values in the Sections parameter of the GetCapabilities operation request, any combination of these sections can be requested and returned.

Table 16 — 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 7.4.3 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 Subclause 7.4.4 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 schema of this section shall be the same as for all OWSs, as specified in Subclause 7.4.5 and owsOperationsMetadata.xsd of [OGC 06-121r3].
Contents	Metadata about the data served by this server. For the WCTS, this section shall contain data about the coordinate transformations and CRSs known to this server, as specified in Subclauses 11.3.3 and 11.3.4 below.

NOTE The name Contents is slightly misleading because a WCTS server does not provide any feature or coverage data Contents, but provides processing abilities. However, a WCTS server does contain data defining the coordinate transformations and CRSs which it supports, which can be considered to be geospatial data.

These data structures are shown graphically in the UML class diagram in Figure 4. In addition to the sections listed above, each service metadata document shall include the mandatory “version” and optional updateSequence parameters specified in Table 6 in Subclause 7.4.1 of [OGC 06-121r3].

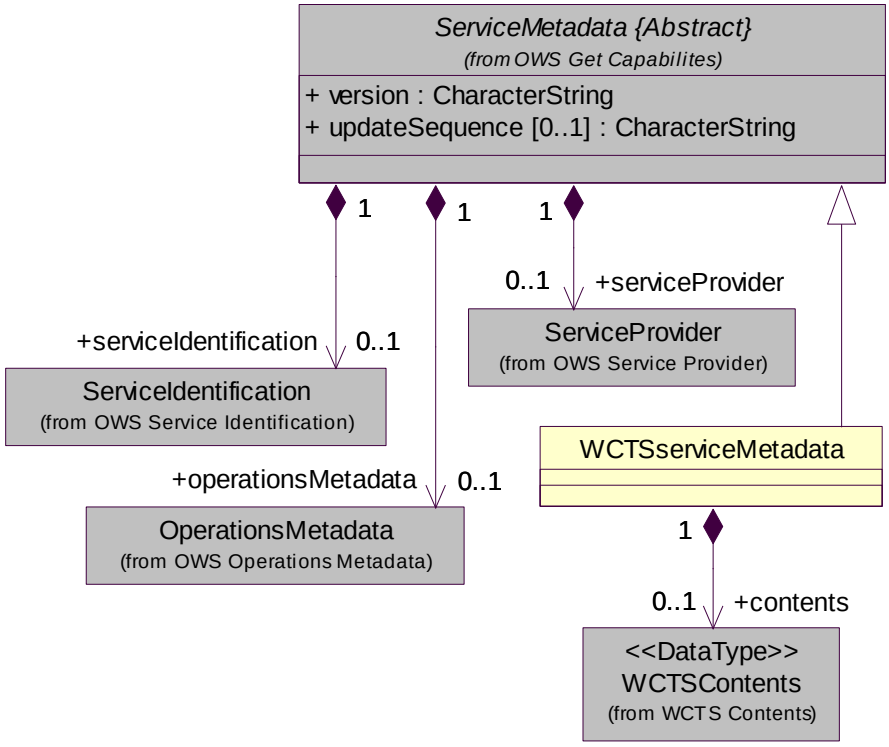

Figure 4 — WCTS service metadata UML class diagram

9.3.3 OperationsMetadata section standard contents

9.3.3.1 Overview

For the WCTS, the OperationsMetadata section structure shall be the same as for all OGC Web Services, as specified in Subclause 7.4.5 and owsOperationsMetadata.xsd of [OGC 06-121r3]. The mandatory values of various (XML) attributes shall be as specified in Table 17. Similarly, the items listed in Table 18 shall be included when these operations are implemented. In Table 17 and Table 18, the "Attribute name" column uses dot-separator notation to identify parts of a parent item.

Table 17 — Required values of OperationMetadata section attributes

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

Table 18 — Optional values of OperationMetadata section attributes

Attribute name	Attribute value	Meaning of attribute value
Operation.name	IsTransformable	The IsTransformable operation is implemented by this server.
	GetTransformation	The GetTransformation operation is implemented by this server.
	GetResourceByID	The GetResourceByID operation is implemented by this server.

In addition to the optional values listed in Table 18, there are many optional values of the "name" attributes and "value" elements in the OperationsMetadata section, which may be included when considered useful. Most of these attributes and elements are for recording the domains of various parameters and quantities.

EXAMPLE 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.

9.3.3.2 Indicating allowed encodings

All WCTS 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 G.

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

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 shall be included in the OperationsMetadata element.

9.3.3.3 Indicating support for the "store" parameter

All WCTS servers shall specify whether or not that server implements the "store" parameter in Transform operation requests, doing this in the ows:Operation element for the Transform operation. When this WCTS 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.

9.3.4 Contents section contents

The Contents section of a service metadata document contains metadata about the data served by this server. For the WCTS, this Contents section shall contain data about the coordinate transformations and CRSs known to this server, plus other abilities of this

server. The Contents section shall include the parameters and data structures shown graphically in the UML diagram in Figure 5 and specified in Table 19 through Table 22.

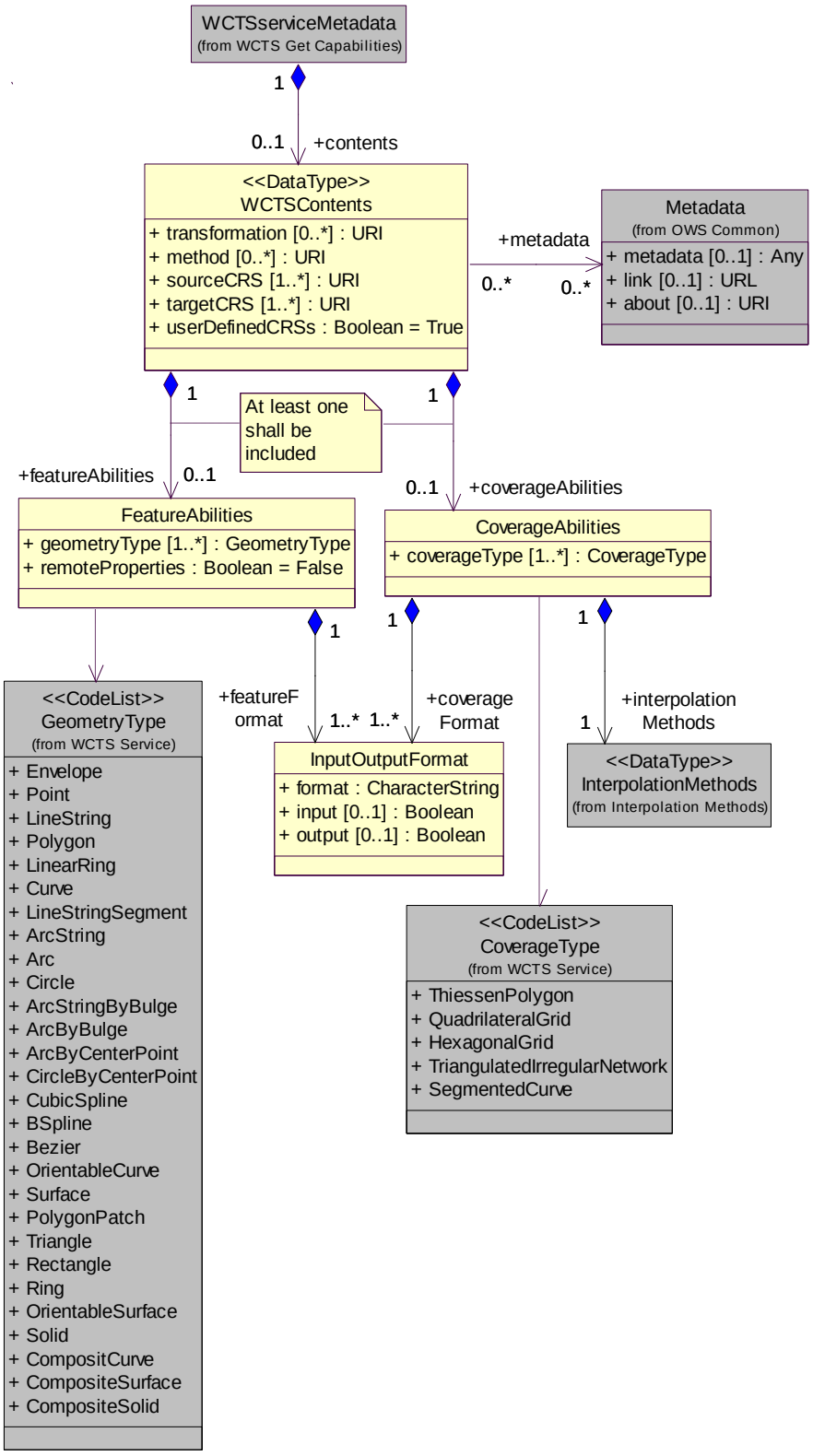

Figure 5 — WCTS contents section UML class diagram

Table 19 — Parameters included in Contents section

Names	Definition and data type	Multiplicity and use
transformation Transformation	Identifier of well-known coordinate operation which the server can perform URI [a]	Zero or more (optional) One for each coordinate operation implemented by this server [b]
method Method	Identifier of well-known operation method which the server can use in user-defined coordinate operations URI [a]	Zero or more (optional) One for each such operation method implemented by this server
sourceCRS SourceCRS	Reference to well-known CRS in which the server can accept sourceCRS values URI [a]	One or more (mandatory) One for each source CRS implemented by this server
targetCRS TargetCRS	Reference to well-known CRS in which the server can accept targetCRS values URI [a]	One or more (mandatory) One for each target CRS implemented by this server
featureAbilities FeatureAbilities	Information about the feature handling abilities of this WCTS server FeatureAbilities data structure, see Table 20	Zero or one (optional) Include when server can transform features
coverageAbilities CoverageAbilities	Information about the coverage handling abilities of this WCTS server CoverageAbilities data structure, see Table 21	Zero or one (optional) Include when server can transform coverages
metadata Metadata	Metadata about the data served by this WCTS implementation [c] Metadata contents or reference to metadata	Zero or more (optional) One for each such metadata object, unordered
userDefinedCRSs userDefinedCRSs	Specifies if this server supports user defined Coordinate Reference Systems (CRSs) Boolean type, true or false	One (mandatory)

a A URI shall reference a Transformation, Method, or CRS as specified in Subclause 10.3 of [OGC 06-121r3].

b The details of well-known transformations are not always available to clients. That is, only the sourceCRS and targetCRS information may be available to clients. Also, a WCTS server could support only user-defined transformations, and no well-known Transformations.

c For example, this metadata could include more detailed definitions of the CRSs, Transformations, and Methods known to this server, perhaps in the form of a gml:Dictionary of such information.

NOTE 1 This WCTS specification assumes that many implementations will list all of the coordinate operations they implement, as Transformations in the Contents section of a Capabilities document. One alternative is to not list all or most coordinate operations. In that case, Metadata parameters could be included in the Contents section to indicate the transformation abilities (loosely or precisely). If not, clients will need to discover which transformations are supported by trial and error (for example, using the IsTransformable and Transform operations).

NOTE 2 If highly useful Metadata is included or referenced in the Contents section, a WCTS profile should be written that specifies the contents of that Metadata.

EDITOR'S NOTE The Transformations (or coordinate operations) implemented by prototype WCTS servers (and listed in the Contents section) are currently limited to transformations in one direction only (not including the reverse direction). The team working on the WCTS specification has not yet considered what would be required to implement two-way transformations, as allowed in Topic 2 [OGC 05-103].

Any WCTS server has to understand at least two coordinate reference systems so that it is able to perform one coordinate transformation. The well-known transformations and coordinate reference systems are defined in the “transformation”s, “sourceCRS”s, and “targetCRS”s parameters.

The operation “method”s refer to general methods for transformation of coordinates from one CRS into another, and not to complete transformations between two particular reference systems. Only the methods that can be used in user-specified coordinate operations shall be listed. Examples of such methods are longitude rotation, abridged Molodenski, and geocentric to ellipsoid.

The FeatureAbilities data structure is specified in Table 20. The geometryType parameters identify the types of geometric primitives the WCTS is able to transform. Although GML 3 defines a wide spectrum of geometries, a WCTS is not required to support all those different kinds of geometries. Many WCTS servers will only need to support “simple” geometries as they were defined in GML 2. Other servers will not support all of the possible curve and surface interpolations available in GML 3 or are not interested in 3D geometries. The geometryType parameters allow the client to find out which GML 3 geometric primitives the WCTS supports. It is assumed that a WCTS has the ability to work with the corresponding geometric complexes and aggregates if used.

Table 20 — Parameters included in FeatureAbilities data structure

Names	Definition	Data type	Multiplicity and use
geometryType GeometryType	Name of GML geometric primitive type supported by this WCTS server [a]	GeometryType code list, see Table 4	One or more (mandatory) One for each primitive type supported by this server
featureFormat FeatureFormat	Identifier of feature format in which the Transform operation can input and output features	InputOutputFormat data structure, see Table 22	One or more (mandatory) One for each feature format supported by this server
remoteProperties remoteProperties	Specifies if this server supports remote properties in features transformed	Boolean type, true or false	One (mandatory)

a It is assumed that a WCTS server can also transform the corresponding geometric complexes and aggregates. It is also assumed that this server can transform at least one geometric primitive type.

The featureFormat parameters identify the formats that can be used for feature data input and/or output from this WCTS server. GML 3 allows the definition of remote properties that are referenced by Xlink components. When these remote properties contain geometries, the WCTS would have to follow the Xlink, transform the corresponding geometries and deliver those back to the client. This functionality might or might not be implemented by a WCTS server and is therefore specified by the Boolean remoteProperties parameter.

The CoverageAbilities data structure is specified in Table 21. The coverageType parameters identify the types of coverages the WCTS is able to transform. In the case of non-grid coverages which use specific geometric primitives, a client would have to check if the corresponding geometric primitive types and the coverage types are supported. The coverageFormat parameters identify the formats that can be used for coverage input

and/or output from this WCTS server. The InterpolationMethods parameters identify the interpolation methods the WCTS is able to use to transform coverages, in case an interpolation is necessary (such as for transformation of rectified grids).

Table 21 — Parameters included in CoverageAbilities data structure

Names	Definition	Data type	Multiplicity and use
coverageType CoverageType	Name of coverage type supported by this WCTS server	CoverageType code list, see Table 3	One or more (mandatory) One for each coverage type supported by this server
coverage Format Coverage Format	Identifier of coverage format in which Transform operation can input and/or output coverages	InputOutputFormat data structure, see Table 22	One or more (mandatory) One for each coverage format supported by this server
interpolation Methods Interpolation Methods	Identifiers of interpolation methods	InterpolationMethods data structure, see I.4 of OGC 07-067r2	One (mandatory)

Table 22 — Parameters included in InputOutputFormat data structure

Names	Definition	Data type	Multiplicity and use
format Format	Identifier of format type supported by this WCTS server	Character String type, not empty	One (mandatory)
input input	Defines if this format can be input to this server	Boolean type, default true	Zero or one (optional) When omitted, can be input by this server
output output	Defines if this format can be output from this server	Boolean type, default true	Zero or one (optional) When omitted, can be output by this server

EDITOR'S NOTE Questions were raised during OWS-2 about richer description of the format options available (tile, pyramid, tile size, compression rate…). Another issue is the lack of standard MIME type for geospatial grid coverage formats and guidelines to create new MIME types.

9.3.5 GetCapabilities response implementation requirements

The "Multiplicity and use" columns in Table 6 through Table 16 in [OGC 06-121r3], and in Table 19 through Table 22 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 OWS servers, using a specified value(s).

The "updateSequence" parameter defined in Table 6 of [OGC 05-008] is optional implementation by OWS servers. As indicated in Table 15 of this document, the "updateSequence" parameter may be implemented by each WCTS 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 OWS servers using specified values, whenever and wherever each is considered useful metadata for that server.

9.3.6 Capabilities document XML encoding (required)

A XML Schema fragment for a WCTS service metadata document extends the ows:CapabilitiesBaseType in owsGetCapabilities.xsd in [OGC 06-121r3], and is specified by the Capabilities element in the attached normative wctsGetCapabilities.xsd file. This fragment also uses the wctsContents.xsd attached normative schema document.

9.4 Examples

9.4.1 GetCapabilities request

To request a capabilities document, a client could issue the following GetCapabilities operation request with minimum contents encoded using KVP:

```
www.lat-lon.de/transform?service=WCTS&request=getCapabilities
```

The corresponding minimum request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities xmlns="http://www.opengis.net/wcts/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcts
../../wctsGetCapabilities.xsd"
service="WCTS"/>
<!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28 -->
```

A maximum request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities xmlns="http://www.opengis.net/wcts/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ows="http://www.opengis.net/ows/1.1"
xsi:schemaLocation="http://www.opengis.net/wcts/0.0
../../wctsGetCapabilities.xsd"
service="WCTS" updateSequence="XYZ123">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28 -->
   <ows:AcceptVersions>
      <ows:Version>0.0.0</ows:Version>
   </ows:AcceptVersions>
   <ows:Sections>
      <ows:Section>All</ows:Section>
   </ows:Sections>
</GetCapabilities>
```

9.4.2 GetCapabilities response

In response to such a request, the web coordinate transformation server might generate a XML document that looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities xmlns="http://www.opengis.net/wcts/0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
```



```
xmlns:wcs="http://www.opengis.net/wcs/1.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcts/0.0
../../wctsGetCapabilities.xsd
http://www.opengis.net/wcs/1.2 ../../../../wcs/1.2.0/wcsAll.xsd"
version="0.0.0">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28-->
   <ows:ServiceIdentification>
      <ows:Title>Web Coordinate Transformation Service</ows:Title>
      <ows:Abstract>Network service for transforming coordinates from
one CRS to another</ows:Abstract>
      <ows:Keywords>
         <ows:Keyword>Coordinate Reference System</ows:Keyword>
         <ows:Keyword>transformation</ows:Keyword>
         <ows:Keyword>conversion</ows:Keyword>
         <ows:Keyword>coordinate operation</ows:Keyword>
      </ows:Keywords>
      <ows:ServiceType>OGC WCTS</ows:ServiceType>
      <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
      <ows:Fees>NONE</ows:Fees>
      <ows:AccessConstraints>NONE</ows:AccessConstraints>
   </ows:ServiceIdentification>
   <ows:ServiceProvider>
      <ows:ProviderName>lat/lon</ows:ProviderName>
      <ows:ProviderSite xlink:href="TBD"/>
      <ows:ServiceContact>
         <ows:IndividualName>Andreas Poth</ows:IndividualName>
         <ows:PositionName>Senior Software Engineer</ows:PositionName>
         <ows:ContactInfo>
            <ows:Phone>
               <ows:Voice>++49 228 732838</ows:Voice>
               <ows:Facsimile>++49 228 732153</ows:Facsimile>
            </ows:Phone>
            <ows:Address>
               <ows:DeliveryPoint>Meckenheimer Allee
176</ows:DeliveryPoint>
               <ows:City>Bonn</ows:City>
               <ows:AdministrativeArea>NRW</ows:AdministrativeArea>
               <ows:PostalCode>53115</ows:PostalCode>
               <ows:Country>Germany</ows:Country>
               <ows:ElectronicMailAddress>poth@lat-
lon.de</ows:ElectronicMailAddress>
            </ows:Address>
         </ows:ContactInfo>
      </ows:ServiceContact>
   </ows:ServiceProvider>
   <ows:OperationsMetadata>
      <ows:Operation name="GetCapabilities">
         <ows:DCP>
            <ows:HTTP>
               <ows:Get xlink:href="www.lat-lon.de/transform"/>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
      <ows:Operation name="Transform">
         <ows:DCP>
            <ows:HTTP>
```



```
               <ows:Get xlink:href="www.lat-lon.de/transform"/>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
      <ows:Operation name="IsTransformable">
         <ows:DCP>
            <ows:HTTP>
               <ows:Get xlink:href="www.lat-lon.de/transform"/>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
      <ows:Operation name="DescribeTransformation">
         <ows:DCP>
            <ows:HTTP>
               <ows:Get xlink:href="www.lat-lon.de/transform"/>
            </ows:HTTP>
         </ows:DCP>
      </ows:Operation>
   </ows:OperationsMetadata>
   <Contents userDefinedCRSs="true">

   <Transformation>urn:ogc:def:coordinateOperation:EPSG::WWWW</Transfor
mation>

   <Transformation>urn:ogc:def:coordinateOperation:EPSG::XXXX</Transfor
mation>

   <Transformation>urn:ogc:def:coordinateOperation:EPSG::YYYY</Transfor
mation>

   <Transformation>urn:ogc:def:coordinateOperation:EPSG::ZZZZ</Transfor
mation>
      <Method>urn:ogc:def:method:EPSG::9801</Method>
      <Method>urn:ogc:def:method:EPSG::9802</Method>
      <Method>urn:ogc:def:method:EPSG::9803</Method>
      <SourceCRS>urn:ogc:def:crs:EPSG::4326</SourceCRS>
      <SourceCRS>urn:ogc:def:crs:EPSG::23031</SourceCRS>
      <SourceCRS>urn:ogc:def:crs:EPSG::23032</SourceCRS>
      <SourceCRS>urn:ogc:def:crs:EPSG::31467</SourceCRS>
      <TargetCRS>urn:ogc:def:crs:EPSG::4326</TargetCRS>
      <TargetCRS>urn:ogc:def:crs:EPSG::23031</TargetCRS>
      <TargetCRS>urn:ogc:def:crs:EPSG::23032</TargetCRS>
      <TargetCRS>urn:ogc:def:crs:EPSG::31467</TargetCRS>
      <CoverageAbilities>
         <CoverageType>GRID</CoverageType>
         <CoverageFormat>image/pix</CoverageFormat>
         <CoverageFormat>image/HDF-EOS</CoverageFormat>
         <CoverageFormat>image/DTED</CoverageFormat>
         <CoverageFormat>image/GeoTIFF</CoverageFormat>
         <CoverageFormat>image/NITF</CoverageFormat>
         <wcs:InterpolationMethods>
            <wcs:InterpolationMethod>nearest</wcs:InterpolationMethod>
            <wcs:InterpolationMethod>linear</wcs:InterpolationMethod>
         </wcs:InterpolationMethods>
      </CoverageAbilities>
      <FeatureAbilities remoteProperties="false">
         <GeometryType>Point</GeometryType>
         <GeometryType>LineString</GeometryType>
```



```
         <GeometryType>Polygon</GeometryType>
         <FeatureFormat>text/xml; gmlVersion=3.1.0</FeatureFormat>
      </FeatureAbilities>
   </Contents>
</Capabilities>
```

This service indicates it supports transformation of point/line/polygon features and grid coverages.

10 IsTransformable operation (optional)

10.1 Introduction

The IsTransformable operation allows WCTS clients to check if coordinate transformation of an identified set of geometry primitive types or coverage types is possible between two coordinate reference systems. This check will succeed only if the identified types are supported, and if there is a valid sequence of one or more known transformations to transform the coordinates from the source CRS to the target CRS. This check will NOT evaluate if this transformation makes any sense, nor if it is possible to be performed for each set of coordinates that may be submitted by a Transform request. Implementation of the IsTransformable operation is optional for a web coordinate transformation server.

NOTE In principle, this information could be extracted from the Capabilities document of the WCTS, since the parameters GeometryType, CoverageType, InterpolationType, Transformation, SourceCRS, and TargetCRS provide the necessary information. However, analysis of these parameters to check the possibility of a transformation between two CRSs, for some specific geometries or coverages, is in some cases rather complicated and would need powerful client logic. It seems to make more sense to provide this information directly via a distinct operation, allowing simple clients to use the service.

10.2 Operation request

10.2.1 IsTransformable request parameters

A request to perform the IsTransformable operation shall include the parameters and data structures shown graphically in the UML diagram in Figure 6, and specified in Table 23 and Table 24. These tables specify the meaning to servers when each optional parameter is not included in the operation request.

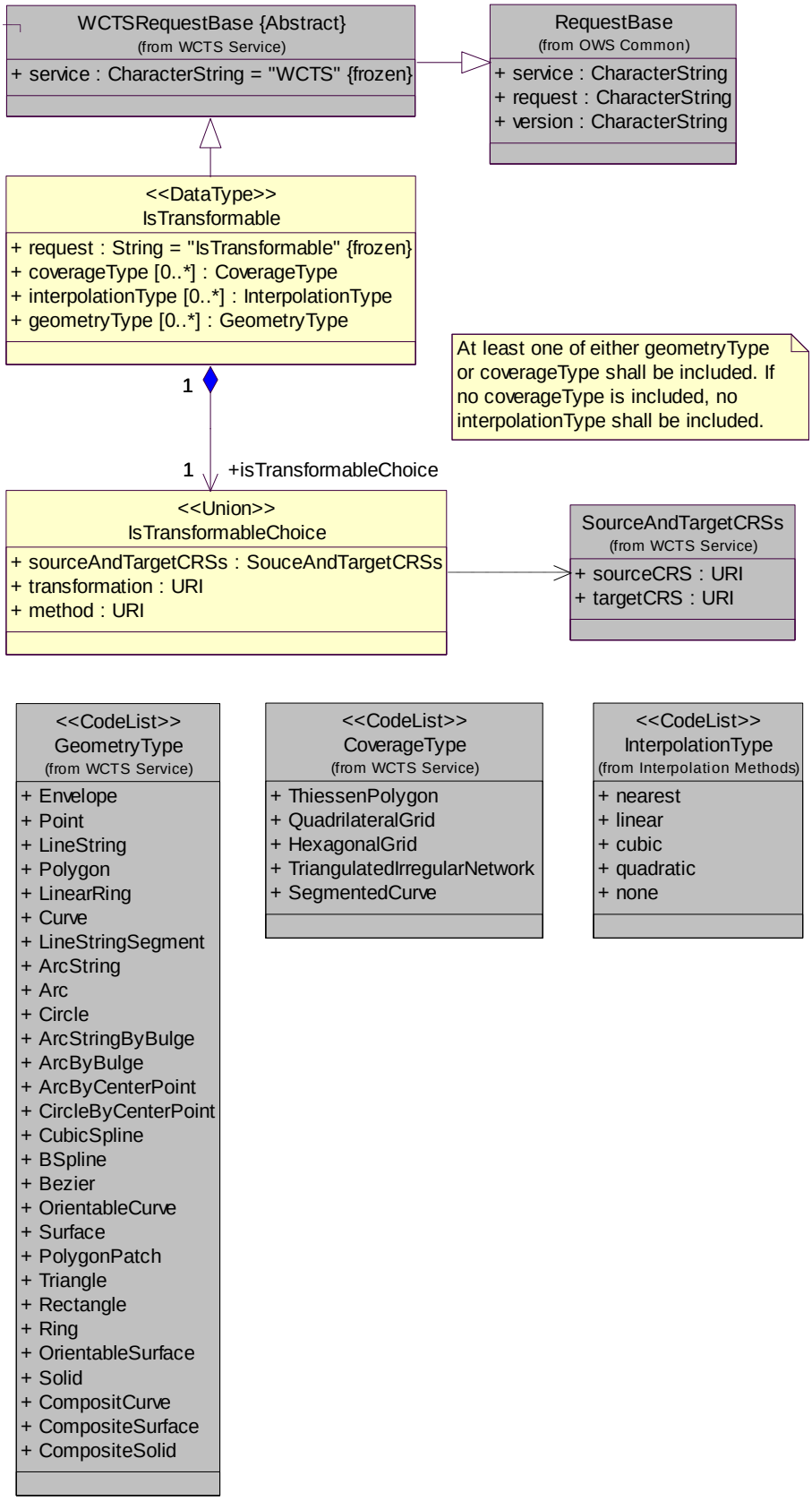

Figure 6 — WCTS IsTransformable request UML class diagram

NOTE 1 The first three parameters listed below are largely copied from Table 22 in Subclause 9.2.1 of [OGC 06-121r3]. The last parameter listed below is largely copied from Table I.6 in Subclause I.4.1 of WCS 1.1 [OGC 07-067r2].

Table 23 — Parameters in IsTransformable operation request

Names	Definition	Data type and value	Multiplicity and use
service service	Service type identifier	Character String, not empty Value is OWS type abbreviation, namely "WCTS"	One (mandatory)
request request	Operation name	Character String, not empty Value is operation name, namely "IsTransformable"	One (mandatory)
version version	Specification version for operation	Character String, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
isTransformable Choice (none)	Identification or definition of desired coordinate operation	IsTransformableChoice data structure, see Table 24	One (mandatory)
geometryType GeometryType	Identifier of type of GML 3 geometric primitive that will be requested to be transformed	GeometryType code list, see Table 4 Unordered list of geometric primitive types	Zero or more (Optional) [a] Included for transforming specific geometric primitive types
coverageType CoverageType	Identifier of coverage type that will be requested to be transformed	CoverageType code list, see Table 3 Unordered list of coverage type identifiers	Zero or more (Optional) [a] Included for transforming coverages
interpolation Type Interpolation Type	Identifier of interpolation method which should be used to transform coverage	Character string, not empty Values defined in service metadata, unordered list	Zero or more (Optional) [b] Included when transforming coverages

a Either at least one CoverageType or at least one GeometryType shall be included.

b One or more Interpolation Methods shall be included only if one or more Coverage Types are included.

NOTE 2 The data type of many parameters is specified as "Character String, not empty". In the XML Schemas specified herein, these parameters are encoded with the xsd:string type, which does NOT require that these strings not be empty.

Table 24 — Alternatives in IsTransformableChoice data structure

Names	Definition	Data type	Multiplicity
sourceAndTargetCRSs SourceAndTargetCRSs	References to SourceCRS and desired TargetCRS	SourceAnd TargetCRSs data structure, see Table 1	Zero or one (Mutually exclusive) [a]
transformation Transformation	Identification of desired coordinate operation	URI	Zero or one (Mutually exclusive) [b]
method Method	Identification of desired operation method	URI	Zero or one (Mutually exclusive) [c]

a Included when the server is permitted to select any coordinate operation which it knows about that can transform coordinates from the sourceCRS to the targetCRS identified.

b Included when client is specifying a specific well-known coordinate operation, whose definition is known to the WCTS server

c Included when client is considering specifying a specific user-defined coordinate operation that uses a well-known operation method.

10.2.2 IsTransformable request implementation requirements

The "Multiplicity and use" columns in Table 23 and Table 24 specify the optionality of each listed parameter in the IsTransformable operation request. Table 25 specifies the implementation of those parameters by servers and clients that implement the IsTransformable operation.

Table 25 — Implementation of parameters in IsTransformable operation request

Name	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 allowed values
request	One (mandatory)		
version	One (mandatory)		
sourceAnd TargetCRSs	One (Mutually exclusive)	One or more shall be implemented by all servers [a]	One or more shall be implemented by all clients
transformation			
method			
coverageType	Zero or one (optional)	Each shall be implemented by all servers that can transform coverages [a]	Each should be implemented by all clients that can transform coverages
interpolation Type	Zero or one (optional)		
geometryType	Zero or more (optional)	Shall be implemented by all servers that can transform features or geometries [a]	Should be implemented by clients that can transform features or geometries [b]

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

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

10.2.3 IsTransformable request KVP encoding (optional)

WCTS servers that implement the IsTransformable operation may implement HTTP GET transfer of the IsTransformable operation request, using KVP encoding. The KVP

encoding of the IsTransformable operation request shall use the parameters specified in Table 26. The parameters listed in Table 26 shall be as specified in Table 23 and Table 24above.

Table 26 — IsTransformable operation request URL parameters

Name and example [a]	Optionality and use	Definition and format
service=WCTS	Mandatory	Service type identifier
request= IsTransformable	Mandatory	Operation name
version=0.0.0	Mandatory	Specification and schema version for this operation
sourceCRS=urn:ogc:def:crs:EPSG:6.3:4326	Optional [b]	Identifier URI of input coordinate reference system
targetCRS=urn:ogc:def:crs:EPSG:6.3:23032	Optional [b]	Identifier URI of desired output coordinate reference system
transformation=urn:ogc:def:coordinatOperation:EPSG:6.3:TBD	Optional [b]	Identifier URI of desired coordinate operation
method=urn:ogc:def:method:EPSG:6.3:TBD	Optional [b]	Identifier URI of operation method to be used in user-defined coordinate transformation
geometryTypes=TBD	Optional Included for transforming specific geometric primitive types	Unordered list of types of GML 3 geometric primitives that will be requested to be transformed, separated by commas
coverageTypes=TBD	Optional Included for transforming coverages	Unordered list of coverage types that will be requested to be transformed, separated by commas
interpolationTypes=linear	Optional Included for transforming coverages	Unordered list of interpolation methods which could be used to transform coverages, separated by commas

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 Either Transformation, Method, or both SourceCRS and TargetCRS shall be included.

10.2.4 IsTransformable request XML encoding (required)

All WCTS servers that implement the IsTransformable operation shall implement HTTP POST transfer of the IsTransformable operation request, using XML encoding only. The XML Schema fragment for the IsTransformable operation is specified by the IsTransformable element in the attached normative wctsIsTransformable.xsd file. As indicated, that XML Schema fragment uses the Grid CRSs profile of GML 3.1.1 [OGC 05-096r1].

10.2.5 IsTransformable request SOAP encoding (optional)

WCTS servers that implement the IsTransformable operation may implement SOAP version 1.2 transfer of the IsTransformable operation request as specified in Annex G, using the XML encoding referenced above.

10.3 Operation response

10.3.1 IsTransformable operation exceptions

When a web coordinate transformation server encounters an error while performing an IsTransformable operation, it shall return an exception report message as specified in Subclause 8.5 of [OGC 06-121r3]. The allowed standard exception codes shall include those listed in Table 3 in Subclause 7.3 of this document.

EDITOR'S NOTE Do any more exceptionCode values need to be defined, for any other problem(s) that may prevent completing the IsTransformable operation?

10.3.2 IsTransformable normal response

The normal response to an IsTransformable request shall be "true" or "false" depending on whether the WCTS server can perform the requested transformation. When the response is "false", this response shall include an indication of the problem. More completely, a response from the IsTransformable operation shall include the data structures shown graphically in the UML diagram in Figure 7, and specified in Table 27 and Table 28.

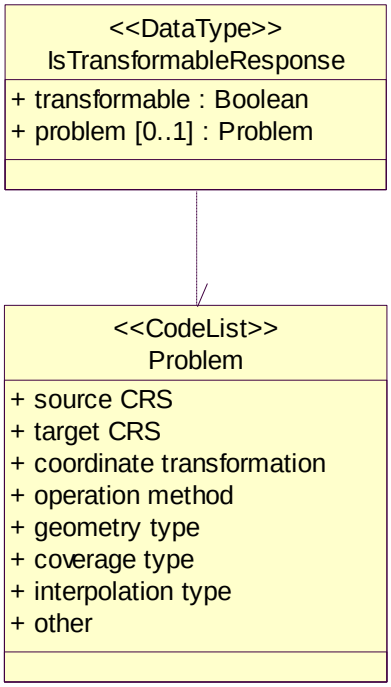

Figure 7 — WCTS IsTransformable response UML class diagram

Table 27 — Parameters in IsTransformable operation response

Names	Definition	Data type and value	Multiplicity and use
transformable transformable	Indicates whether this server can perform the transformation indicated in operation request [a]	Boolean type Value shall be "true" or "false"	One (mandatory)
problem problem	Type of transformation problem detected by WCTS server	Problem code list, see Table 28	Zero or one (optional) Included when "transformable" is false

a The indicated transformation is from the sourceCRS to the targetCRS identified by the input sourceAnd TargetCRSs when included in the operation request. Alternately, the indicated transformation is identified by the input "transformation" when included, or the "method" when included.

Table 28 — Problem values for IsTransformable operation response

"problem" value	Meaning
source CRS	WCTS server cannot transform from identified source CRS.
target CRS	WCTS server cannot transform to identified target CRS from identified source CRS.
geometry type	WCTS server cannot transform one or more identified geometry types.
coverage type	WCTS server cannot transform one or more identified coverage types.
interpolation method	WCTS server cannot perform one or more identified interpolation methods.
other	WCTS server cannot perform identified transformation due to some other problem, including incompatibility between identified parameters.

10.3.3 IsTransformable response XML encoding (required)

The XML Schema fragment for the IsTransformable operation response is specified by the IsTransformableResponse element in the attached normative wctsIsTransformable.xsd file. The Problem UML class is <CodeList> stereotyped, which is thus converted into XML Schema following the GML pattern for XML encoding of <CodeList> classes. That is, in wctsCommon.xsd, the normal values of Problem are NOT encoded in the XML Schema, but in the default-referenced GML Dictionaries named problemType.xml, and also attached to this document. This approach is used since not all useful values of this quantity are currently known, and this approach allows additional or different values to be more easily specified and used.

10.3.4 IsTransformable response SOAP encoding (optional)

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

10.4 Examples

10.4.1 IsTransformable request

An IsTransformable operation request for checking whether a transformation from EPSG:4326 to EPSG:23032 is possible may look like this when encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<IsTransformable xmlns="http://www.opengis.net/wcts/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcts
../../wctsIsTransformable.xsd"
service="WCTS" version="0.0.0">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28-->
   <SourceCRS>urn:ogc:def:crs:EPSG:6.0:4326</SourceCRS>
   <TargetCRS>urn:ogc:def:crs:EPSG:6.0:23032</TargetCRS>
   <GeometryType>LineString</GeometryType>
</IsTransformable>
```

The corresponding IsTransformable operation request encoded in KVP is:

```
www.latlon.de/transform&service=WCTS&request=isTransformable&version=0.
0.20&sourceCRS=urn:ogc:def:crs:EPSG:6.3:4326&targetCRS=urn:ogc:def:crs:
EPSG:6.3:23032&
```

10.4.2 IsTransformable response

Example “true” and “false” responses to the IsTransformable operation request are:

```
<?xml version="1.0" encoding="UTF-8"?>
<IsTransformableResponse xmlns="http://www.opengis.net/wcts/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcts/0.0
../../wctsIsTransformable.xsd"
transformable="true"/>
<!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28 -->

<?xml version="1.0" encoding="UTF-8"?>
<IsTransformableResponse xmlns="http://www.opengis.net/wcts/0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=http://www.opengis.net/wcts/0.0
../../wctsIsTransformable.xsd
transformable ="false">
<!-- Primary editor: Arliss Whiteside. Last updated 2007-05-01 -->
   <problem>geometryType</problem>
</IsTransformableResponse>
```

11 GetTransformation operation (optional)

11.1 Introduction

The GetTransformation operation allows clients to retrieve the definition of the transformation(s) from one coordinate reference system (CRS) into another. The definition(s) in the operation response are encoded in XML using GML 3.1.1.

Implementation of the GetTransformation operation is optional for a web coordinate transformation server.

11.2 Operation request

11.2.1 GetTransformation request parameters

A request to perform the GetTransformation operation shall include the parameters and data structures shown graphically in the UML diagram in Figure 8, and specified in Table 29.

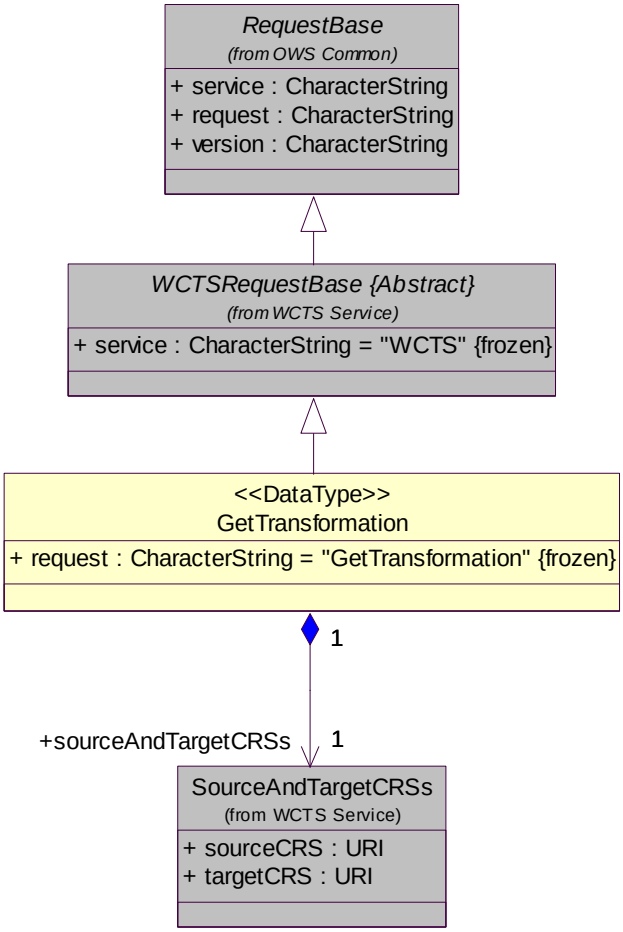

Figure 8 — WCTS GetTransformation request UML class diagram

NOTE The first three parameters listed below are largely copied from Table 22 in Subclause 9.2.1 of [OGC 06-121r3]. The other parameter listed below is defined in Table 1 in Subclause 7.2 of this document.

Table 29 — Parameters in GetTransformation operation request

Names	Definition	Data type and value	Multiplicity
service service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, namely "WCTS"	One (mandatory)
request request	Operation name	Character String type, not empty Value is operation name, namely "GetTransformation"	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)
sourceAndTargetCRSs (none)	References to SourceCRS and desired TargetCRS	SourceAndTargetCRSs data structure, see Table 1	One (mandatory)

11.2.2 GetTransformation request implementation requirements

The "Multiplicity and use" column in Table 29 specifies the optionality of each listed parameter and data structure in the GetTransformation operation request. Since all parameters and data structures are mandatory in the operation request, all parameters and data structures shall be implemented by all WCTS clients, using specified value(s). Similarly, all parameters and data structures shall be implemented by all WCTS servers, checking that each request parameter is received with allowed value(s).

11.2.3 GetTransformation request KVP encoding (required)

All WCTS servers that implement the GetTransformation operation may implement HTTP GET transfer of the GetTransformation operation request, using KVP encoding. The KVP encoding of the GetTransformation operation request shall use the parameters specified in Table 30. The parameters listed in Table 30 shall be as specified in Table 29 above.

Table 30 — GetTransformation operation request URL parameters

Name and example [a]	Optionality	Definition and format
service=WCTS	Mandatory	Service type identifier
request=GetTransformation	Mandatory	Operation name
version=0.0.20	Mandatory	Specification and schema version for this operation
sourceCRS=urn:ogc:def:crs:EPSG:6.3:4277	Mandatory	Identifier URI of input coordinate reference system
targetCRS=urn:ogc:def:crs:EPSG:6.3:2770	Mandatory	Identifier URI of desired output coordinate reference system

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].

11.2.4 GetTransformation request XML encoding (optional)

WCTS servers that implement the GetTransformation operation may implement HTTP POST transfer of the GetTransformation operation request, using XML encoding only. The XML Schema fragment for the GetTransformation operation request is specified by the GetTransformation element in the attached normative wctsGetTransformation.xsd file. As indicated, this XML Schema fragment uses the coordinate reference systems portion of GML 3.1.1 [OGC 04-092r4].

11.2.5 GetTransformation request SOAP encoding (optional)

WCTS servers may implement SOAP version 1.2 transfer of the GetTransformation operation request as specified in Annex G, using the XML encoding referenced above.

11.3 Operation response

11.3.1 GetTransformation operation exceptions

When a web coordinate transformation server encounters an error while performing a GetTransformation operation, it shall return an exception report message as specified in Subclause 8.5 of [OGC 06-121r3]. The allowed exception codes shall include those listed in Table 3 in Subclause 7.3 of this document.

EDITOR'S NOTE Do any more exceptionCode values need to be defined, for any other problem(s) that may prevent completing the GetTransformation operation?

11.3.2 GetTransformation response XML encoding (required)

The normal response to a GetTransformation operation request shall be the definitions of one or more coordinate transformations. Each of these transformations shall be encoded in XML using a concrete element in the gml:_CoordinateOperation substitutionGroup, specified in coordinateOperations.xsd of GML 3.1.1. Those transformation definitions shall always be encoded in one gml:Dictionary element as specified in the GML 3.1.1 simple dictionary profile [OGC 05-099r2].

It is possible that a transformation is not feasible. Possible reasons are that the web coordinate transformation server does not know the necessary transformation steps or that a transformation is not possible at all (for example, the transformation of a 2D to a 3D CRS). If a transformation is not feasible for a service, the WCTS shall return an exception report message.

11.3.3 GetTransformation response SOAP encoding (optional)

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

11.4 Examples

11.4.1 GetTransformation request

A GetTransformation operation request might be encoded in KVP like this:


```
www.lat-lon.de/transform&service=WCTS&request=GetTransformation
&version=0.0.0&sourceCRS=urn:ogc:def:crs:EPSG:6.3:4277&targetCRS=urn:og
c:def:crs:EPSG:6.3:2770&
```

The corresponding GetTransformation operation request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetTransformation xmlns="http://www.opengis.net/wcts/0.0"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcts/0.0
../../wctsGetTransformation.xsd"
service="WCTS" version="0.0.0">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28-->
   <SourceCRS>urn:ogc:def:crs:EPSG:6.0:4277</SourceCRS>
   <TargetCRS>urn:ogc:def:crs:EPSG:6.0:27700</TargetCRS>
</GetTransformation>
```

11.4.2 GetTransformation response

If no exception occurs, the server will respond to a GetTransformation request with a list of one or more transformations. To transform EPSG:4277 to EPSG:2770 just one transformation might be known:

```
<?xml version="1.0" encoding="UTF-8"?>
<Dictionary xmlns="http://www.opengis.net/gml"
xmlns:gml="http://www.opengis.net/gml"
xmlns:wcts="http://www.opengis.net/wcts/0.0"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=http://www.opengis.net/gml
../../../../gml/3.1.1/base/gml.xsd http://www.opengis.net/wcts/0.0
../../wctsOperationParameter.xsd
gml:id="root">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28 -->
   <!-- ==================================================== -->
   <name>GetTransformationOperationResponse</name>
   <!-- ==================================================== -->
   <dictionaryEntry>
      <!-- This XML example is for a specific coordinate operation that
uses the well-known Transverse Mercator operation method as defined by
the EPSG.
   sourceCRS: urn:ogc:def:srs:EPSG:6.0:4277 (OSGB 1936)
   targetCRS: urn:ogc:def:srs:EPSG:6.0:27700 (OSGB 1936 / British
National Grid) -->
      <Conversion gml:id="EPSG19916">
         <coordinateOperationName>Transverse
Mercator</coordinateOperationName>
         <coordinateOperationID>
            <name
codeSpace="urn:ogc:def:coordinateOperation:EPSG:6.0:">19916</name>
         </coordinateOperationID>
         <usesMethod xlink:href="urn:ogc:def:method:EPSG:6.0:9807"
xlink:title="Transverse Mercator"/>
         <usesValue>
            <value uom="urn:ogc:def:uom:OGC:1.0:degree">49</value>
```



```
            <valueOfParameter
xlink:href="urn:ogc:parameter:EPSG:6.0:8801" xlink:title="Latitude of
natural origin"/>
         </usesValue>
         <usesValue>
            <value uom="urn:ogc:def:uom:OGC:1.0:degree">-2</value>
            <valueOfParameter
xlink:href="urn:ogc:parameter:EPSG:6.0:8802" xlink:title="Longitude of
natural origin"/>
         </usesValue>
         <usesValue>
            <value
uom="urn:ogc:def:uom:OGC:1.0:unity">0.999601272</value>
            <valueOfParameter
xlink:href="urn:ogc:parameter:EPSG:6.0:8805" xlink:title="Scale factor
at natural origin"/>
         </usesValue>
         <usesValue>
            <value uom="urn:ogc:def:uom:OGC:1.0:metre">400000</value>
            <valueOfParameter
xlink:href="urn:ogc:parameter:EPSG:6.0:8806" xlink:title="False
Easting"/>
         </usesValue>
         <usesValue>
            <value uom="urn:ogc:def:uom:OGC:1.0:metre">-100000</value>
            <valueOfParameter
xlink:href="urn:ogc:parameter:EPSG:6.0:8807" xlink:title="False
Northing"/>
         </usesValue>
      </Conversion>
   </dictionaryEntry>
</Dictionary>
```

12 GetResourceByID operation (optional)

12.1 Introduction

The GetResourceByID operation allows WCTS clients to retrieve definitions of one or more identified abilities provided by the WCTS server. Those resources can be coordinate transformations, coordinate reference systems (CRSs), or operation methods. All definitions in the operation response are encoded in XML using GML 3.1.1. Implementation of the GetResourceByID operation is optional for a WCTS server. This WCTS GetResourceByID operation is adapted from the GetResourceByID operation specified in Subclause 9.4 of OWS Common [OGC 06-121r3].

12.2 Operation request

12.2.1 GetResourceByID request parameters

A request to perform the GetResourceByID operation shall include the parameters listed and defined in Table 31. This table also specifies the UML model data type, source of values, and multiplicity of each listed parameter.

NOTE All four parameters listed below (with grey background) are largely copied from Table 23 in Subclause 9.2.1 of [OGC 06-121r3].

Table 31 — Parameters in GetResourceByID operation request

Name	Definition	Data type and value	Multiplicity and use
service service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, namely "WCTS"	One (mandatory)
request request	Operation name	Character String type, not empty Value is operation name, namely "GetResourceByID"	One (mandatory)
version version	Specification version for operation	Character String type, not empty Value is specified by each specification and Schemas version	One (mandatory)
resourceID ResourceID	Unambiguous identifier of desired object, identifying specific version when needed	URI Values defined in service metadata (Capabilities) or in other metadata known to client	One or more (optional) Include one for each object requested
a If there are multiple versions of the same basic resource, each version shall have a different ResourceID. The version may be identified within that ResourceID, but this version shall be opaque to OWS servers.			

12.2.2 ResourceID values

For the WCTS, three categories of ResourceID may be allowed, each category being optional implementation by a server. Only one category of resource can be included in one GetResourceByID operation request. These three categories are:

a) Transformation – Each of these transformations can be a single transformation, or a ConcatenatedOperation that combines multiple single transformations.

b) CRS – Each of these CRSs can be any one of the six different concrete types of CRS included in the GML 3.1.1 Grid CRSs profile [OGC 05-096r1].

c) Method – Each operation method can include zero or more operation parameters.

12.2.3 GetResourceByID request implementation requirements

The "Multiplicity and use" column in Table 31 specifies the optionality of each listed parameter in the GetResourceByID operation request. Since all parameters are mandatory in the operation request, all parameters shall be implemented by all WCTS clients, using a specified value(s). Similarly, all parameters shall be implemented by all WCTS servers, checking that each request parameter is received with allowed value(s).

12.2.4 GetResourceByID request KVP encoding (optional)

A server that implements the GetResourceByID operation may implement HTTP GET transfer of the operation request, using KVP encoding. The KVP encoding of the GetResourceByID operation request shall use the parameters specified in Table 32, which shall be as specified in Table 31 above.

Table 32 — GetResourceByID operation request URL parameters

Name and example [a]	Optionality	Definition and format
service=WCTS	Mandatory	Service or profile type identifier
request=GetResourceByID	Mandatory	Operation name
version=0.0.0	Mandatory	Specification and schema version for this operation
ResourceIDs=urn:ogc:def:coordinateOperation:EPSG:6.3:19916	Mandatory	Identifier URIs of one or more objects, comma-separated list

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].

12.2.5 GetResourceByID request XML encoding (optional)

A server that implements the GetResourceByID operation may implement HTTP POST transfer of the operation request, using XML encoding only. The XML Schema fragment for the GetResourceByID operation request is specified by the GetResourceByID element in the attached normative wctsGetResourceByID.xsd file.

12.2.6 GetResourceByID request SOAP encoding (optional)

WCTS servers may implement SOAP version 1.2 transfer of the GetResourceByID operation request as specified in Annex G, using the XML encoding referenced above.

12.3 Operation response

12.3.1 GetResourceByID operation exceptions

When a web coordinate transformation server encounters an error while performing an GetResourceByID operation, it shall return an exception report message as specified in Subclause 8.5 of [OGC 06-121r3]. The allowed standard exception codes shall include those listed in Table 3 in Subclause 7.3 of this document.

12.3.2 GetResourceByID normal response

The normal response to a GetResourceByID operation request shall be the definitions of the one or more requested objects. Those definitions shall always be encoded in one Dictionary element as specified in dictionary.xsd of the GML 3.1.1 simple dictionary profile [OGC 05-099r2]. For the three allowed categories of objects, the definitions in the dictionary shall be:

a) Transformation – Transformation or ConcatenatedOperation elements encoded as specified in coordinateOperations.xsd of the GML 3.1.1 Grid CRSs profile [OGC 05-096r1].

b) CRS – GeographicCRS, ProjectedCRS, VerticalCRS, CompoundCRS, ImageCRS, or DerivedCRS elements encoded as specified in coordinateReferenceSystems.xsd of the GML 3.1.1 Grid CRSs profile [OGC 05-096r1].

c) Method – Operation Method elements encoded as specified in coordinateOperations.xsd of the GML 3.1.1 Grid CRSs profile [OGC 05-096r1].

12.3.3 GML Application Schema

The definition of a Method should use a GML Application Schema developed for use by the WCTS. This GML Application Schema extends the definition of the gml:OperationParameter element to allow providing more information defining each operation parameter. This additional information includes the allowed parameter values, plus the supported Units of Measure for the values. This information is expected to be useful by a client that chooses to specify a user-defined coordinate transformation, using an operation method supported by a WCTS server.

This GML Application Schema is the attached normative XML Schema Document named wctsOperationParameter.xsd. This schema adds five XML elements to the definition of an OperationParameter, namely:

a) One of the alternative elements in the PossibleValues group, specifying the allowed values or range(s) of this parameter.

b) DefaultValue element, which should be included to specify the default value whenever a default value exists for this parameter.

c) Meaning element, which should be included to provide the meaning of this parameter.

d) DataType element, which should be included to identify the data type of this parameter.

e) ValuesUnit element, which should be included to identify the units of measure or reference system for values of this parameter whenever that parameter has units or a reference system.

f) otherUOM elements, which should be included to any other supported units of measure for values of this parameter.

NOTE This GML Application Schema uses the owsDomainType.xsd XML Schema document specified in OWS Common [OGC 06-121r3]

12.3.4 GetResourceByID response SOAP encoding (optional)

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

12.3.5 GetResourceByID response implementation requirements

TBD

12.4 Examples

12.4.1 GetResourceByID request for a Transformation

A GetResourceByID operation request for one Transformation might be encoded in KVP like this:


```
www.lat-
lon.de/transform&service=WCTS&request=GetResourceByID&version=0.0.
0&ResourceIDs=urn:ogc:def:coordinateOperation:EPSG:6.3:19916&
```

The corresponding operation request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetResourceByID xmlns="http://www.opengis.net/wcts/0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcts/0.0
../../wctsGetResourceByID.xsd"
service="WCTS" version="0.0.0">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28 -->
   <ows:ResourceID>urn:ogc:def:coordinateOperation:EPSG:6.3:19916</ows:
ResourceID>
</GetResourceByID>
```

12.4.2 GetResourceByID response for a Transformation

If no exception occurs, the server will reply to a GetResourceByID operation request for one or more Transformation with a gml:Dictionary containing one or more transformations. If the one EPSG 19916 transformation is requested, the response might be:

```
<?xml version="1.0" encoding="UTF-8"?>
<Dictionary xmlns="http://www.opengis.net/gml"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=http://www.opengis.net/gml
../../../../gml/3.1.1/base/gml.xsd
gml:id="root">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28 -->
   <!-- ====================================================== -->
   <name>GetResourceByIdOperationResponse</name>
   <!-- ====================================================== -->
   <dictionaryEntry>
      <Conversion gml:id="EPSG19916">
         <coordinateOperationName>Transverse
Mercator</coordinateOperationName>
         <coordinateOperationID>
            <name
codeSpace="urn:ogc:def:coordinateOperation:EPSG:6.0:">19916</name>
         </coordinateOperationID>
         <usesMethod xlink:href="urn:ogc:def:method:EPSG:6.0:9807"
xlink:title="Transverse Mercator"/>
         <usesValue>
            <value uom="urn:ogc:def:uom:OGC:1.0:degree">49</value>
            <valueOfParameter
xlink:href="urn:ogc:def:parameter:EPSG:6.0:8801" xlink:title="Latitude
of natural origin"/>
         </usesValue>
         <usesValue>
            <value uom="urn:ogc:def:uom:OGC:1.0:degree">-2</value>
```



```
            <valueOfParameter
xlink:href="urn:ogc:def:parameter:EPSG:6.0:8802" xlink:title="Longitude
of natural origin"/>
         </usesValue>
         <usesValue>
            <value
uom="urn:ogc:def:uom:OGC:1.0:unity">0.999601272</value>
            <valueOfParameter
xlink:href="urn:ogc:def:parameter:EPSG:6.0:8805" xlink:title="Scale
factor at natural origin"/>
         </usesValue>
         <usesValue>
            <value uom="urn:ogc:def:uom:OGC:1.0:metre">400000</value>
            <valueOfParameter
xlink:href="urn:ogc:def:parameter:EPSG:6.0:8806" xlink:title="False
Easting"/>
         </usesValue>
         <usesValue>
            <value uom="urn:ogc:def:uom:OGC:1.0:metre">-100000</value>
            <valueOfParameter
xlink:href="urn:ogc:def:parameter:EPSG:6.0:8807" xlink:title="False
Northing"/>
         </usesValue>
      </Conversion>
   </dictionaryEntry>
</Dictionary>
```

12.4.3 GetResourceByID request for a CRS

A GetResourceByID operation request for the EPSG 4277 CRS can look like this encoded in KVP:

```
http://www.lat-
lon.de/transform&service=WCTS&request=GetResourceByID&version=0.0.0&Res
ourceIDs=urn:ogc:def:crs:EPSG:6.3:4277&
```

The corresponding operation request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetResourceByID xmlns="http://www.opengis.net/wcts/0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcts/0.0
../../wctsGetResourceByID.xsd"
service="WCTS" version="0.0.0">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28 -->
   <ows:ResourceID>urn:ogc:def:crs:EPSG:6.3:4277</ows:ResourceID>
</GetResourceByID>
```

12.4.4 GetResourceByID response for a CRS

If no exception occurs, the server will reply to a GetResourceByID operation request with a gml:Dictionary containing one or more CRS definitions. If the one EPSG 4277 CRS is requested, the response might be:

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



```
<Dictionary 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/base/gml.xsd
gml:id="root">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28 -->
   <!-- ====================================================== -->
   <name>GetResourceByIdOperationResponse</name>
   <!-- ====================================================== -->
   <dictionaryEntry>
      <GeographicCRS gml:id="EPSG4277">
         <srsName>OSGB 1936</srsName>
         <srsID>
            <name codeSpace="urn:ogc:def:crs:EPSG:6.0:">4277</name>
         </srsID>
         <validArea>
            <description>United Kingdom (UK) - Great Britain - England
Scotland Wales - onshore; Isle of Man. </description>
         </validArea>
         <usesEllipsoidalCS>
            <EllipsoidalCS gml:id="EPSG6402">
               <csName>ellipsoidal</csName>
               <csID>
                  <name
codeSpace="urn:ogc:def:ellipsoid:EPSG:6.0:">6402</name>
               </csID>
               <usesAxis>
                  <CoordinateSystemAxis gml:id="EPSG9901"
gml:uom="urn:ogc:uom:OGC:1.0:degree">
                     <name>Geodetic latitude</name>
                     <axisID>
                        <name
codeSpace="urn:ogc:def:axis:EPSG:6.0:">9901</name>
                     </axisID>
                     <axisAbbrev>Lat</axisAbbrev>
                     <axisDirection
codeSpace="urn:ogc:def:axisDirection:OGC:1.0:">north</axisDirection>
                  </CoordinateSystemAxis>
               </usesAxis>
               <usesAxis>
                  <CoordinateSystemAxis gml:id="EPSG9902"
gml:uom="urn:ogc:uom:OGC:1.0:degree">
                     <name>Geodetic longitude</name>
                     <axisID>
                        <name
codeSpace="urn:ogc:def:axis:EPSG:6.0:">9902</name>
                     </axisID>
                     <axisAbbrev>Lon</axisAbbrev>
                     <axisDirection
codeSpace="urn:ogc:def:axisDirection:OGC:1.0:">east</axisDirection>
                  </CoordinateSystemAxis>
               </usesAxis>
            </EllipsoidalCS>
         </usesEllipsoidalCS>
         <usesGeodeticDatum>
            <GeodeticDatum gml:id="EPSG6277">
               <datumName>OSGB 1936</datumName>
```



```
               <datumID>
                  <name
codeSpace="urn:ogc:def:datum:EPSG:6.0:">6277</name>
               </datumID>
               <usesPrimeMeridian>
                  <PrimeMeridian gml:id="EPSG8901">
                     <meridianName>Greenwich</meridianName>
                     <meridianID>
                        <name
codeSpace="urn:ogc:def:meridain:EPSG:6.0:">8901</name>
                     </meridianID>
                     <greenwichLongitude>
                        <angle
uom="urn:ogc:uom:OGC:1.0:degree">0</angle>
                     </greenwichLongitude>
                  </PrimeMeridian>
               </usesPrimeMeridian>
               <usesEllipsoid>
                  <Ellipsoid gml:id="EPSG7001">
                     <ellipsoidName>Airy 1830</ellipsoidName>
                     <ellipsoidID>
                        <name
codeSpace="urn:ogc:def:ellipsoid:EPSG:6.0:">7001</name>
                     </ellipsoidID>
                     <semiMajorAxis
uom="urn:ogc:uom:OGC:1.0:metre">6377563.396</semiMajorAxis>
                     <secondDefiningParameter>
                        <inverseFlattening
uom="urn:ogc:uom:EPSG:6.0:9201">299.3249646</inverseFlattening>
                     </secondDefiningParameter>
                  </Ellipsoid>
               </usesEllipsoid>
            </GeodeticDatum>
         </usesGeodeticDatum>
      </GeographicCRS>
   </dictionaryEntry>
</Dictionary>
```

12.4.5 GetResourceByID request for a Method

A GetResourceByID operation request for the EPSG 9807 operation method (Transverse Mercator) can look like this encoded in KVP:

```
http://www.lat-
lon.de/transform&service=WCTS&request=GetResourceByID&version=0.0.0&Res
ourceIDs=urn:ogc:def:method:EPSG:6.3:9807&
```

The corresponding operation request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<GetResourceByID xmlns="http://www.opengis.net/wcts/0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcts/0.0
../../wctsGetResourceByID.xsd"
service="WCTS" version="0.0.0">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28 -->
```



```
   <ows:ResourceID>urn:ogc:def:method:EPSG:6.3:9807</ows:ResourceID>
</GetResourceByID>
```

12.4.6 GetResourceByID response for a Method

If no exception occurs, the server will reply to a GetResourceByID operation request with a gml:Dictionary containing one or more operation method definitions. If the one EPSG 9807 operation method is requested, the response might be:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetResourceByID xmlns="http://www.opengis.net/wcts/0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcts/0.0
../../wctsGetResourceByID.xsd"
service="WCTS" version="0.0.0">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28 -->
   <ows:ResourceID>urn:ogc:def:method:EPSG:6.3:9807</ows:ResourceID>
</GetResourceByID>
```

12.4.7 GetResourceByID response for an orthorectification method

If no exception occurs, the server will reply to a GetResourceByID operation request for methods with a gml:Dictionary containing of one or more operation method definitions. If one user defined orthorectification operation method is requested, the response might be:

EDITOR'S NOTE This example was showcased during OWS-2.

```
<?xml version="1.0" encoding="UTF-8"?>
<Dictionary xmlns="http://www.opengis.net/gml"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:wcts=http://www.opengis.net/wcts/0.0
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/gml
../../../../gml/3.1.1/base/gml.xsd http://www.opengis.net/wcts/0.0
../../wctsOperationParameter.xsd"
gml:id="root">
   <!-- Primary editor: Arliss Whiteside. Last updated 2007-04-28 -->
   <!-- ================================================== -->
   <name>GetResourceByIdOperationResponse</name>
   <!-- ================================================== -->
   <dictionaryEntry>
      <OperationMethod gml:id="_SPOT_SRIT">
         <methodName>SPOT SRIT (Orthorectification)</methodName>
         <methodID>
            <name
codeSpace="http://www.pcigeomatics.com/ontology/opengis/services/wcts/w
arper#">SPOT_SRIT</name>
            <version>1.0.0</version>
         </methodID>
         <remarks>PCI Geomatics' SRIT (rectification system for remote-
sensing imagery) mathematical model is applied in correcting SPOT
images. This rigorous method of geometric correction takes into account
all global system distortions (satellite, sensor, earth). This
```



```
implementation relies on DIMAP information to compute the positioning
of the imagery. It does not use Ground Control Points. It provides a
good accuracy of ~20-40 m even without Ground Control Points.
</remarks>
         <methodFormula>CCRS SRIT Model</methodFormula>
         <sourceDimensions>2</sourceDimensions>
         <targetDimensions>2</targetDimensions>
         <usesParameter>
            <wcts:OperationParameter gml:id="_bands">
               <parameterName>List of channels</parameterName>
               <parameterID>
                  <name
codeSpace="http://www.pcigeomatics.com/ontology/opengis/services/wcts/w
arper#">bands</name>
               </parameterID>
               <remarks>Comma delimited list of channel number (index
starting at 1)</remarks>
               <wcts:allowedValues>
                  <ows:AllowedValues>
                     <ows:Range>
                        <ows:MinimumValue>1</ows:MinimumValue>
                     </ows:Range>
                  </ows:AllowedValues>
               </wcts:allowedValues>
            </wcts:OperationParameter>
         </usesParameter>
         <usesParameter>
            <wcts:OperationParameter gml:id="_DIMAP_URL">
               <parameterName>DIMAP URL</parameterName>
               <parameterID>
                  <name
codeSpace="http://www.pcigeomatics.com/ontology/opengis/services/wcts/w
arper#">DIMAP_URL</name>
               </parameterID>
               <remarks>DIMAP Metadata URL for SPOT Imagery typically
metadata.dim file</remarks>
               <wcts:allowedValues>
                  <ows:AllowedValues>
                     <ows:Value>TBD</ows:Value>
                  </ows:AllowedValues>
               </wcts:allowedValues>
            </wcts:OperationParameter>
         </usesParameter>
         <usesParameter>
            <wcts:OperationParameter gml:id="_pixel-size">
               <parameterName>Pixel X, Y output size</parameterName>
               <parameterID>
                  <name
codeSpace="http://www.pcigeomatics.com/ontology/opengis/services/wcts/w
arper#">pixel-size</name>
               </parameterID>
               <remarks>pixel size in X and Y in either decimal degree
or meter. The pixel size are defined as comma-delimited value
pixelSizeX,pixelSizeY. This value may be moved to the WCTS
parameter</remarks>
               <wcts:allowedValues>
                  <ows:AllowedValues>
                     <ows:Range>
```



```
                        <ows:MinimumValue>0.001</ows:MinimumValue>
                     </ows:Range>
                  </ows:AllowedValues>
               </wcts:allowedValues>
            </wcts:OperationParameter>
         </usesParameter>
         <usesParameter>
            <wcts:OperationParameter gml:id="_DEM_URL">
               <parameterName>Elevation (DEM) datafile
URL</parameterName>
               <parameterID>
                  <name
codeSpace="http://www.pcigeomatics.com/ontology/opengis/services/wcts/w
arper#">DEM_URL</name>
               </parameterID>
               <remarks>Digital elevation model (DEM) url used for the
orthorectification process</remarks>
               <wcts:allowedValues>
                  <ows:AllowedValues>
                     <ows:Range/>
                  </ows:AllowedValues>
               </wcts:allowedValues>
            </wcts:OperationParameter>
         </usesParameter>
      </OperationMethod>
   </dictionaryEntry>
</Dictionary>
```

NOTE The orthorectified image resolution in each direction needs to be specified by the values of one or more of the Transformation parameters. In this example, the orthorectified image resolution is set by the value of a transformation parameter named pixel_size. This is probably not the optimum parameter name or meaning.

Annex A
(normative)

Abstract test suite

NOTE This annex is not yet an Abstract Test Suite as specified in ISO 19105, in either structure or content. This annex will thus need to be revised before this Implementation Specification can be submitted to ISO/TC 211.

A.1 Conformance testing overview

The first step in testing conformance of a server with the interface specified in this document shall be review of the server implementation documentation. This review shall determine the abilities that are implemented which need to be tested, and shall verify that the abilities implemented are sufficiently specified to allow testing. Documentation review is discussed in more detail in Subclause A.2.

Conformance testing of server software that implements this Implementation Specification is logically organised by the coordinate transformations which are implemented by that software. Only those coordinate transformations identified as being implemented by that implementation should be tested, except when testing error handling.

Ideally, all coordinate transformations identified as being implemented by an implementation should be tested. However, if more than 100 (TBR) different coordinate transformations are implemented, a diverse sample of 100 (TBR) of those transformations may be tested. That diverse sample should include each supported source CRS, target CRS, and operation method (including those used by coordinate Conversions without specific source and target CRSs).

For each implemented operation, a sequence of operation requests shall be made using that operation. These sequences of operation requests are outlined below in Subclauses A.3 through A.9. These sequences of operation requests shall be intermixed with other operation requests, involving the same and different coordinate transformations.

A.2 Documentation review

Conformance testing shall begin with a documentation review, to determine the abilities that are implemented and need to be tested. This documentation review shall also check if all the abilities implemented are sufficiently specified to support adequate testing. Specifically, this review shall check if the documentation clearly specifies the abilities implemented, including:

a) Which coordinate transformations it implements

b) Which operation methods it implements

c) Which coordinate reference systems it implements

d) Which interface operations it implements

e) Which data formats it implements

a) Which of the specified compliance classes it fully supports

The documentation reviewed shall include the complete Capabilities (service metadata) document that can be retrieved from the server, plus any other documentation available for that server (including the documentation referenced by that Capabilities document).

All of this information shall be used to select and define the conformance tests to be performed. The list of implemented coordinate transformations shall be used to select the set of transformations that should each be tested. The specification of each implemented coordinate transformation shall be used to determine the correct relationship of output data to the corresponding input data. Each implemented interface operation shall be tested.

A.3 GetCapabilities operation testing

The GetCapabilities operation shall be tested by using that operation to retrieve:

a) Only the ServiceIdentification section of the Capabilities document

b) Only the ServiceProvider section of the Capabilities document

c) Only the OperationsMetadata section of the Capabilities document

d) Only the Contents section of the Capabilities document

e) All sections of the Capabilities document

KVP, (plain) XML, and/or SOAP encoding of the GetCapabilities operation request shall be tested, depending on which encoding(s) are implemented. In addition to testing this operation with correct inputs, this operation shall be tested with incorrect inputs, including at least one incorrect value for each operation request parameter (TBR).

The Capabilities document including all sections shall be compared against the individual section Capabilities documents to verify that they are consistent. Each Capabilities document shall be checked to ensure that it is valid with respect to the XML Schema definition of that service metadata document.

Each section of the complete Capabilities document shall be checked to ensure that it corrected represents the specific server. These checks shall include checking that the:

a) OperationsMetadata section lists all and only the operations implemented by this server

b) All URLs listed for each operation are supported and provide correct results

c) Contents section lists all and only the abilities implemented by this server

A.4 Transform operation testing

The Transform operation shall be tested by using that operation to perform each of the selected set of coordinate transformations on a selected set of example data. Transforming features shall be tested if the server implements transforming features, and the set of features tested shall include each geometry type and feature format that is implemented by the server. Transforming coverages shall be tested if the server implements transforming coverages, and the set of coverages tested shall include each coverage type, coverage format, and interpolation method that is implemented by the server.

KVP, (plain) XML, and/or SOAP encoding of the Transform operation request shall be tested, depending on which encoding(s) are implemented. In addition to testing this operation with correct inputs, this operation shall be tested with incorrect inputs, including at least one incorrect value for each operation request parameter (TBR).

The response from each Transform operation requested shall be checked for the proper correspondence to the operation inputs, depending on the specific coordinate transformation and input feature(s) or coverage. Each response shall also be checked to ensure that it is valid with respect to the XML Schema definition of that response.

NOTE The bulk of the work involved in preparing conformance tests will be in obtaining the "correct" outputs that correspond to the various inputs, for each coordinate transformation. For each "correct" output coordinate value, a suitable allowed error tolerance must also be selected. Significant work is also required to find suitable inputs, including the coordinates of one or more points for each coordinate transformation tested.

A.5 IsTransformable operation testing

If implemented, the IsTransformable operation shall be tested by using that operation to check the transformability of a selected set of examples. These examples shall be selected to use a variety of values for the IsTransformable operation request parameters, targeting a set of the coordinate Transformations that are implemented by the server.

KVP, (plain) XML, and/or SOAP encoding of the IsTransformable operation request shall be tested, depending on which encoding(s) are implemented. In addition to testing this operation with correct inputs, this operation shall be tested with incorrect inputs, including at least one incorrect value for each operation request parameter (TBR).

The response from each IsTransformable operation requested shall be checked for the proper correspondence to the operation inputs, depending on the specific coordinate transformations and other abilities implemented. Each response shall also be checked to ensure that it is valid with respect to the XML Schema definition of that response.

A.6 GetTransformation operation testing

If implemented, the GetTransformation operation shall be tested by using that operation to retrieve a variety of coordinate transformation definitions, singly and in groups. These examples shall be selected from the set of the coordinate transformations that are implemented by the server, using the SourceCRS and TargetCRS of each selected transformation.

KVP, (plain) XML, and/or SOAP encoding of the GetTransformation operation request shall be tested, depending on which encoding(s) are implemented. In addition to testing this operation with correct inputs, this operation shall be tested with incorrect inputs, including at least one incorrect value for each operation request parameter (TBR).

The response from each GetTransformation operation requested shall be checked for the proper correspondence to the operation inputs, depending on the specific coordinate transformations implemented. Each response shall also be checked to ensure that it is valid with respect to the XML Schema definition of that response.

A.7 GetResourceByID operation testing

If implemented, the GetResourceByID operation shall be tested by using that operation to retrieve a variety of coordinate transformation, CRS, and operation method definitions, singly and in groups. These examples shall be selected from the sets of the coordinate transformations, CRSs, and operation methods that are implemented by the server, using the identifier of each selected objects.

KVP, (plain) XML, and/or SOAP encoding of the GetResourceByID operation request shall be tested, depending on which encoding(s) are implemented. In addition to testing this operation with correct inputs, this operation shall be tested with incorrect inputs, including at least one incorrect value for each operation request parameter (TBR).

The response from each GetResourceByID operation requested shall be checked for the proper correspondence to the operation inputs, depending on the specific objects implemented. Each response shall also be checked to ensure that it is valid with respect to the XML Schema definition of that response.

Annex B
(normative)

Conformance classes

B.1 Introduction

This annex specifies a set of implementation conformance classes for this WCTS specification. These conformance classes include one core class that shall always be implemented, plus many extension conformance classes (or modules) which may also be implemented. These extension conformance classes are based on the optional abilities specified and implied in this document.

NOTE 1 Most of the possible extension conformance classes listed below are intended to be independently selectable server abilities, although interface and implementation details will often overlap. However, at least one of the extension classes in some categories must be implemented. Extension classes that extend other classes can be defined, but few of the possible classes listed below extend other listed classes.

NOTE 2 One could define extension conformance classes that are mutually exclusive, but none of the extension classes listed below are (intended to be) mutually exclusive. This was done to simplify selecting extension class groups for interoperability, as discussed below.

EDITOR'S NOTE This annex is currently a first draft, and defines too many individual extension classes. Therefore, most of these extension classes should be combined into a few groups. On the other hand, some of these extension classes could be sub-divided.

B.2 WCTS core conformance class

The WCTS core conformance class includes the abilities that shall always be implemented, and excludes all the abilities included in all the extension conformance classes defined below. Briefly, the core conformance class includes implementation and testing of the:

a) Transform operation, with the operation request parameters:
    1) service
    2) request
    3) version
    4) SourceAndTargetCRSs
    5) Transformation
    6) InputData
b) GetCapabilities operation, with the operation request parameters:
    1) service
    2) request

3) AcceptVersions

c) Capabilities document, with the Sections:

1) ServiceIdentification

2) ServiceProvider

3) OperationsMetadata

4) Contents, not including FeatureAbilities and CoverageAbilities

d) (At least one specific coordinate Transformation supported by Transform operation, one of multiple alternate profiles)

B.3 WCTS additional operation conformance classes

Some WCTS extension conformance classes support additional operations, providing more abilities. These "operation" conformance classes include (optional) implementation and testing of zero or more of the following:

a) GetTransformation operation, with all operation request parameters

b) IsTransformable operation, with the operation request parameters:

1) service

2) request

3) version

4) SourceAndTargetCRSs

5) Transformation

6) Method, if any Methods are implemented that the server can use in user-defined coordinate Transformations

c) GetResourceByID operation, with all operation request parameters, to get one or more of following:

1) One or more CRS definitions

2) One or more Transformation definitions

3) One or more Method definitions that the server can use in user-defined coordinate Transformations

B.4 Operation request encoding conformance classes

Some WCTS extension conformance classes support the alternate methods of encoding all implemented operation requests. These "encoding" conformance classes include (optional) implementation and testing of ONE or more of the following:

a) KVP encoding in HTTP GET transfer of all implemented operation requests (in addition to the GetCapabilities operation)

b) XML encoding in HTTP POST transfer of all implemented operation requests

c) SOAP encoding (in HTTP POST transfer) of all implemented operation requests and responses

B.5 Operation request parameter conformance classes

Some WCS extension conformance classes support additional (optional) parameters in operation requests. These "parameters" conformance classes include (optional) implementation and testing of zero or more of the following:

a) Support optional Sections parameter in GetCapabilities operation request

b) Support optional UpdateSequence parameter in GetCapabilities operation request

c) Support optional AcceptFormats parameter in GetCapabilities operation request

d) Support optional "store" parameter in Transform operation request

e) Support OutputFormat parameter in Transform operation request that is different than the format of the input, supporting changing formats

B.6 Other general conformance classes

Some WCS extension conformance classes support other options that are independent of abilities to transform (non grid coverage) features and/or grid coverages, which are listed later. These "general" conformance classes include (optional) implementation and testing of zero or more of the following:

a) Each specific coordinate Transformation supported by Transform operation (multiple profiles, at least one implemented)

b) Each specific operation Method, which servers can use in user-defined coordinate Transformations, supported by Transform operations (multiple profiles, zero or more implemented)

B.7 Features transformation core conformance class

Some of the WCTS extension conformance classes support coordinate transformation of (non grid coverage) features. These extension conformance classes may be implemented independent of whether any extension conformance classes are implemented to support coordinate transformation of grid coverages. In order to transform features, a "feature core" extension profile may be provided, that includes implementing and testing of ALL of the following abilities:

a) FeatureAbilities data structure in the Contents section of the Capabilities document, with all its contents

b) Additional GeometryType parameter in the IsTransformable operation request

c) Ability to transform features, feature collections, and geometries

d) (At least one specific geometry type supported in inputs, one of multiple alternate profiles)

e) (At least one specific(non-coverage) feature format supported in inputs, one of multiple alternate profiles)

B.8 Features transformation additional conformance classes

In addition to the above "feature core" profile, additional feature extension conformance classes include (optional) implementation and testing of:

a) Additional OutputFormat parameter in the Transform operation request, supporting changing feature formats

b) Each specific (non-coverage) feature format supported in inputs (multiple profiles, at least one implemented)

c) Each specific group of geometry types supported in inputs (multiple profiles, at least one implemented)

NOTE We assume that the geometry types with similar transformation requirements will be grouped for implementation.

B.9 Coverages transformation core conformance class

Some of the WCTS extension conformance classes support coordinate transformation of grid coverages. These extension classes may be implemented independent of whether any extension conformance classes are implemented to support coordinate transformation of (non grid coverage) features. In order to transform coverages, a "coverage core" extension profile may be provided, that includes implementation and testing of ALL of the following abilities:

a) Additional parameters in the Transform operation request, both:
   1) GridCRS
   2) InterpolationType

b) Additional parameters in the IsTransformable operation request, both:
   1) CoverageType
   2) InterpolationType

c) Transform coverages with (only) one range scalar field

d) QuadrilateralGrid and GRID coverage type supported in inputs

NOTE We assume that only grid coverage transformation will be implemented in the short term.

B.10 Coverage resampling conformance classes

In addition to the above "coverage core" conformance class, some WCTS extension conformance classes support optional grid position interpolation abilities. These "coverage resampling" conformance classes include (optional) implementation and testing of zero or more of the following:

a) Support rotation and skew grid resampling by Transform operation

b) Support non-linear Transformations between CRSs

c) Each specific coordinate Transformation supported by Transform operation for georectified coverages (multiple profiles)

NOTE These Transformations are used to change the base CRS of GridCRSs.

d) Each specific coverage rectification transformation method supported by Transform operation for unrectified coverages (multiple profiles)

e) Each specific grid elevation data format supported by Transform operation for image orthorectification (multiple profiles)

B.11 Interpolation method conformance classes

In addition to the above "coverage core" conformance class, some WCTS extension conformance classes support implementing different interpolation methods for resampling of offered grid coverages. These "interpolation method" conformance classes include (optional) implementation and testing of ONE or more of the following:

a) Support the "none" interpolation method by GetCoverage operation

b) Support the "nearest" interpolation method by GetCoverage operation

c) Support the "linear" interpolation method by GetCoverage operation

d) Support the "cubic" interpolation method by GetCoverage operation

e) Support the "quadratic" interpolation method by GetCoverage operation

f) Each additional specific interpolation method supported by GetCoverage operation (multiple classes)

B.12 Coverage domain conformance classes

In addition to the above "coverage core" conformance class, some WCTS extension conformance classes support coverage domain options, to handle different numbers and types of domain dimensions. These "coverage domain" conformance classes include (optional) implementation and testing of ONE or more of the following:

a) Transform coverages with two spatial domain dimensions (not necessarily both horizontal)

b) Transform coverages with three spatial domain dimensions

c) Transform coverages with one spatial domain dimension (not necessarily horizontal)

d) Transform coverages with one temporal domain dimension

e) Transform coverages with two or more temporal domain dimensions

f) Transform coverages with one or more non-spatial-temporal domain dimensions

B.13 Coverage range conformance classes

In addition to the above "coverage core" conformance class, some WCTS extension conformance classes support coverage range options, to handle more complex ranges than one scalar field in a coverage. These "coverage range" conformance classes include (optional) implementation and testing of zero or more of the following:

a) Support more than one range field

b) Support vector range field(s), with one axis in a range field

c) Support more than one axis in range field(s) (requires above profile)

d) Support RangeSubset in GetCoverage operation request (requires one or more above conformance classes)

B.14 Grid coverages transformation additional conformance classes

In addition to the above coverage extension conformance classes, additional coverage extension conformance classes include (optional) implementation and testing of:

a) Each specific grid coverage format supported in inputs (multiple profiles, including how this format is used)

b) Additional OutputFormat parameter in the Transform operation request, supporting changing coverage formats

NOTE Many people will think that some of the extension conformance classes listed above should be included in the core profile, or should be combined with another extension profile. However, a simple WCTS for rectifying (unrectified) ordinary images in one format might include only a few of these extension conformance classes.

B.15 Interoperability

For one client to use one server, each must implement the core conformance class plus a compatible set of WCTS conformance classes. That client will require use of some set of extension conformance classes in addition to the core conformance class. That is, this client will use some set of extension conformance classes when it is used in one or more use cases (which should be identified and specified). That server must implement at least this required set of extension conformance classes, but can implement more extensions.

One definition of implementation interoperability is that two or more clients can equally use two or more servers. Such interoperability requires that all these clients and servers implement a shared set of extension classes. Each server in such an interoperable group must implement at least this shared set of extension classes, but can implement more extensions. Each client in such an interoperable group must not require implementation of any additional extensions, but can require implementation of only a subset of that shared set.

It is possible for one interoperable group of clients and servers to share two or more sets of extension conformance classes. Each server in such an interoperable group must implement all these sets of extension classes. Each client in such an interoperable group must require implementation of (a subset of) one or more of those sets of extension classes. For example, these sets might differ only in which one of the extension classes in the operation request encoding category must be implemented.

NOTE Implementation of some of these listed extension conformance classes will be, or may be, explicitly listed as implemented Profiles in the OperationsMetadata Section of the Capabilities document. However, such explicit indication does NOT reduce the needs stated above for interoperable clients and servers to implement shared set(s) of extension conformance classes.

Annex C
(normative)

XML schema documents

The coordinate transformation abilities now specified in this document use the XML Schema specified in nine normative XML Schema Documents, bundled in a zip file with the present document. These XML Schema files roughly match the eight UML packages defined in Annex E, and are named:

a) wctsCommon.xsd

b) wctsContents.xsd

c) wctsGetCapabilities.xsd

d) wctsGetResourceByID.xsd

e) wctsGetTransformation.xsd

f) wctsIsTransformable.xsd

g) wctsOperationParameter.xsd

h) wctsTransform.xsd

i) wctsAll.xsd

The WCTS Schema references three XML documents that specify some values and their meanings of the three UML CodeList classes, namely:

a) coverageTypes.xml, referenced at http://schemas.opengis.net/wcts/0.0.0/coverageTypes.xml

b) geometryTypes.xml, referenced at http://schemas.opengis.net/wcts/0.0.0/geometryTypes.xml

c) problemTypes.xml, referenced at http://schemas.opengis.net/wcts/0.0.0/problemTypes.xml

The WCTS Schema builds on the OWS common XML Schema Documents listed in Annex A of [OGC 06-121r3], and named:

a) ows19115subset.xsd

b) owsCommon.xsd

c) owsContents.xsd

d) owsDataIdentification.xsd

e) owsDomainType.xsd

f) owsExceptionReport.xsd

g) owsGetCapabilities.xsd

h) owsGetResourceByID.xsd

i) owsOperationsMetadata.xsd

j) owsServiceIdentification.xsd

k) owsServiceProvider.xsd

l) owsAll.xsd

This WCTS Schema also uses parts of the WCS 1.1 XML Schema listed in Annex TBD of [OGC 07-067r2], and named:

a) wcsGridCRS.xsd

b) wcsInterpolationMethod.xsd

This WCTS Schema also uses parts of the CRS-related XML Schemas for the GML 3.1.1 grid CRSs profile, specified in [OGC 05-096r1] and named

a) coordinateOperations.xsd

b) coordinateReferenceSystems.xsd

c) coordinateSystems.xsd

d) datums.xsd

e) dataQuality.xsd

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

EXAMPLES Most of the example XML documents listed in this document are included in the files attached to this document.

Annex D
(informative)

Use cases

D.1 Introduction

Transformation of geospatial data from one CRS to another is frequently required when integrating data from different sources into one application. Not all applications are capable of performing such transformations. This annex describes use cases that show how a web based transformation service can solve this problem. These use cases also help clarify the requirements on a WCTS interface and its' implementations.

The basis of all use cases is the following scenario:

A map must be created that uses EPSG:23032 (UTM band 32) as its CRS. The borders of the Federal Republic of Germany (layer 1), the major rivers of the Federal Republic of Germany (layer 2), and the largest cities of the Federal Republic of Germany (layer 3) shall be used to create this map. The borders of the Federal Republic of Germany are available in EPSG:31467 (Gauß-Krüger band 3). The major rivers are stored in EPSG:4230 (International 1924), and the largest cities are in EPSG:4326

D.2 Desktop application

The term "Desktop Application" describes an application running on a PC that accesses the data it uses through a file system. It is assumed that this application is allowed to access a network (internet or intranet) where a WCTS is available. It is also assumed that all map layers defined by the scenario are available from a local data store. Figure D.1 shows a diagram for this use case.

EDITORS NOTE The diagram format used in these four figures needs to be explained!

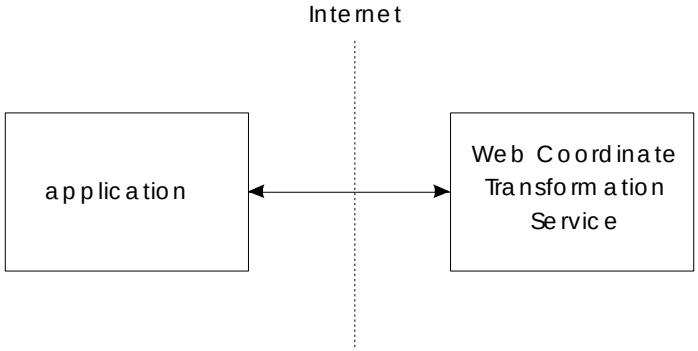

Figure D.1 — Desktop Application use case: application - WCTS interaction

a) The application defines a new map with the target CRS.

b) The application loads layer 1 and recognises that its' coordinates are not stored in EPSG:4326.

c) The application connects to a WCTS through the network, and requests its capabilities.

d) Using the WCTS capabilities, the application evaluates if a transformation from EPSG:31467 to EPSG:4326 can be performed by the WCTS. (This can also be done by sending an IsTransformable request to the WCTS.)

e) The application sends the geospatial data of layer 1 to the WCTS requesting their transformation to EPSG:4326.

f) The WCTS sends the transformed data back to the application.

g) Steps b) to f) are repeated for layers 2 and 3.

h) The application renders the map by combining the transformed data.

D.3 WMS

The map defined in the scenario could be requested by a client through a network using a valid Web Map Service (WMS) GetMap request. All required data are assumed to be stored in sources that can be directly accessed by the WMS (an integrated WMS). Figure D.2 shows a diagram for this use case.

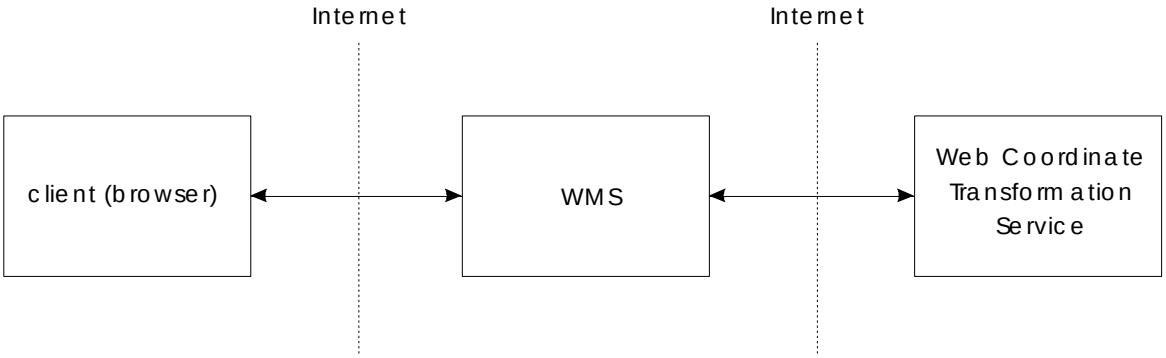

Figure D.2 — WMS use case: client - WMS - WCTS interaction

The sequence of steps is similar to the Desktop Application use case. First, however, the client must inquire of the WMS (using a GetCapabilities request) whether it can render the map in the desired coordinate reference system. Afterwards the map will be rendered and passed to the client (web browser) via the internet or intranet.

The following steps are required in this use case:

a) The client defines the parameters for a new map.

b) The client posts a GetMap request to a WMS.

c) The WMS determines that the requested layer is not stored in the correct CRS.

d) The WMS already knows that the WCTS offers the necessary transformation; it therefore sends the data of the first layer for transformation to the WCTS.

e) Step d) is repeated for layers 2 and 3.

f) The WMS renders the map and sends it back to the client (web browser)

D.4 Service chain

This use case is the same as the WMS use case except the data to be rendered are not directly stored by the WMS. Therefore, the WMS must request them via the Internet from a WFS (Web Feature Service (a component WMS). This WFS supplies the data in the CRS specified above. Figure D.3 shows a diagram for this use case.

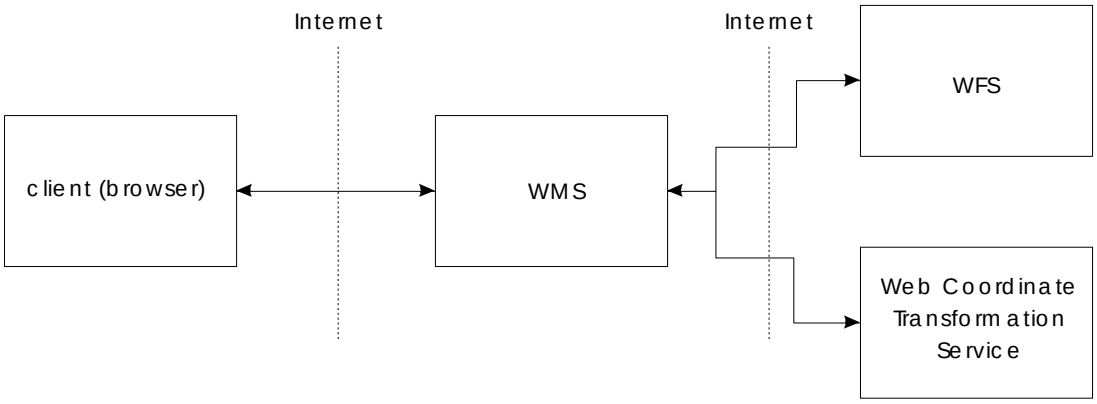

Figure D.3 — Service Chain use case: client - WMS - WFS - WCTS interaction

In this use case, the WMS sends a request to the WCTS, providing the input data as a getFeature operation request to a specific WFS. The WCTS then sends this request to the WFS, transforms the coordinates and sends those as response to the WMS. The remaining steps are performed as described above.

EDITORS NOTE The above paragraph does not match the figure. Is the figure wrong?

In principle, the model is expandable for further levels e.g., in the case of a cascading WMS or a service chain. Figure D.4 shows a diagram for this use case.

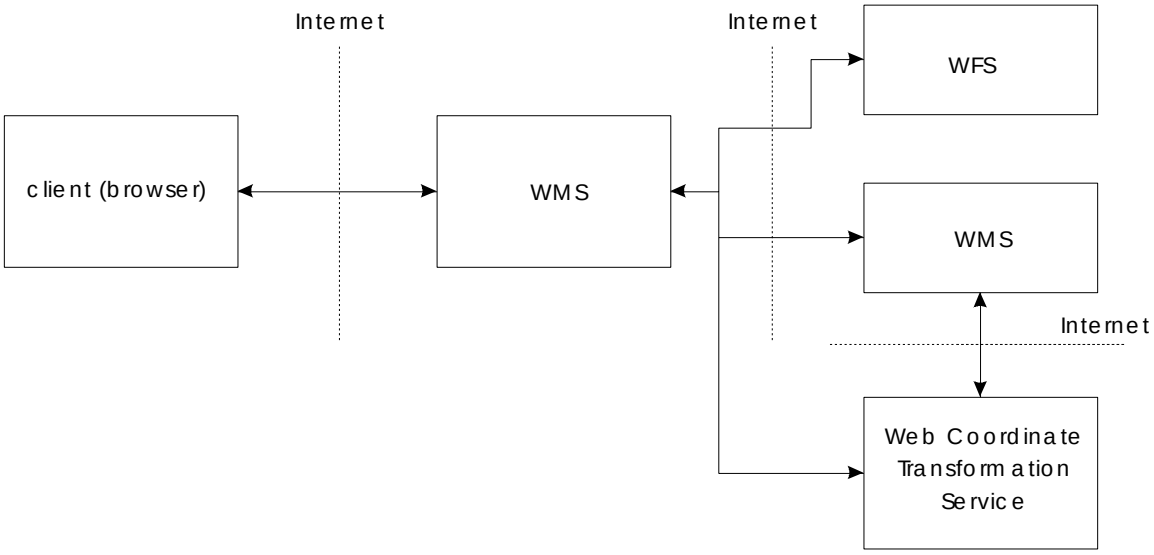

Figure D.4 — Service Chain use case two4: service chain

EDITORS NOTE The above figure needs an explanation.

In the case of a cascading WMS, the WMS actually providing each layer must supply the requested Layer in the target CRS, since the transformation of a raster image would be too expensive. In addition, the result of transforming a raster image possibly would not meet expectations (e.g., by substantial distortions of the labelling).

D.5 Defining and using user-defined coordinate transformation

EDITOR'S NOTE This subclause needs further editing.

In addition to known coordinate transformations, a WCTS server can implement user-defined coordinate transformations, which use known operation methods with parameters whose values are supplied by user. For example, it is possible to perform image orthorectification, using a complex transformation that uses image geometry (sensor) model data, Digital Elevation Model (DEM), and perhaps ground control points.

The operation method definitions, including the list of parameters with their valid values (for example URI for DEM or SensorML resource, coefficient of equations etc.), can be obtained using the GetResourceByID operation.

The following steps are required in this use case:

a) A user wants to perform a custom transformation based on a well-known method.

b) The user performs a GetCapabilities request to the WCTS to retrieve the list of supported methods.

c) Based on the brief descriptions of the methods or their well-known identifiers, the user selects the operation method of interest.

d) The user invokes GetResourceByID operation to obtain the method definition from the WCTS by providing the methodID published in the Contents section of the Capabilities document.

e) The server returns the wcts:OperationMethod description along with the list of parameters its supports. Each parameter has a unique name, cardinality information, list of valid values and supported Unit of measures.

f) The parameters description is used to build a form on the client side, which the parameters value to validate the user inputs.

g) The client builds an Operation based on the method parameters and refers the Method identifier in the Coordinate Operation. Select the input data and send a Transform query.

h) The Server process the query and performs the transformation.

i) The WCTS sends the response back.

Annex E
(informative)

UML model

E.1 Introduction

This annex provides a UML model of the WCTS interface, using the OGC/ISO profile of UML summarized in Subclause 5.3 of [OGC 06-121r3]. This UML model builds on the UML model for OWS Common, specified in Annex C of [OGC 06-121r3]. This model also builds on the UML model specified in [ISO 19111] and OGC Abstract Specification Topic 2.

Figure E.1 is a simple UML diagram summarizing the WCTS interface. This class diagram shows that the WCTService class inherits the getCapabilities and GetResourceByID operations from the OGCWebService interface class, and adds the "transform", isTransformable, and getTransformation operations. (This capitalization of names uses the OGC/ISO profile of UML.)

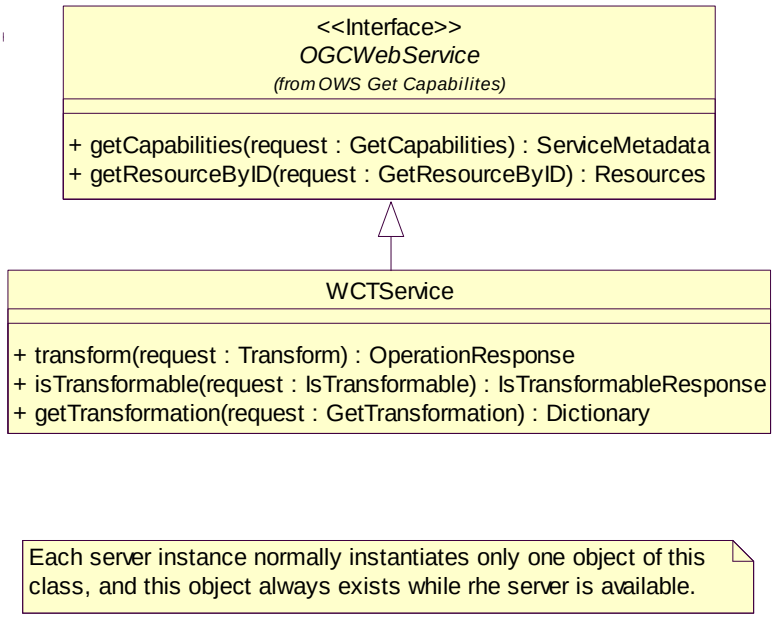

Figure E.1 — WCTS interface UML diagram

Each of the six operations uses a request and a response data structure, each of which is defined by one or more additional UML classes. The following subclauses provide a more complete UML model of the WCTS interface, adding UML classes defining the operation request and response data types. This annex uses the capitalization of names defined in the OGC/ISO profile of UML, which is different from the capitalization of names used in XML encoding.

E.2 UML packages

The WCTS interface UML model is organized in 7 packages, as shown in the package diagram in Figure E.2. These WCTS-specific packages make direct use of 5 OWS common packages, named OWS Common, OWS Input Output Data, OWS GetCapabilities, OWS Get resource By ID, and OWS Domain. The WCTS-specific packages also make direct and indirect use of all six packages in the UML model for OGC Abstract Specification Topic 2 [OGC 05-103], here shown as part of an ISO 19111 package.

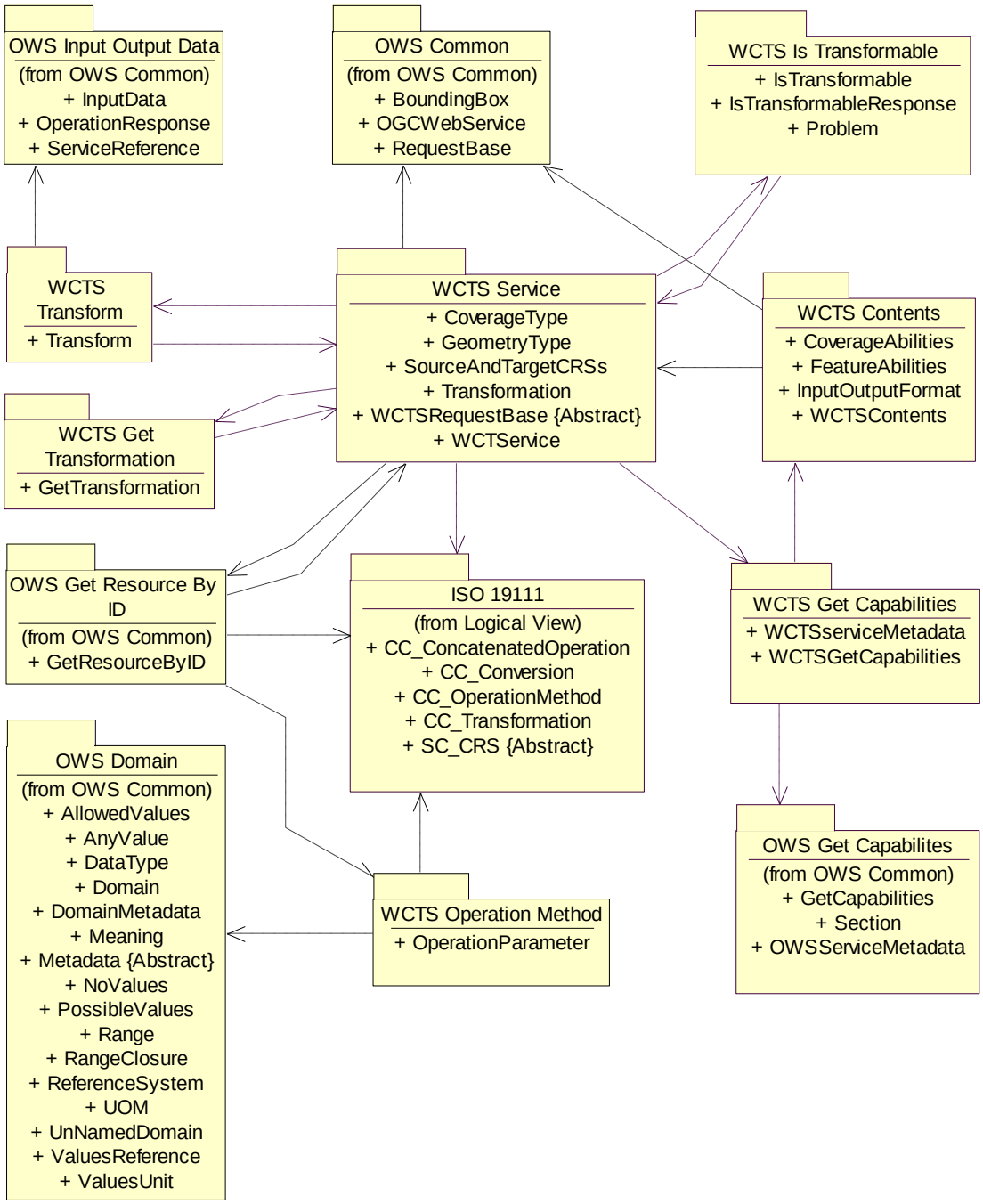

Figure E.2 — WCTS interface package diagram

This package diagram shows the dependencies among the WCTS packages, and the direct dependencies of the WCTS packages on OWS Common packages. This diagram does not show the dependencies among the OWS Common packages. This diagram does not show two packages of the WCS 1.1 UML model, named WCS GridCRS and Interpolation Methods. The WCTS Transform package uses both of these WCS packages, and the WCTS Is Transformable package uses the Interpolation Methods package.

Each of the WCTS-specific packages shown in Figure E.2 is described in the following subclauses. The OWS Common packages are described in Annex C of [OGC 06-121r3]. The ISO 19111 packages are described in the UML model in Clauses 7-13 of [OGC 05-103].

E.3 WCTS Service package

The WCTS Service package is shown in the class diagram in Figure E.3. This diagram does not show the classes used by the six operation requests and responses, which are shown (with part of this package) in the WCTS Transform, WCTS Is Transformable, WCTS Get Transformation, and WCTS Get Capabilities packages. This diagram also shows two used classes from the OWS Web Service package, which is common to all OGC Web Services, plus one used class from ISO 19111. The four classes introduced by this package are further defined by Table 1 through Table 4 in this document.

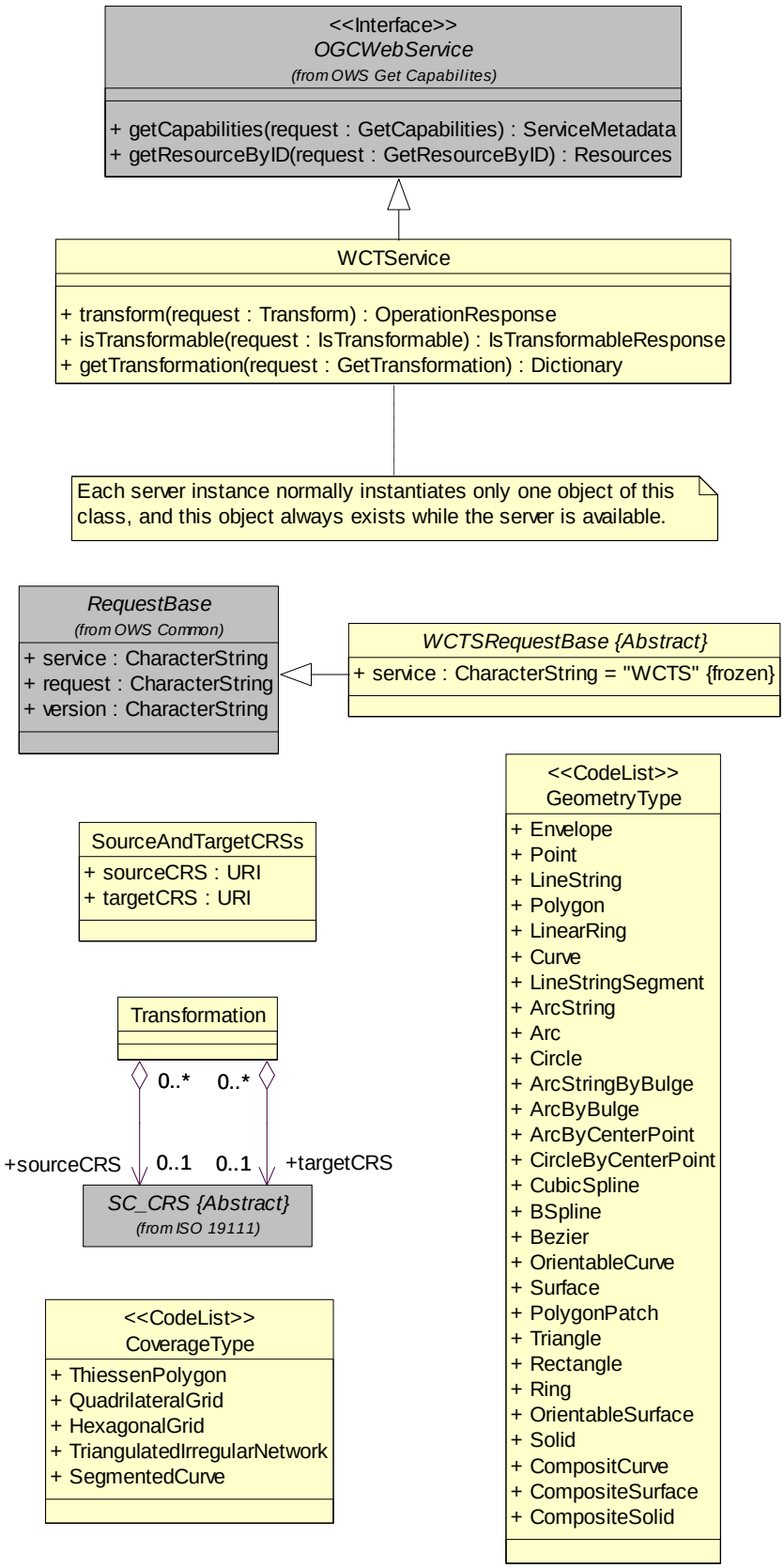

Figure E.3 — WCTS Service package class diagram

E.4 WCTS Transform package

The WCTS Transform package is shown in the class diagram in Figures E.4 and E.5. This diagram also shows two classes of the WCTS Service package, three classes from OWS Common, and two classes from WCS 1.1. The Transform and TransformationChoice classes introduced by this package are further defined by Table 7 and Table 8 in this document.

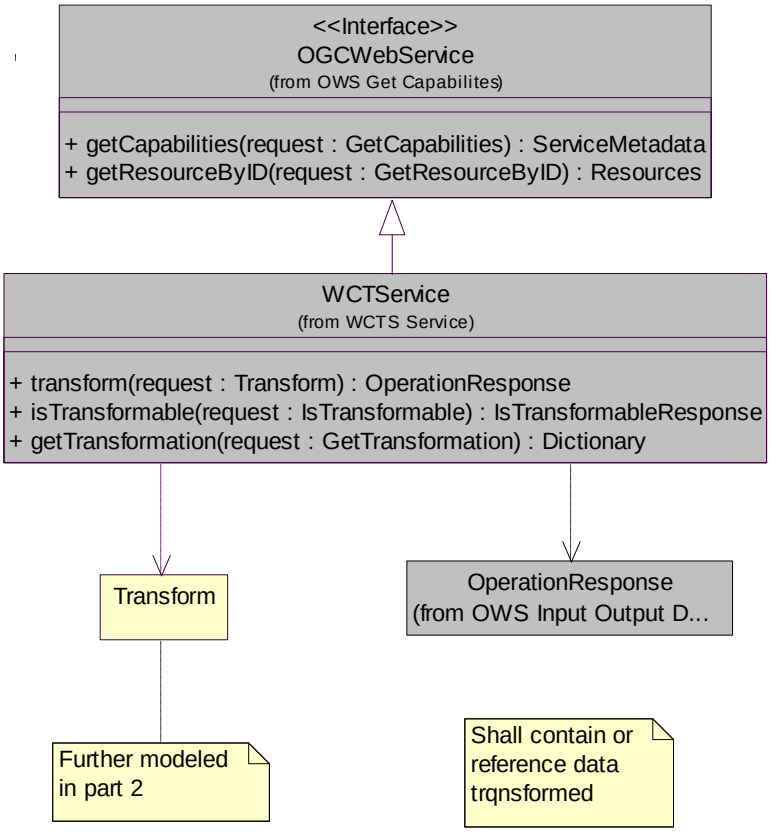

Figure E.4 — WCTS Transform package class diagram, part 1

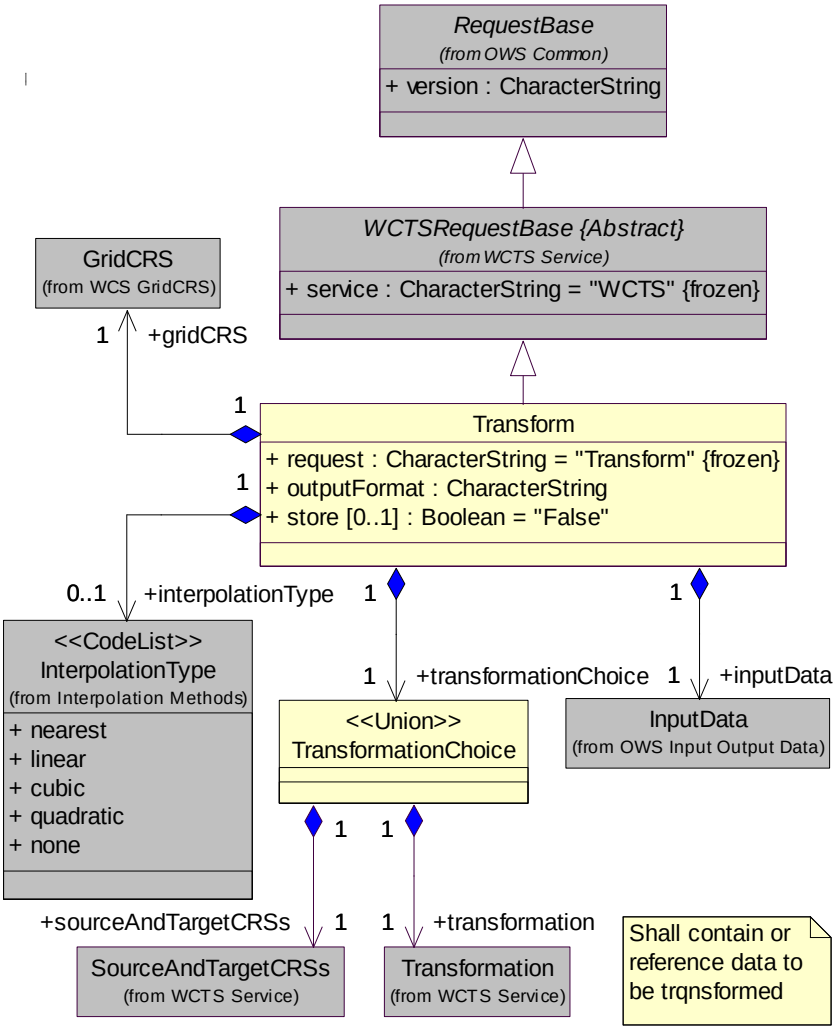

Figure E.5 — WCTS Transform package class diagram, part 2

E.5 WCTS Is Transformable package

The WCTS Is Transformable package is shown in the class diagram in Figure E.6. This diagram also shows two classes of the WCTS Service package plus several used classes from OWS Common packages. The four classes introduced by this package are further defined in Table 23 through Table 28 in this document.

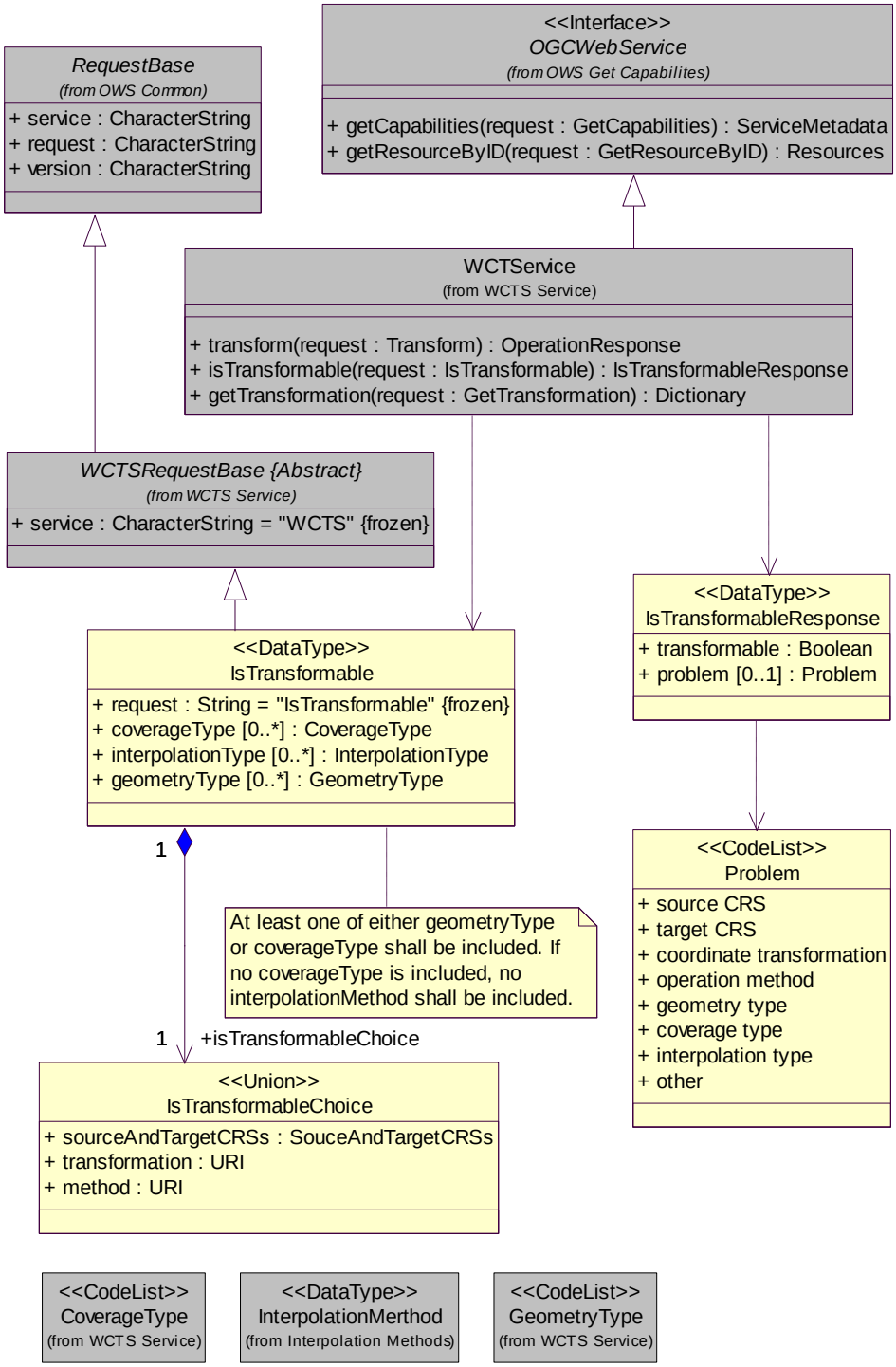

Figure E.6 — WCTS Is Transformable package class diagram

E.6 WCTS Get Transformation package

The WCTS Get Transformation package is shown in the class diagram in Figure E.7. This diagram also shows two classes of the WCTS Service package plus several used classes from OWS Common packages. The GetTransformation class introduced by this package is further defined by Table 29 in this document.

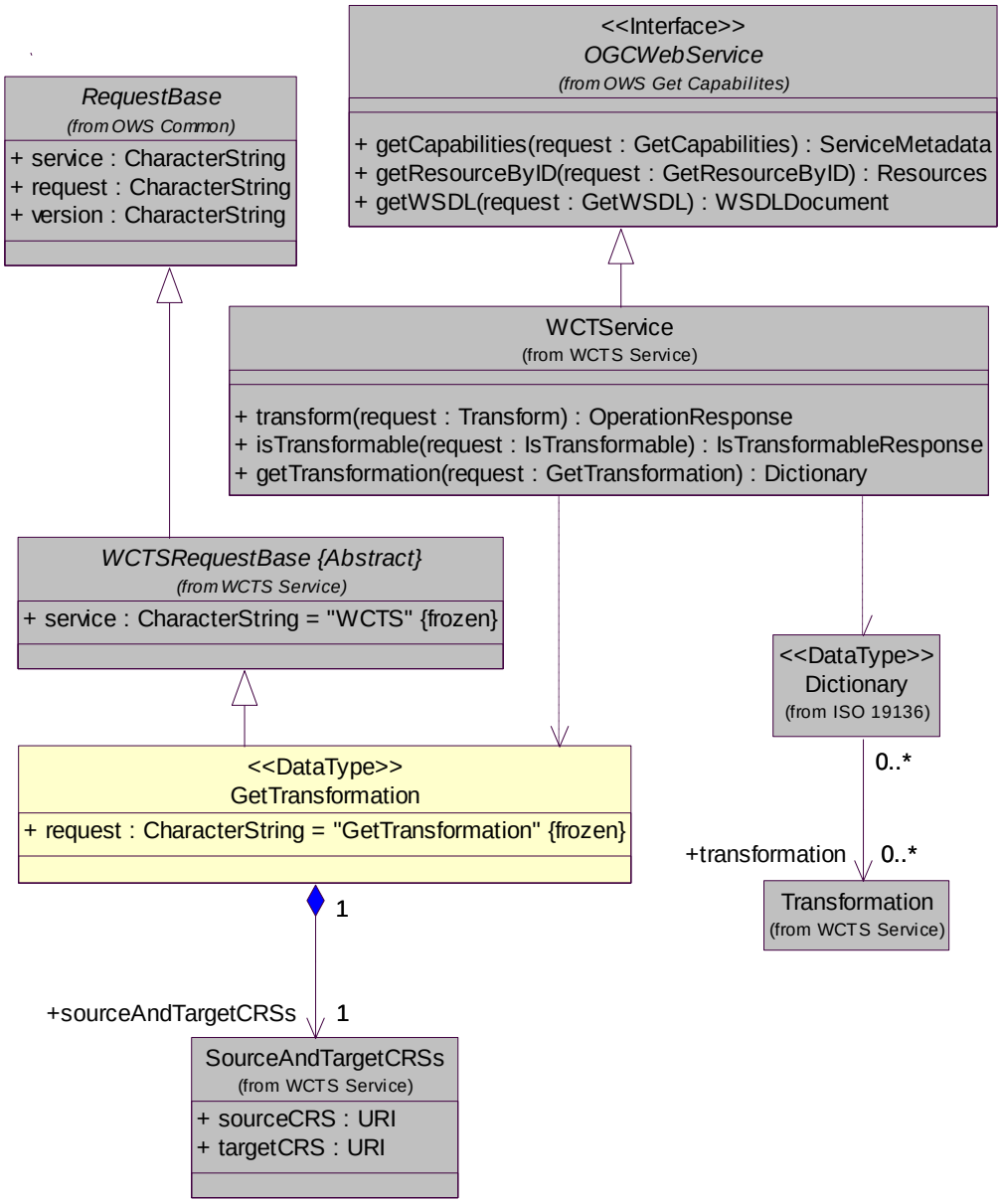

Figure E.7 — WCTS Get Transformation package class diagram

E.7 WCTS Get Capabilities package

The WCTS Get Capabilities package is shown in the class diagram in Figure E.8. This diagram does not show details of the WCTSContents, ServiceIdentification, ServiceProvider, and OperationsMetadata classes, which are in separate packages. The WCTSContents class is in the WCTS Contents package that is described in the following subclause. The ServiceIdentification, ServiceProvider, and OperationsMetadata classes are in the OWS Service Identification, OWS Service Provider, and OWS Operations Metadata packages that are described in Annex B of [OGC 06-121r3].

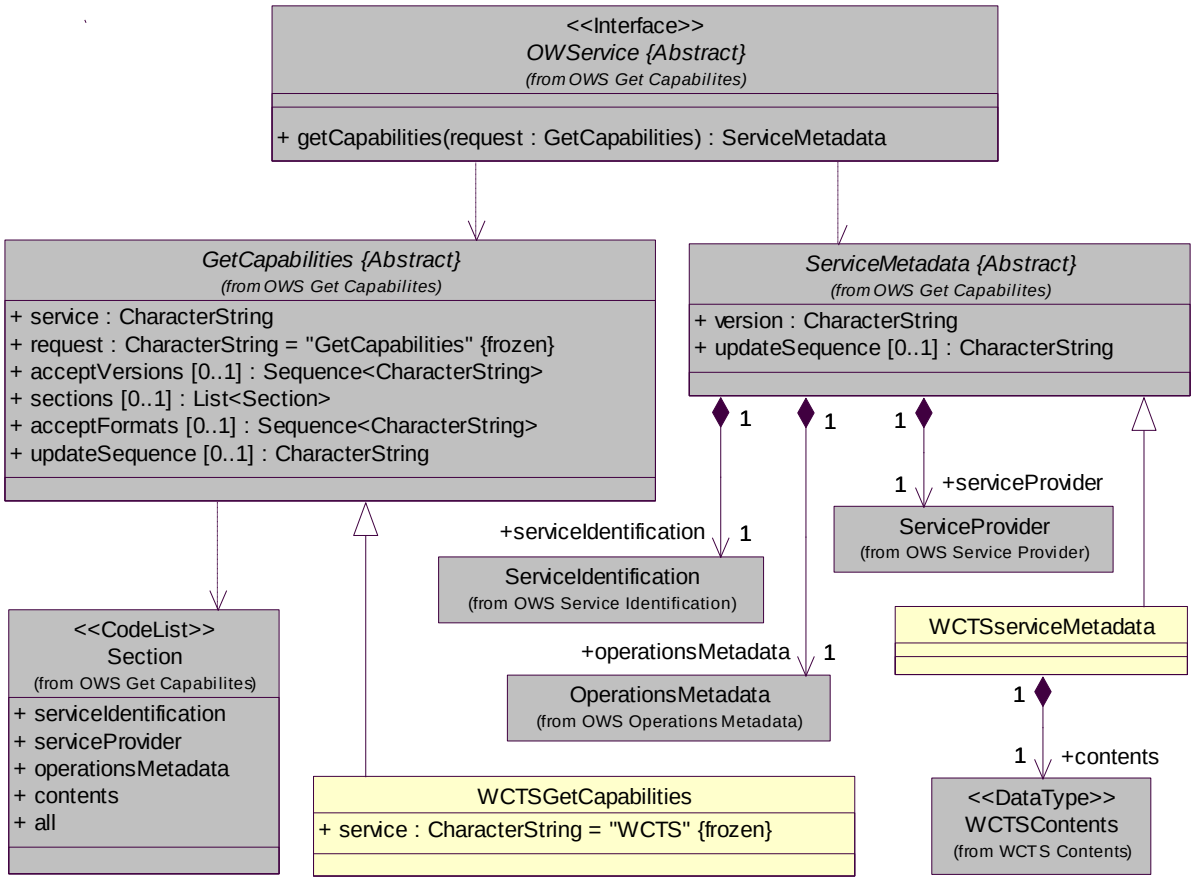

Figure E.8 — WCTS Get Capabilities package class diagram

E.8 WCTS Contents package

The WCTS Contents package is shown in the class diagram in Figure E.9. This diagram also shows one class from the WCTS Get Capabilities package. The WCTSContents, CoverageAbilities, and FeatureAbilities classes introduced by this package are further defined by Table 19 through Table 22 in this document.

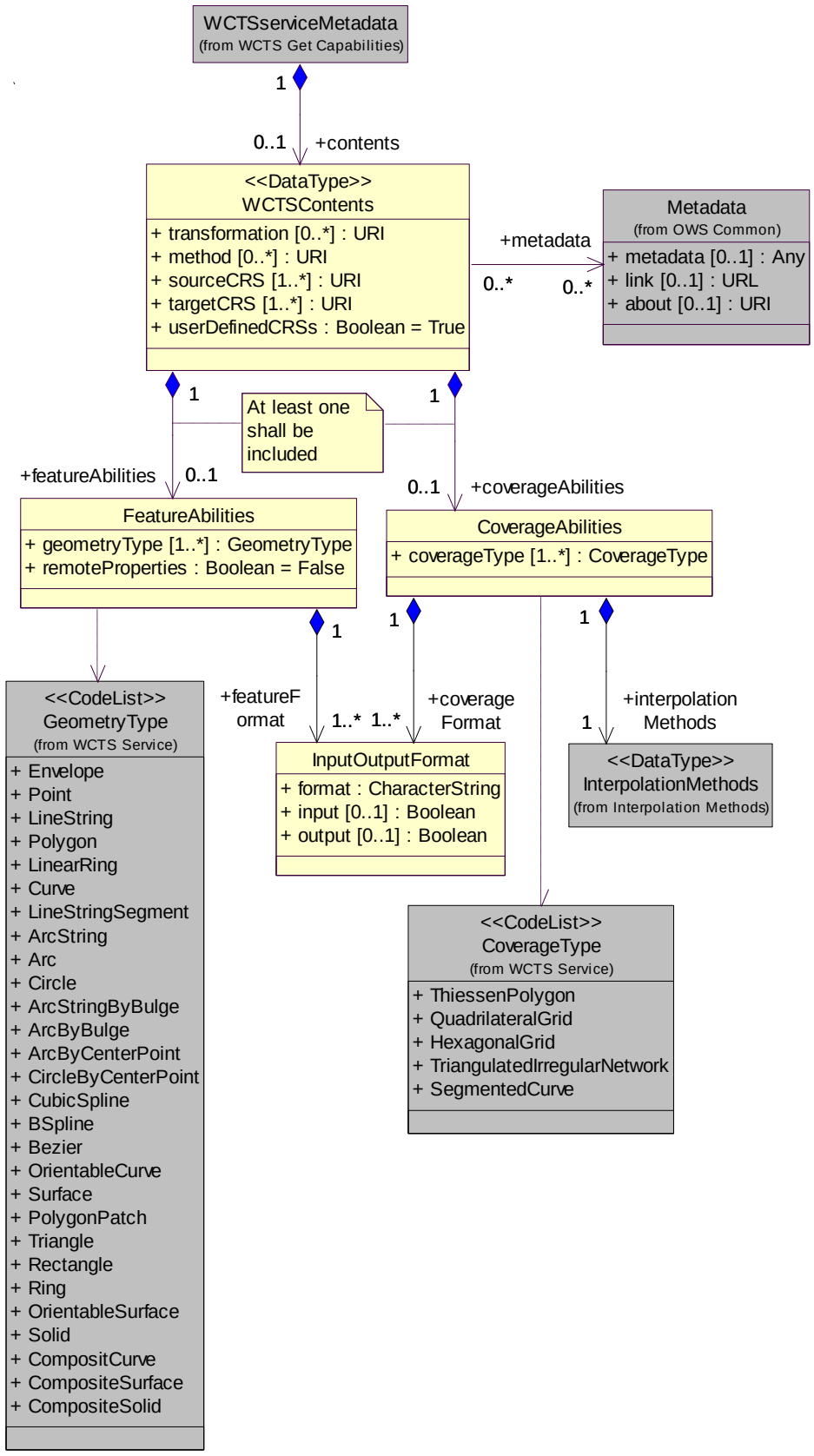

Figure E.9 — WCTS Contents package class diagram

E.9 WCTS Operation Method package

The WCTS Operation Method package is shown in the class diagram in Figure E.10. This package builds on the CC Coordinate Operation package of ISO 19111 [OGC 05-103]. This package uses the UnNamedDomain and UOM classes from the OWS Domain package.

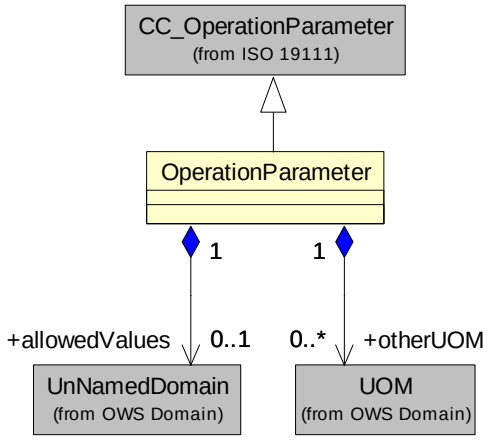

Figure E.10 — WCTS Operation Method package class diagram

Annex F
(informative)

GML application schemas for features

F.1 Introduction

This WCTS interface transforms the coordinates of points contained within feature geometries encoded in GML 3.1.1. In order to encode a feature in GML, a GML Application Schema is required, which specifies specific feature type name(s) and included geometry properties. This annex provides two such GML Application Schemas, one used in the examples in Subclause 8.4, and another for a pseudo-feature that can contain a sequence of point coordinates to be transformed that are not a real feature.

F.2 Simple example GML application schemas for features

Some of the examples in Subclause 8.4 use this simple example GML 3.1.1 Application Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengeospatial.net/examples"
xmlns:ex="http://www.opengeospatial.net/examples"
xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
version="0.0">
   <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/07 -->
   <import namespace="http://www.opengis.net/gml"
schemaLocation="../gml/3.1.1/base/feature.xsd"/>
   <element name="Capital" type="ex:CapitalType"
substitutionGroup="gml:_Feature"/>
   <complexType name="CapitalType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element name="cityName" type="string"/>
               <element name="stateName" type="string"/>
               <element ref="gml:pointProperty"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
</schema>
```

This example GML Application Schema defines a feature type named “Capital” with properties “cityName”, “stateName”, and “pointProperty”. The property “pointProperty” contains a gml:Point geometry primitive. One or more of these “Capital” features can be included in a feature collection.

F.3 GML application schemas for pseudo-features

To transform the coordinates of a sequence of points, a pseudo-feature can be used. A simple GML Application Schemas that can be used to encode an ordered sequence of point coordinates that are not a real feature is:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/wcts"
xmlns:wcts="http://www.opengis.net/wcts"
xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
version="0.0">
   <annotation>
      <appinfo>pointsApplicationSchema.xsd 2004/12/20</appinfo>
      <documentation>This XML Schema is a GML Application Schema for
encoding point positions in a pseudo-feature for transformation by a
WCTS.
      Primary editor: Arliss Whiteside. Last updated 2007-04-
09</documentation>
   </annotation>
   <!-- ============================================================
      includes and imports
   ============================================================= -->
   <import namespace="http://www.opengis.net/gml"
schemaLocation="../gml/3.1.1/base/feature.xsd"/>
   <!-- ============================================================
      elements and types
   ============================================================= -->
   <element name="Points" type="wcts:PointsType"
substitutionGroup="gml:_Feature"/>
   <complexType name="PointsType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element ref="gml:pointProperty" maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
</schema>
```

This GML Application Schema defines a feature type named “Points” with the property “pointProperty” which can be repeated. Each “pointProperty” contains a gml:Point geometry primitive. A template XML document containing three point coordinates using this Application Schema is:

```
<?xml version="1.0" encoding="UTF-8"?>
<Points xmlns="http://www.opengis.net/wcts"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcts
pointsApplicationSchema.xsd">
   <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/20 -->
   <gml:pointProperty>
      <gml:Point>
         <gml:pos>99.9999 99.9999</gml:pos>
      </gml:Point>
```



```
   </gml:pointProperty>
   <gml:pointProperty>
      <gml:Point>
         <gml:pos>99.9999 99.9999</gml:pos>
      </gml:Point>
   </gml:pointProperty>
   <gml:pointProperty>
      <gml:Point>
         <gml:pos>99.9999 99.9999</gml:pos>
      </gml:Point>
   </gml:pointProperty>
</Points>
```

In this template, the dummy coordinate values 99.9999 can be replaced by the desired coordinate values, using negative values where needed. The number of coordinates provided for each Point can be increased, and the number of Points can be increased or decreased.

If the WCTS server and client do not validate these feature XML documents, a shorter template could be used:

```
<?xml version="1.0" encoding="UTF-8"?>
<Points>
   <pointProperty>
      <Point>
         <pos>99.9999 99.9999</pos>
      </Point>
   </pointProperty>
   <pointProperty>
      <Point>
         <pos>99.9999 99.9999</pos>
      </Point>
   </pointProperty>
   <pointProperty>
      <Point>
         <pos>99.9999 99.9999</pos>
      </Point>
   </pointProperty>
</Points>
```


Annex G
(normative)

SOAP transfer

All compliant WCTS servers may implement SOAP 1.2 transfer of all WCTS 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 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 WCTS server shall return operation responses and error messages using only SOAP transfer when the operation request is sent using SOAP.

All compliant WCTS 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 7.4.7 of [OGC 06-121r3].

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:

```
<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 H
(informative)

Design decisions

H.1 Introduction

This annex briefly lists some of the decisions made in the design of the WCTS interface specified herein. These decisions are organized by included abilities, excluded abilities, implementation of abilities, and specification approaches.

EDITOR'S NOTE This annex needs to be updated based on recent changes to this document.

H.2 Included interface abilities

The decisions made to include certain interface abilities included:

a) Allow simple referencing of specified sets of well-known transformations (and other coordinate operations), coordinate reference systems, units, and operation methods with associated operation parameters. Require conforming server implementations to support a minimum subset of each of these specified sets, and allow implementation of additional specified subsets.

b) Allow returning a list of possible transformations or other coordinate operations in one GetTransformation operation response.

c) Allow returning (one or more) Concatenated Operation coordinate operations in a GetTransformation operation response, where each Concatenated Operation is treated as one coordinate operation.

d) Allow sending one Concatenated Operation in a Transform operation request, where this Concatenated Operation is defined as a sequence of two or other coordinate operations.

e) Allow use of any GML 3.1.1 Application Schema defining feature and feature collection types, in inputs to the Transform operation, as long as only supported geometric primitive types are included in the input features and feature collections.

f) Compute an accurate transformed "boundedBy" gml:Envelope in any enclosing feature or feature collection, by the Transform operation. (The boundedBy gml:Envelope is defined as the minimum bounding rectangle.)

g) When the IsTransformable operation returns "false", return one problem identifier text string, where identifiers are specified for the common error conditions.

h) Support one or more of either Feature or Remote OWS (e.g., WFS) inputs in one Transform operation request.

i) Allow handling operation methods that decrease the number of dimensions in each coordinate. (For example, allow image coordinate transformations from 3D ground coordinate to 2D image coordinates.) That is, the interface will be specified to allow

this, although no such operation methods will be specified or required to be implemented.

H.3 Excluded interface abilities

The decisions made to exclude and limit interface abilities, in this version of this specification, included:

a) Not support returning information on the quality of the available transformation(s), from the IsTransformable operation.

b) Not support returning information on the quality of the transformation performed, from the Transform operation.

c) Not require supporting transformations that can change the type of a geometry element. (For example, North Pole point becomes a line.)

d) Not check the contents of any input srsName attributes.

e) Not allow combinations of Feature, File, and Remote OWS (e.g., WFS) inputs in one Transform operation request.

f) Not support use of two-way coordinate operations and operation methods. (A coordinate operation and its operations method are generally two-way, as defined in Abstract Specification Topic 2.)

g) Not support user-defined operation methods.

h) Limit the set of GML geometric primitive types that can be handled by the Transform operation, to a specified set of types.

i) Not provide special abilities for transforming a single point or few points, not contained in a GML geometry or feature.

j) Not support additions made to GML 3.1.1 for identifying the axis order and units of coordinates, in addition to the srsName in geometry elements.

H.4 Implementation of interface abilities

The decisions made on how to implement needed interface abilities included:

a) In responses from the Transform operation, always use the same GML Application Schema as used for the input to that Transform operation.

b) Only modify the supported types of geometry elements found in the inputs to the Transform operation, and copy the rest of the input XML document to the output.

c) In responses from the Transform operation, include the target srsName values in exactly the same places as srsName values were included in input feature data.

d) Support URN and URL forms of srsName contents in responses from the Transform operation, and in the target SRS inputs to the Transform and IsTransformable operations. Support the URN and URL forms specified in the CRS Schemas Application Profile.

e) Use gml:CRSRefType for source and target CRS inputs to operations. In the gml:CRSRefType, usually use a GML association xlink:href (with type anyURI) and

use the URN or URL forms specified in the CRS Schemas Application Profile. Also allow in-line encoding of the CRS definition.

f) Use gml:CoordinateOperationRefType for transformation inputs to, and outputs from, operations. In the gml:CoordinateOperationRefType, usually use a GML association xlink:href (with type anyURI) and use the URN or URL forms specified in the CRS Schemas Application Profile. Also allow in-line encoding of the coordinate operation definition when the operation method and associated operation parameters are well-known, implemented by the server, and referenced.

H.5 Specification approaches

The documentation approaches used in this Implementation Specification include:

a) Use parts of the CRS Schemas specified in Recommendation Paper 04-092r4 when applicable, where the used parts constitute a GML 3.1.1 Profile. Normatively reference the applicable parts of GML 3.1.1 (04-092r4).

b) Specify standard URN values for well-known coordinate operations, coordinate reference systems, units, and operation methods with corresponding operation parameters.

c) Provide a standardized GML Application Schema, and corresponding example XML document template, of a simple feature containing one or a few points, for use when a client needs to transform a single point or few points. Do this in an informative Annex in the Implementation Specification.

d) Normatively reference parts of OWS Common Specification [OGC 06-121r3], instead of copying this material.

Bibliography

[1] ISO 19118, Geographic information – Encoding

[2] OGC 01-004, Implementation Specification: Grid Coverage

[3] OGC 01-009, Implementation Specification: Coordinate Transformation Services

[4] OGC 04-071, Some image geometry models

[5] OGC 04-094, Web Feature Service Implementation Specification, version 1.1.0

