

Open Geospatial Consortium Inc.

Date: 2005-01-30

Reference number of this OGC™ document: OGC 05-013

Version: 0.3.0

Category: OGC™ Discussion Paper

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

Web Coordinate Transformation Service (WCTS) draft Implementation Specification

Copyright © Open Geospatial Consortium (2005)

Warning

This draft document is not an OGC Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.

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

Document type:	OGC™ Discussion Paper
Document subtype:	Draft future RFC
Document stage:	Approved Discussion Paper
Document language:	English

Contents	Page
i. Preface.....	viii
ii. Submitting organizations	viii
iii. Document contributor contact points.....	ix
iv. Revision history	ix
v. Changes to the OGC Abstract Specification.....	x
vi. Future work.....	x
Foreword.....	xiii
Introduction.....	xiv
1 Scope.....	1
2 Conformance.....	2
3 Normative references	2
4 Terms and definitions	3
5 Conventions	3
5.1 Abbreviated terms	3
5.2 UML notation.....	3
5.3 Document terms and definitions	4
6 WCTS overview.....	4
7 Shared aspects.....	6
7.1 Introduction.....	6
7.2 Shared operation request parameters	6
7.3 Standard operation exceptions	7
7.4 Operation request encoding	8
8 GetCapabilities operation (required).....	8
8.1 Introduction.....	8
8.2 Operation request	9
8.3 Operation response.....	10
8.3.1 Normal response	10
8.3.2 OperationsMetadata section standard contents	10
8.3.3 Contents section contents.....	11
8.3.4 Capabilities document XML encoding	14
8.3.5 Exceptions.....	16
8.4 Examples.....	17
8.4.1 GetCapabilities request	17
8.4.2 GetCapabilities response.....	17

9	Transform operation (required).....	20
9.1	Introduction.....	20
9.2	Operation request.....	20
9.2.1	Transform request parameters.....	20
9.2.2	Transform request KVP encoding (optional).....	23
9.2.3	Transform request XML encoding (required).....	24
9.3	Operation response.....	31
9.3.1	Transform normal response	31
9.3.2	Exceptions.....	33
9.4	Examples.....	34
9.4.1	GML feature collection transformation	34
9.4.2	Image coordinate transformation	37
10	IsTransformable operation (optional)	38
10.1	Introduction.....	38
10.2	Operation request.....	38
10.2.1	IsTransformable request parameters	38
10.2.2	IsTransformable request KVP encoding (optional)	40
10.2.3	IsTransformable request XML encoding (required)	41
10.3	Operation response.....	46
10.3.1	IsTransformable normal response.....	46
10.3.2	Exceptions.....	49
10.4	Examples.....	49
10.4.1	IsTransformable request.....	49
10.4.2	IsTransformable response	49
11	GetTransformation operation (optional)	50
11.1	Introduction.....	50
11.2	Operation request.....	50
11.2.1	GetTransformation request parameters.....	50
11.2.2	GetTransformation request KVP encoding (required).....	50
11.2.3	GetTransformation request XML encoding (optional)	51
11.3	Operation response.....	52
11.3.1	GetTransformation normal response.....	52
11.3.2	Exceptions.....	55
11.4	Examples.....	55
11.4.1	GetTransformation request	55
11.4.2	GetTransformation response	56
12	DescribeTransformation Operation (optional).....	57
12.1	Introduction.....	57
12.2	Operation request.....	57
12.2.1	DescribeTransformation request parameters	57
12.2.2	DescribeTransformation request KVP encoding (required)	58
12.2.3	DescribeTransformation request XML encoding (optional).....	58
12.3	Operation response.....	60
12.3.1	DescribeTransformation normal response	60
12.3.2	Exceptions.....	62
12.4	Examples.....	62

12.4.1	DescribeTransformation request	62
12.4.2	DescribeTransformation response	63
13	DescribeCRS operation (optional)	64
13.1	Introduction	64
13.2	Operation request	64
13.2.1	DescribeCRS request parameters	64
13.2.2	DescribeCRS request KVP encoding (required)	65
13.2.3	DescribeCRS request XML encoding (optional)	65
13.3	Operation response	67
13.3.1	DescribeCRS normal response	67
13.3.2	DescribeCRS exceptions	68
13.4	Examples	68
13.4.1	DescribeCRS request	68
13.4.2	DescribeCRS response	68
14	DescribeMethod operation (optional)	70
14.1	Introduction	70
14.2	Operation request	70
14.2.1	DescribeMethod request parameters	70
14.2.2	DescribeMethod request KVP encoding (required)	71
14.2.3	DescribeMethod request XML encoding (optional)	71
14.3	Operation response	73
14.3.1	DescribeMethod normal response	73
14.3.2	GML Application Schema	74
14.3.3	DescribeMethod exceptions	75
14.4	Examples	75
14.4.1	Transverse Mercator example	75
14.4.2	User defined orthorectification method example	77
Annex A	(normative) Abstract test suite	80
A.1	Conformance testing overview	80
A.2	Documentation review	80
A.3	GetCapabilities operation testing	81
A.4	Transform operation testing	82
A.5	IsTransformable operation testing	82
A.6	GetTransformation operation testing	82
A.7	DescribeTransformation operation testing	83
A.8	DescribeCRS operation testing	83
A.9	DescribeMethod operation testing	84
Annex B	(normative) XML schemas	85
Annex C	(informative) Use cases	87
C.1	Introduction	87
C.2	Desktop application	87
C.3	WMS	88
C.4	Service chain	89
C.5	Defining and using user-defined coordinate transformation	90

Annex D (informative) Example XML documents	92
D.1 Introduction.....	92
D.2 TBD.....	92
Annex E (informative) UML model	93
E.1 Introduction.....	93
E.2 UML packages	94
E.3 WCTS Service package	96
E.4 WCTS Transform package	98
E.5 WCTS Is Transformable package.....	99
E.6 WCTS Get Transformation package.....	100
E.7 WCTS Describe Transformation package	101
E.8 WCTS Describe CRS package.....	102
E.9 WCTS Describe Method package.....	103
E.10 WCTS Get Capabilities package	104
E.11 WCTS Contents package	105
E.12 OWS Additions package.....	106
E.13 WCTS Operation Method package.....	106
E.14 Domain package.....	108
Annex F (informative) GML application schemas for features.....	109
F.1 Introduction.....	109
F.2 Simple example GML application schemas for features	109
F.3 GML application schemas for pseudo-features	110
Annex G (informative) Design decisions	112
G.1 Introduction.....	112
G.2 Included interface abilities.....	112
G.3 Excluded interface abilities.....	113
G.4 Implementation of interface abilities	114
G.5 Specification approaches	114
Bibliography	116

Figures	Page
Figure 1 — WCTS interface UML diagram.....	6
Figure C.1 — Desktop Application use case: application - WCTS interaction	87
Figure C.2 — WMS use case: client - WMS - WCTS interaction.....	88
Figure C.3 — Service Chain use case: client - WMS - WFS - WCTS interaction.....	89
Figure C.4 — Service Chain use case two4: service chain	90
Figure E.1 — WCTS interface UML diagram	93
Figure E.2 — WCTS interface package diagram.....	95
Figure E.3 — WCTS Service package class diagram	97
Figure E.4 — WCTS Transform package class diagram	98
Figure E.5 — WCTS Is Transformable package class diagram.....	99
Figure E.6 — WCTS Describe Transformation package class diagram	100
Figure E.7 — WCTS Describe Transformation package class diagram	101
Figure E.8 — WCTS Describe CRS package class diagram	102
Figure E.9 — WCTS Describe Method package class diagram.....	103
Figure E.10 — WCTS Get Capabilities package class diagram	104
Figure E.11 — WCTS Contents package class diagram.....	105
Figure E.12 — OWS Additions package class diagram.....	106
Figure E.13 — WCTS Operation Method package class diagram.....	107
Figure E.14 — Domain package class diagram	108

Tables	Page
Table 1 — SourceAndTargetCRSs parameter group	7
Table 2 — Transformation parameter group.....	7
Table 3 — Exception codes for WCTS optional operations	8
Table 4 — Operation request encoding.....	8
Table 5 — Section name values and contents	10
Table 6 — Required values of OperationMetadata section attributes	11
Table 7 — Optional values of OperationMetadata section attributes.....	11
Table 8 — Parameters included in Contents section	12
Table 9 — Parameters included in FeatureAbilities subsection.....	13
Table 10 — Parameters included in CoverageAbilities subsection.....	14

Table 11 — Parameters included in InputOutputFormat subsection.....	14
Table 12 — Parameters in Transform operation request.....	21
Table 13 — Parameters in Reference data structure	23
Table 14 — Transform operation request URL parameters	23
Table 15 — Parts of Transform operation response.....	31
Table 16 — Exception codes for Transform operation	33
Table 17 — Parameters in IsTransformable operation request	39
Table 18 — Alternatives in IsTransformableChoice subsection.....	40
Table 19 — IsTransformable operation request URL parameters.....	41
Table 20 — Problem values for IsTransformable operation	46
Table 21 — Parameters in IsTransformable operation response.....	47
Table 22 — Parameters in GetTransformation operation request.....	50
Table 23 — GetTransformation operation request URL parameters	51
Table 24 — Parts of GetTransformation operation response	53
Table 25 — Parameters in DescribeTransformation operation request.....	58
Table 26 — DescribeTransformation operation request URL parameters	58
Table 27 — Parts of DescribeTransformation operation response.....	60
Table 28 — Parameters in DescribeCRS operation request.....	65
Table 29 — DescribeCRS operation request URL parameters	65
Table 30 — Parts of DescribeCRS operation response.....	67
Table 31 — Parameters in DescribeMethod operation request.....	71
Table 32 — DescribeMethod operation request URL parameters	71
Table 33 — Parts of DescribeMethod operation response	73

i. Preface

NOTE The previous version of this document was numbered 04-072r3, now renumbered 05-013 after approval and final editing in 2005.

This draft document includes a draft abstract test suite plus numerous small improvements and corrections from the previous version of this draft specification for the interface to a Web Coordinate Transformation Service (WCTS). This draft Implementation Specification that builds on an Interoperability Program Report from the OGC Web Services phase 2 (OWS-2) interoperability initiative. That Interoperability Program Report built on a previous draft RFC which build upon OGC Discussion Paper 02-061r1.

All these documents 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 04-046r3], including both coordinate “Transformations” and “Conversions”.

To use together data stored in different CRSs, such data must be transformed or converted into the same CRS. At present, each OGC service has to perform its own coordinate transformations, or access a service using the interfaces specified in the “Coordinate Transformation Services Implementation Specification” [OGC 01-009].

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. 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 (?)

iii. Document contributor contact points

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

Contact	Company	Address	Phone	Email
Arliss Whiteside	BAE SYSTEMS National Security Solutions	10920 Technology Dr. San Diego, CA 92127-1608 United States	+1 858-592-1608	Arliss.Whiteside@bae systems.com
Stephane Fellah	PCI Geomatics	490, Boulevard St Joseph Suite 400 Hull, QC		
Markus U. Müller	lat/lon	Meckenheimer Allee 176, 53115 Bonn, Germany	+49 +177 2470742	mueller@lat-lon.de
Andreas Poth	lat/lon	Meckenheimer Allee 176, 53115 Bonn, Germany	+49 +228 73 9666	poth@lat-lon.de
Frank Warmerdam	DM Solutions Group			warmerdam@pobox. com
Paul Daisey	U.S. Census			pdaisey@geo.census. gov
Jeffrey Simon	General Dynamics AIS			Jeffrey.simon@usigs. com
George Percivall	NASA / GST, Inc.	NASA Goddard Space Flight Center, Code 101.4	1+301-286-4073	percivall@gsfc.nasa. gov

iv. 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.

Date	Release	Editor	Primary clauses modified	Description
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
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

v. Changes to the OGC Abstract Specification

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

vi. 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 when it is completed. 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 information will be included in the contents of a complete Capabilities document returned by a WCTS server? What additional information? Remove what current information?
- d) What standard srsName contents should be defined (in the CRS Schemas Application Profile or WCTS)?
- e) 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.)
- f) What are the GML geometric primitive types that should be transformed by a WCTS server?
- g) 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?
- h) What abilities will be included in the “limited” WCTS profile? (The questions in this list that are answered with non-minimum abilities might have lesser abilities defined as required in the limited WCTS profile.)
- i) What standard GML Application Schema(s) should be specified for use in transforming one or more point positions?

EDITOR'S NOTE I urge reviewers of this draft to point out any open questions I failed to include in the above list. Also, please point out any listed and other questions that you think have been decided.

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

EDITOR'S NOTE I urge reviewers of this draft to point out other possibly-omitted abilities which I failed to include in the above list.

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.

EDITOR'S NOTE I also ask reviewers of this draft to point out other possible future expansions which I failed to include in the above list.

Foreword

This revision supersedes OGC document 04-072 and OGC Discussion Paper 02-061r1, titled Web Coordinate Transformation Service draft Implementation Specification. This specification supplements, and does not supersede, OGC document 00-009 titled OpenGIS Implementation Specification: Coordinate Transformation Services. This version of this specification supersedes all previous versions.

This WCTS interface uses parts 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 WCTS interface also uses parts of the recommended XML encoding of coordinate reference system definitions, now specified in [OGC 04-092r4]. Those coordinate reference system definitions are largely based on OGC Abstract Specification Topic 2 [OGC 04-046r3], which is largely based on ISO 19111:2003.

This specification includes six annexes; Annexes A and B are normative and the rest 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.

Introduction

This document specifies the interface to a Web Coordinate Transformation Service (WCTS), which can be used by geospatial applications and other services. 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 or converted into the same CRS. Not all applications or services are capable of directly 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.

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]. The geometries in that document 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 [OGC 04-092r4]. Those coordinate reference system definitions are largely based on OGC Abstract Specification Topic 2 [OGC 04-046r3], which is largely based on ISO 19111:2003.

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 OGC[®] interface standard, a software implementation shall choose to implement any one of the conformance levels specified in Annex A (normative). (TBR)

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: *EPSG Geodesy Parameters V 6.3*, available through EPSG: <http://www.epsg.org/>

OGC 03-065r6, *Web Coverage Service (WCS)*, Version 1.0.0, August 2003

OGC 04-046r3, *The OpenGIS[®] Abstract Specification Topic 2: Spatial referencing by coordinates*, August 2004

OGC 04-092r4, *GML 3.1.1 Schema documents (bugfix release)*

OGC 05-008, *OGC Web Services Common Specification*

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 04-046r3] and in OGC Web Services Common Specification [OGC 05-008] 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 05-008] 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

Most 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 05-008].

5.3 Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 05-008].

6 WCTS overview

The specified 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 interface thus has similarities to the Web Map Service (WMS), Web Feature Service (WFS), and Web Coverage Service (WCS) interfaces.

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

- a) GetCapabilities (required implementation by servers) – This operation allows a client to request and receive back service metadata (or Capabilities) documents that describe the abilities of the specific server implementation. This operation also supports negotiation of the specification version being used for client-server interactions.
- b) 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.
- 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 no.

- d) `GetTransformation` (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 CRS Schemas [OGC 04-092r4].
- e) `DescribeTransformation` (optional implementation by servers) – This operation allows a client to request and receive back the definition of one or more identified coordinate transformations. Each of these transformations 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 CRS Schemas.
- f) `DescribeCRS` (optional implementation by servers) – This operation allows a client to request and receive back the definition of one or more identified coordinate reference systems (CRSs). Each of these CRSs can be any one of the nine different concrete types of CRS. Each CRS definition is encoded in XML using the GML 3.1.1 CRS Schemas.
- g) `DescribeMethod` (optional implementation by servers) – This operation allows a client to request and receive back the definition of one or more identified operation methods. Each operation method definition is encoded in XML using a specified GML Application Schema that builds on the GML 3.1.1 CRS Schemas.

The `GetCapabilities` operation has many similarities to the `GetCapabilities` operation in other OGC Web Services, including the WMS, WFS, and WCS. The other six 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 05-008]. 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` operation from the `OGCWebService` interface class, and adds the “`transform`”, `isTransformable`, `getTransformation`, `describeTransformation`, `describeCRS`, and `describeMethod` 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).

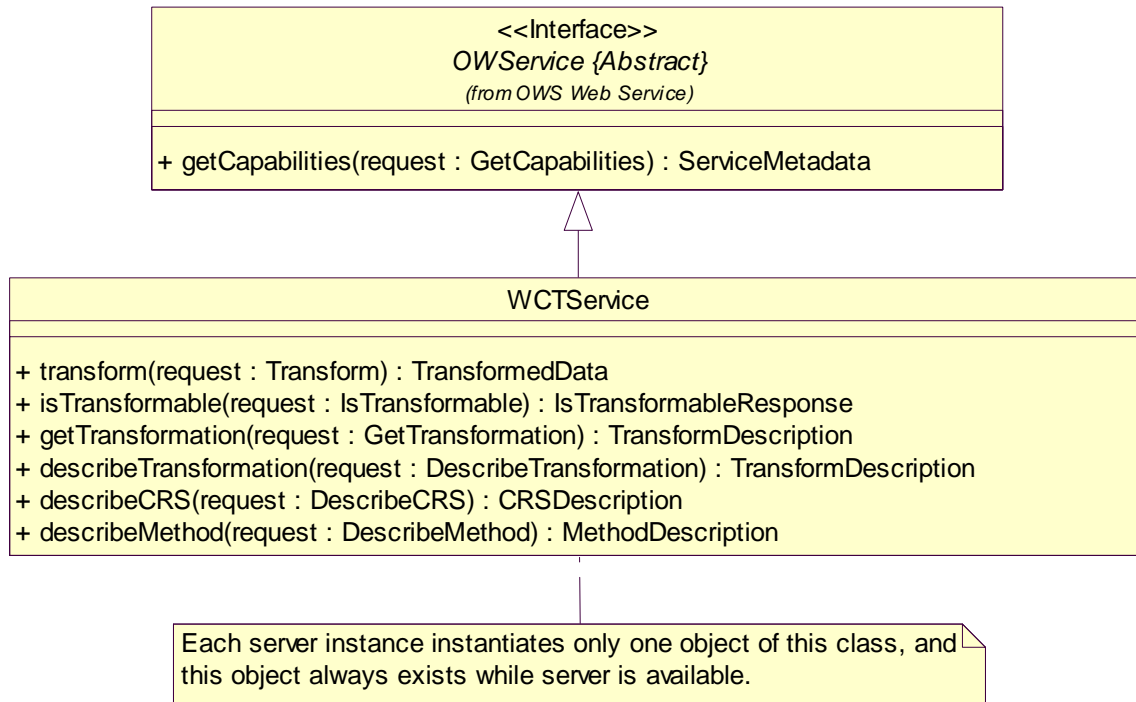


Figure 1 — WCTS interface UML diagram

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 seven 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 operation request parameters

This clause specifies some parameters used by multiple operation requests and responses specified in the following clauses. The parameter names, meanings, and formats shall be as specified below.

Table 1 specifies a group of two parameters named SourceAndTargetCRSs that is used by multiple operation requests.

Table 1 — SourceAndTargetCRSs parameter group

Name	Definition	Data type and value	Multiplicity
SourceCRS	Identifier of input coordinate reference system	URI ^a Values defined in service metadata or in data known to client	One (mandatory)
TargetCRS	Identifier of 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 05-008].			

Table 2 specifies a group of parameters named Transformation that is used by multiple operation requests and responses.

Table 2 — Transformation parameter group

Name	Definition	Data type and value	Multiplicity and use
Coordinate Operation	Identifier or definition of coordinate operation	URI ^a or CC_CoordinateOperation ^b Values defined in service metadata or known to client	One (mandatory)
SourceCRS	Identifier 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	Identifier 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
<p>a A URI shall reference a CRS or coordinate operation as specified in Subclause 10.3 of [OGC 05-008].</p> <p>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.</p> <p>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.</p>			

7.3 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 05-008]. For all optional WCTS operations, the allowed standard exception codes shall include those listed in Table 10. For each listed exceptionCode, the contents of the “locator” parameter value shall be as specified in the right column of Table 3.

NOTE To reduce the need for readers to refer to other documents, all four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 3 — 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

7.4 Operation request encoding

The encoding of operation requests shall use HTTP GET with KVP encoding and HTTP POST with XML encoding as specified in Clause 11 of [OGC 05-008]. Table 4 summarizes the WCTS operations and their encoding methods defined in this specification.

Table 4 — Operation request encoding

Operation	Request encoding
GetCapabilities (required)	KVP and optional XML
Transform (required)	XML and optional KVP
IsTransformable (optional)	XML and optional KVP
GetTransformation (optional)	KVP and optional XML
DescribeTransformation (optional)	KVP and optional XML
DescribeCRS (optional)	KVP and optional XML
DescribeMethod (optional)	KVP and optional XML

The KVP values may include XML elements. For example, this is the case when encoding a Transform operation request with an in-line feature. The geospatial data that shall be transformed may be encoded as GML. If so, the geospatial data must be GML 3.1.1 encoded to use it within a HTTP-GET request.

8 GetCapabilities operation (required)

8.1 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 specific information about its transformation abilities. This clause specifies the XML document that a web coordinate transformation server must return to describe its capabilities.

8.2 Operation request

The GetCapabilities operation request shall be as specified in Subclauses 7.2.2 through 7.2.4 of [OGC 05-008]. The “service”, “request”, and “AcceptVersions” parameters shall be implemented by all WCTS servers. The “Sections” and “updateSequence” parameters are optional implementation by WCTS servers. All WCTS servers shall implement HTTP GET transfer of the GetCapabilities operation request, using KVP encoding. Servers may also implement HTTP POST transfer of the GetCapabilities operation request, using XML encoding only.

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 05-008].

The XML Schema fragment for encoding a WCTS GetCapabilities operation request extends the ows:GetCapabilitiesType in owsCommon.xsd of [OGC 05-008], and is:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:wcts="http://www.opengeospatial.net/wcts"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengeospatial.net/wcts"
elementFormDefault="qualified" xml:lang="en">
  <annotation>
    <appinfo>fragmentGetCapabilitiesRequest.xsd 2004-12-20</appinfo>
    <documentation>This XML Schema encodes the WCTS GetCapabilities
operation request message. </documentation>
  </annotation>
  <!-- =====
includes and imports
===== -->
  <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="../../../ows/0.4.0/owsGetCapabilities.xsd"/>
  <!-- =====
elements and types
===== -->
  <element name="GetCapabilities">
    <annotation>
      <documentation>Request to a WCTS to perform the
GetCapabilities operation. This operation allows a client to retrieve a
Capabilities XML document providing metadata for the specific WCTS
server. In this XML encoding, no "request" parameter is included, since
the element name specifies the specific operation. </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base="ows:GetCapabilitiesType">
          <sequence/>
          <attribute name="service" type="ows:ServiceType"
use="required" fixed="WCTS"/>
        </extension>
      </complexContent>
    </complexType>
  </element>
</schema>
```

GetCapabilities operation request examples are very similar to those listed in Subclauses 7.2.2 and 7.2.3 of [OGC 05-008].

8.3 Operation response

8.3.1 Normal response

The complete service metadata document shall contain the four sections specified in Table 5. Depending on the values in the Sections parameter of the GetCapabilities operation request, any combination of these sections can be requested and returned.

Table 5 — 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 05-008].
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 05-008].
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 05-008].
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 8.3.3 and 8.3.4 below.

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

In addition to these sections, each service metadata document shall include the mandatory “version” and optional updateSequence parameters specified in Table 6 in Subclause 7.4.1 of [OGC 05-008].

8.3.2 OperationsMetadata section standard contents

For the WCTS, the OperationsMetadata section shall be the same as for all OGC Web Services, as specified in Subclause 7.4.5 and owsOperationsMetadata.xsd of [OGC 05-008]. The mandatory values of various (XML) attributes shall be as specified in Table 6. Similarly, the items listed in Table 7 shall be included when these operations are implemented. In Table 6 and Table 7, the “Attribute name” column uses dot-separator notation to identify parts of a parent item.

Table 6 — 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 7 — 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.
	DescribeTransformation	The DescribeTransformation operation is implemented by this server.
	DescribeCRS	The DescribeCRS operation is implemented by this server.
	DescribeMethod	The DescribeMethod operation is implemented by this server.

In addition to the optional values listed in Table 7, 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 1 The domain of the exceptionCode parameter could record all the codes implemented for each operation by that specific server. Similarly, each of the GetCapabilities operation optional request parameters might have its domain recorded.

EXAMPLE 2 The domain of the “store” parameter in the Transform operation request could record the allowed values for that parameter, and thus indicate if the “store” parameter is implemented by this server.

8.3.3 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 specified in Table 8.

Table 8 — Parameters included in Contents section

Name	Definition and data type	Multiplicity and use
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	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	Identifier of 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	Identifier of 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
CoverageAbilities	Information about the coverage handling abilities of this WCTS server See Table 10	Zero or one (optional) Include when server can transform coverages
FeatureAbilities	Information about the feature handling abilities of this WCTS server See Table 9	Zero or one (optional) Include when server can transform features
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	Specifies if this server supports user defined Coordinate Reference Systems (CRSs) Boolean type, true or false	One (mandatory)
<p>^a A URI shall reference a Transformation, Method, or CRS as specified in Subclause 10.3 of [OGC 05-008].</p> <p>^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.</p> <p>^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.</p>		

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 04-046r3].

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 Transformations, SourceCRSs, and TargetCRSs parameters.

The operation Methods refer to general methods for transformation of coordinates from one CRS into another, and not the 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 subsection contents are specified in Table 9. 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.

Table 9 — Parameters included in FeatureAbilities subsection

Name	Definition and data type	Multiplicity and use
GeometryType	Name of GML 3 geometric primitive type supported by this WCTS server ^a Character String type, not empty	One or more (mandatory) One for each primitive type supported by this server
FeatureFormat	Identifier of feature format in which the Transform operation can input and output features InputOutputFormat, see Table 11	One or more (optional) One for each feature format supported by this server
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 subsection contents are specified in Table 10. The CoverageType parameters identify the types of coverages the WCTS is able to transform. In the case of 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 InterpolationMethod 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 10 — Parameters included in CoverageAbilities subsection

Name	Definition and data type	Multiplicity and use
CoverageType	Name of coverage type supported by this WCTS server Character String type, not empty	One or more (mandatory) One for each coverage type supported by this server
CoverageFormat	Identifier of coverage format in which Transform operation can input and output coverages InputOutputFormat, see Table 11	One or more (optional) One for each coverage format supported by this server
Interpolation Method	Identifier of interpolation method, as defined in Annex B of ISO 19123 – Schema for coverage geometry and functions Enumeration, of allowed values	One or more (optional) One for each interpolation method supported by this server

Table 11 — Parameters included in InputOutputFormat subsection

Name	Definition and data type	Multiplicity and use
Format	Identifier of format type supported by this WCTS server Character String type, not empty	One (mandatory)
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	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.

8.3.4 Capabilities document XML encoding

A XML Schema fragment for a WCTS service metadata document extends the ows:CapabilitiesBaseType in owsGetCapabilities.xsd in [OGC 05-008], and is:

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:wcts="http://www.opengeospatial.net/wcts"
xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengeospatial.net/wcts"
elementFormDefault="qualified" xml:lang="en">
  <annotation>
    <appinfo>fragmentGetCapabilitiesResponse.xsd 2004/12/20</appinfo>
    <documentation>
      <description>This XML Schema encodes the WCTS GetCapabilities
operation response message. </description>
    </documentation>
  </annotation>
  <!-- =====
includes and imports
===== -->
  <include schemaLocation="wctsContents.xsd"/>
  <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="../ows/0.4.0/owsGetCapabilities.xsd"/>
  <!-- =====
elements and types
===== -->
  <element name="Capabilities">
    <annotation>
      <documentation>XML encoded WCTS GetCapabilities operation
response. This document provides clients with service metadata about a
specific service instance, including metadata about the tightly-coupled
data served. If the server does not implement the updateSequence
parameter, the server shall always return the complete Capabilities
document, without the updateSequence parameter. When the server
implements the updateSequence parameter and the GetCapabilities
operation request included the updateSequence parameter with the
current value, the server shall return this element with only the
"version" and "updateSequence" attributes. Otherwise, all optional
elements shall be included or not depending on the actual value of the
Contents parameter in the GetCapabilities operation request.
</documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base="ows:CapabilitiesBaseType">
          <sequence>
            <element ref="wcts:Contents" minOccurs="0"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
</schema>
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:wcts="http://www.opengeospatial.net/wcts"
xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengeospatial.net/wcts"
elementFormDefault="qualified" xml:lang="en">
  <annotation>
    <appinfo>fragmentGetCapabilitiesResponse.xsd 2004/12/20</appinfo>
    <documentation>

```

```

    <description>This XML Schema encodes the WCTS GetCapabilities
operation response message. </description>
  </documentation>
</annotation>
<!-- =====
includes and imports
===== -->
<include schemaLocation="wctsContents.xsd"/>
<import namespace="http://www.opengeospatial.net/ows"
schemaLocation=" ../ows/0.4.0/owsGetCapabilities.xsd"/>
<!-- =====
elements and types
===== -->
<element name="Capabilities">
  <annotation>
    <documentation>XML encoded WCTS GetCapabilities operation
response. This document provides clients with service metadata about a
specific service instance, including metadata about the tightly-coupled
data served. If the server does not implement the updateSequence
parameter, the server shall always return the complete Capabilities
document, without the updateSequence parameter. When the server
implements the updateSequence parameter and the GetCapabilities
operation request included the updateSequence parameter with the
current value, the server shall return this element with only the
"version" and "updateSequence" attributes. Otherwise, all optional
elements shall be included or not depending on the actual value of the
Contents parameter in the GetCapabilities operation request.
</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="ows:CapabilitiesBaseType">
        <sequence>
          <element ref="wcts:Contents" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
</schema>

```

As indicated, this XML Schema uses the owsGetCapabilities.xsd schema specified in [OGC 05-008], plus the wctsContents.xsd schema attached to this document.

8.3.5 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 05-008]. The allowed exception codes shall include those listed in Table 5 of [OGC 05-008], if the updateSequence parameter is implemented by the server.

8.4 Examples

8.4.1 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.opengeospatial.net/wcts"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsGetCapabilities.xsd" service="WCTS"/>
<!-- Primary editor: Arliss Whiteside. Last updated 2004/11/05 -->
```

A maximum request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities xmlns="http://www.opengeospatial.net/wcts"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ows="http://www.opengeospatial.net/ows"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsGetCapabilities.xsd" service="WCTS"
updateSequence="XYZ123">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/11/05 -->
  <ows:AcceptVersions>
    <ows:Version>0.0.0</ows:Version>
  </ows:AcceptVersions>
  <ows:Sections>
    <ows:Section>All</ows:Section>
  </ows:Sections>
</GetCapabilities>
```

8.4.2 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.opengeospatial.net/wcts"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsGetCapabilities.xsd http://www.opengeospatial.net/ows
../Schemas/owsAdditions.xsd" version="0.0.0">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/20-->
  <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:WWWW</Transformation>
>

    <Transformation>urn:ogc:def:coordinateOperation:XXXX</Transformation>
>

    <Transformation>urn:ogc:def:coordinateOperation:YYYY</Transformation>
>

    <Transformation>urn:ogc:def:coordinateOperation:ZZZZ</Transformation>
>

    <Method>urn:ogc:def:method:9801</Method>
    <Method>urn:ogc:def:method:9802</Method>
    <Method>urn:ogc:def:method:9803</Method>
    <SourceCRS>urn:ogc:def:crs:4326</SourceCRS>
    <SourceCRS>urn:ogc:def:crs:23031</SourceCRS>
    <SourceCRS>urn:ogc:def:crs:23032</SourceCRS>
    <SourceCRS>urn:ogc:def:crs:31467</SourceCRS>
    <TargetCRS>urn:ogc:def:crs:4326</TargetCRS>
    <TargetCRS>urn:ogc:def:crs:EPSG:6.3:23031</TargetCRS>
    <TargetCRS>urn:ogc:def:crs:23032</TargetCRS>
    <TargetCRS>urn:ogc:def:crs:31467</TargetCRS>
    <CoverageAbilities>
        <CoverageType>GridCoverage</CoverageType>
        <CoverageFormat>image/pix</CoverageFormat>
        <CoverageFormat>image/HDF-EOS</CoverageFormat>
        <CoverageFormat>image/DTED</CoverageFormat>
        <CoverageFormat>image/GeoTIFF</CoverageFormat>
        <CoverageFormat>image/NITF</CoverageFormat>
        <ows:InterpolationMethod>nearest
neighbor</ows:InterpolationMethod>
        <ows:InterpolationMethod>bilinear</ows:InterpolationMethod>
        <ows:InterpolationMethod>bicubic</ows:InterpolationMethod>
    </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.

9 Transform operation (required)

9.1 Introduction

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

9.2 Operation request

9.2.1 Transform request parameters

A request to perform the Transform operation shall include the parameters listed and defined in Table 12. This table also specifies the UML model data type, source of values, and multiplicity of each listed parameter, plus the meaning to servers when each optional parameter is not included in the operation request.

NOTE The first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 05-008].

Table 12 — Parameters in Transform operation request

Name	Definition	Data type and value	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, namely “WCTS”	One (mandatory)
request	Operation name	Character String type, not empty Value is operation name, namely “Transform”	One (mandatory)
version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
SourceAndTargetCRSs	Identifiers of SourceCRS and desired TargetCRS	See Table 1	Zero or one (Mutually exclusive with Transformation) ^a
Transformation	Identification or definition of desired coordinate operation	See Table 2	Zero or one (Mutually exclusive with SourceCRS AND TargetCRS) ^b
InputData	Data to be transformed	Either gml:_Feature OR Reference, see Table 13	One or more (mandatory)
Interpolation Method ^c	Identifier of interpolation method which should be used to transform coverage	Character string type, not empty Values defined in service metadata	Zero or one (optional) Included when transforming coverage using interpolation
OutputFormat	Identifier of output format to be used for the transformed features or coverage ^d	Character String type, not empty	Zero or one (optional) Included when output data not in operation response (TBR)
store	Specifies if transformed data to be stored as remote resource or returned directly in response message	Boolean Values are: true or false	Zero or one (optional) Default is false (return directly in response)

- 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 definition object 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
- c 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 rectified 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.
- d Allows advanced 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 The Transformations (or coordinate operations) implemented by current 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.

EDITOR'S NOTE It is not clear if allowing an inline gml:Feature is the best approach, since a GML document referenced as a local payload may be easier to implement using a standard GML parser.

The InputData can be either an inline gml:Feature or a "Reference" to geospatial data supported by the WCTS. The reference is used to address a local payload or a remote resource. The reference can have a human readable description and a link to metadata. The gml:remoteSchema attribute in gml:AssociationAttributeGroup is used to indicate the schema of the remote resource (such as a GML Application Schema). The Grid is used to indicate the Grid Geometry of an image format that does not store this information (PNG, JPEG). This is useful for image format that does not support georeferencing information.

Table 13 — Parameters in Reference data structure

Name	Definition	Data type and value	Multiplicity and use
address	Reference to data in remote resource or local payload ^a	gml:Association-AttributeGroup	One (mandatory)
Abstract	Brief narrative description of referenced data, available for display to a human	Character String type, not empty	Zero or one (optional) Included when useful
Format	Format of referenced data	Character String type, not empty Values defined in service metadata	Zero or one (optional) Omitted when format is indicated in header of referenced file
Metadata	Metadata about this data	Metadata contents or reference to metadata	Zero or more (optional) One for each such metadata object
RequestMessage	Reference to message to be sent as an operation request, when requesting resource from a web server using HTTP Post ^b	URL	Zero or one (optional) Included when HTTP Post operation request message required to get input data
Grid	Definition of grid in referenced data	gml:Grid	Zero or one (optional) Included when not otherwise specified in referenced data
<p>a A remote resource is typically addressed by a URL. For a local payload (multipart mime message), the xlink:href must start with the prefix cid:. The gml:remoteSchema attribute is used to indicate the schema of the remote resource (e.g., XML application schema). In this use, the xlink:href attribute shall be included.</p> <p>b This message shall be attached to the request message, or be accessible using a URL.</p>			

9.2.2 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 shall use the parameters specified in Table 14. The parameters listed in Table 14 shall be as specified in Table 12 above.

Table 14 — 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:6.3:TBD	Optional ^b	Identifier URI of desired coordinate operation
InputData=TBD	Mandatory	Data to be transformed encoded in a format supported by the WCTS, either encoded inline as GML or referenced using a URL
InterpolationMethod=bilinear	Optional	Identifier of interpolation method which should be used to transform a coverage
OutputFormat=TBD	Optional ^c	Identifier of output format to be used for the transformed features or coverage ^d
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).
<p>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 05-008].</p> <p>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:</p> <ul style="list-style-type: none"> * 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 <p>c Included when the output data is not in the operation response message (TBR).</p> <p>d Allows advanced WCTS servers to perform reformatting. The output formats supported by a WCTS are listed in the Contents section of the Capabilities document.</p>		

NOTE This KVP encoding does not include the RequestMessage parameter, because that parameter is for HTTP POST operation requests, normally using XML encoding.

9.2.3 Transform request XML encoding (required)

All WCTS servers shall implement HTTP POST transfer of the Transform operation request, using XML encoding only. The following schema fragment specifies the contents and structure of a Transform operation request encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:wcts="http://www.opengeospatial.net/wcts"
xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengeospatial.net/wcts"
elementFormDefault="qualified" xml:lang="en">
  <annotation>
    <appinfo>fragmentTransformRequest.xsd 2005-01-10</appinfo>
    <documentation>This XML Schema encodes the WCTS Transform
operation request messages. </documentation>
  </annotation>
  <!-- =====
```

```

    includes and imports
    ===== -->
    <import namespace="http://www.opengis.net/gml"
schemaLocation="../gml/3.1.1/base/coordinateReferenceSystems.xsd"/>
    <import namespace="http://www.opengis.net/gml"
schemaLocation="../gml/3.1.1/base/feature.xsd"/>
    <import namespace="http://www.opengis.net/gml"
schemaLocation="../gml/3.1.1/base/grids.xsd"/>
    <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="../ows/0.4.0/owsCommon.xsd"/>
    <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="../ows/0.4.0/ows19115subset.xsd"/>
    <!-- =====
    elements and types
    ===== -->
    <element name="Transform">
      <annotation>
        <documentation>Request to a WCTS to perform the Transform
operation. This operation transforms coordinates from one coordinate
reference system into another, where those coordinates are in geometric
primitives in GML features. The InputData are the feature(s) and/or
feature collection(s) whose geometry elements are to be transformed.
Alternately, this operation transforms a coverage from one coordinate
reference system into another. In this XML encoding, no "request"
parameter is included, since the element name specifies the specific
operation. </documentation>
      </annotation>
      <complexType>
        <sequence>
          <choice>
            <group ref="wcts:SourceAndTargetCRSs">
              <annotation>
                <documentation>Desired well-known SourceCRS and
TargetCRS, included when client is not specifying a specific coordinate
operation. </documentation>
              </annotation>
            </group>
            <element ref="wcts:transformation">
              <annotation>
                <documentation>Desired coordinate operation,
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 definition object 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 </documentation>
              </annotation>
            </element>
          </choice>
          <element ref="wcts:InputData"/>
          <element name="InterpolationMethod"
type="wcts:InterpolationMethodType" minOccurs="0">
            <annotation>

```

```

        <documentation>Identifier of interpolation method to
be used for Coverage transformation. </documentation>
    </annotation>
</element>
    <element name="OutputFormat" type="ows:MimeType"
minOccurs="0">
    <annotation>
        <documentation>Identifier of the output format to be
used for the transformed features or coverage, allowing advanced WCTS
servers to perform reformatting. The output formats supported by a WCTS
server are listed in the Contents section of the Capabilities document.
If this element is omitted, the output format shall be the same as the
format of the InputData. If that output format is not supported by a
WCTS server, an exception message shall be returned. </documentation>
    </annotation>
    </element>
</sequence>
    <attribute name="store" type="boolean" use="optional"
default="true">
    <annotation>
        <documentation>Specifies if the response transformed
data should be stored, remotely from the client at a network URL,
instead of being returned within the operation response. This attribute
should be included only if this operation parameter is supported, as
encoded in the OperationsMetadata section of the Capabilities document.
</documentation>
    </annotation>
    </attribute>
    <attribute name="service" type="string" use="required"
fixed="WCTS">
    <annotation>
        <documentation>Service type identifier. </documentation>
    </annotation>
    </attribute>
    <attribute name="version" type="string" use="required"
fixed="0.0.0">
    <annotation>
        <documentation>Specification version for OWS version and
operation. See Version parameter Subclause 7.3.1 for more information.
</documentation>
    </annotation>
    </attribute>
</complexType>
</element>
<!-- ===== -->
<group name="SourceAndTargetCRSs">
    <annotation>
        <documentation>Group combining SourceCRS and TargetCRS
elements, used by some WCTS operation requests. </documentation>
    </annotation>
    <sequence>
        <element name="SourceCRS" type="anyURI">
            <annotation>
                <documentation>The coordinate reference system (CRS)
used by coordinates input to a Transform operation. This element shall
uniquely identify the desired CRS, but the definition of that CRS need
not be known to the WCTS server. </documentation>
            </annotation>

```



```

    </element>
    <element name="TargetCRS" type="anyURI">
      <annotation>
        <documentation>The coordinate reference system (CRS)
used by coordinates output from a Transform operation. This element
shall uniquely identify the desired CRS, but the definition of that CRS
need not be known to the WCTS server. </documentation>
      </annotation>
    </element>
  </sequence>
</group>
<!-- ===== -->
<complexType name="OutputType">
  <annotation>
    <documentation>Asks for the Transform response to be encoded
in a particular format. Can also ask for the response to be stored
remotely from the client at a URL, instead of being returned in the
operation response. </documentation>
  </annotation>
  <sequence>
    <element name="Format" type="ows:MimeType">
      <annotation>
        <documentation>Identifier of output format to be used
for the transformed features or coverage, allowing advanced WCTS
servers to perform reformatting. The output formats supported by a WCTS
are listed in the Contents section of the Capabilities document.
</documentation>
      </annotation>
    </element>
  </sequence>
  <attribute name="store" type="boolean" use="optional"
default="true">
    <annotation>
      <documentation>Specifies if the response transformed data
should be stored, remotely from the client at a network URL, instead of
being returned within the operation response. This attribute should be
included only if this operation parameter is supported, as encoded in
the OperationsMetadata section of the Capabilities document.
</documentation>
    </annotation>
  </attribute>
</complexType>
<!-- ===== -->
<element name="InputData">
  <annotation>
    <documentation> Input data to a Transform operation.
</documentation>
  </annotation>
  <complexType>
    <choice>
      <element ref="gml:_Feature">
        <annotation>
          <documentation>A feature, feature collection, or
coverage to be transformed. </documentation>
        </annotation>
      </element>
      <element ref="wcts:Reference" maxOccurs="unbounded"/>
    </choice>
  </complexType>
</element>

```

```

    </complexType>
  </element>
  <!-- ===== -->
  <element name="transformation">
    <annotation>
      <documentation>A coordinate operation that can transform
coordinates from a sourceCRS to a targetCRS. This element is often a
reference to a well-known coordinate operation, but can contain the
definition of a user-defined coordinate operation that references a
well-known operation method. Alternately, this element can contain the
definition of a ConcatenatedOperation that combines two or more well-
known coordinate operations or coordinate operations that reference a
well-known operation method. When the definition of a coordinate
operation is contained, the element shall also include a xlink:href
value containing the URN that references this definition. Well-known
coordinate operations and operation methods can be defined in documents
referenced by the WCTS Implementation Specification. Alternately or in
addition, well-known coordinate operations and operation methods can be
defined in metadata referenced by or contained in the Contents section
of the Capabilities XML document available from this WCTS server.
</documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base="gml:CoordinateOperationRefType">
          <sequence minOccurs="0">
            <annotation>
              <documentation>This sourceCRS and targetCRS shall
be included only when the referenced or contained coordinate operation
is a Conversion, which does not identify its sourceCRS and targetCRS.
</documentation>
            </annotation>
            <element name="sourceCRS" type="gml:CRSRefType">
              <annotation>
                <documentation>The coordinate reference system
(CRS) used by coordinates input to a Conversion. This element shall
uniquely identify the desired CRS, but the definition of that CRS need
not be known to the WCTS server. This element is normally a reference
to that CRS, but can contain the definition of a user-defined CRS. When
this element contains the CRS definition object, it shall also contain
a xlink:href containing the URN that references this definition.
</documentation>
              </annotation>
            </element>
            <element name="targetCRS" type="gml:CRSRefType">
              <annotation>
                <documentation>The coordinate reference system
(CRS) used by coordinates output from a Conversion. This element shall
uniquely identify the desired CRS, but the definition of that CRS need
not be known to the WCTS server. This element is normally a reference
to that CRS, but can contain the definition of a user-defined CRS. When
this element contains the CRS definition object, it shall also contain
a xlink:href containing the URN that references this definition.
</documentation>
              </annotation>
            </element>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>

```

```

        </complexContent>
    </complexType>
</element>
<!-- ===== -->
<simpleType name="ServiceType">
    <annotation>
        <documentation>Service type identifier, where the string value
is the OGC Web Service type abbreviation. </documentation>
    </annotation>
    <restriction base="string"/>
</simpleType>
<!-- ===== -->
<simpleType name="VersionType">
    <annotation>
        <documentation>The version of the Implementation Specification
(document) to which the requested operation conforms. The value shall
be N.N.N, where each N is a non-negative integer
up to 99. </documentation>
    </annotation>
    <restriction base="string"/>
</simpleType>
<!-- ===== -->
<element name="Reference">
    <annotation>
        <documentation>The reference is used to address a local
payload or a remote resource. The Grid is used to indicate the Grid
Geometry of an image format that does not store georeferencing
information (PNG, JPEG). This is useful to convert maps produced by WMS
for example.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="wcts:ReferenceType">
                <sequence>
                    <element ref="gml:Grid" minOccurs="0"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
</element>
<!-- ===== -->
<complexType name="ReferenceType">
    <annotation>
        <documentation>Reference to a remote resource or a local
payload. This reference can have a human readable description and a
link to metadata. </documentation>
    </annotation>
    <sequence>
        <element ref="ows:Abstract" minOccurs="0"/>
        <element name="Format" type="ows:MimeType" minOccurs="0">
            <annotation>
                <documentation>The format of the referenced data. This
element is omitted when the mime type is indicated in the http header
of the reference. </documentation>
            </annotation>
        </element>
        <element ref="ows:Metadata" minOccurs="0"
maxOccurs="unbounded"/>

```

```

        <element name="RequestMessage" type="anyURI" minOccurs="0">
          <annotation>
            <documentation>Reference to the message to be sent as an
operation request, when requesting the resource from a web server using
HTTP Post. This message shall be attached to the same message (using
the cid scheme), or be accessible using a URL. This element shall not
be included for a local payload resource or for requesting the resource
using HTTP Get. </documentation>
          </annotation>
        </element>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup">
        <annotation>
          <documentation>Reference to remote resource or local
payload. A remote resource is typically addressed by a URL. For a local
payload (multipart mime message), the xlink:href must start with the
prefix cid:. The gml:remoteschema attribute is used to indicate the
schema of the remote resource (e.g., XML application schema). In this
use, the xlink:href attribute shall be included. </documentation>
        </annotation>
      </attributeGroup>
    </complexType>
    <!-- ===== -->
    <element name="InterpolationMethod"
type="wcts:InterpolationMethodType"/>
    <!-- ===== -->
    <simpleType name="InterpolationMethodType">
      <annotation>
        <documentation>Code that identifies an interpolation method.
The meanings of these codes are defined in Annex B of ISO 19123:
Geographic information - Schema for coverage geometry and functions.
</documentation>
      </annotation>
      <restriction base="string">
        <enumeration value="nearest neighbor"/>
        <enumeration value="bilinear"/>
        <enumeration value="bicubic"/>
        <enumeration value="lost area"/>
        <enumeration value="barycentric"/>
        <enumeration value="none">
          <annotation>
            <documentation>No interpolation. </documentation>
          </annotation>
        </enumeration>
      </restriction>
    </simpleType>
  </schema>

```

As indicated, the above XML Schema fragment uses the coordinate reference systems, features, and grids portions of GML 3.1.1 [OGC 04-092r4], where a feature may be a feature collection or a coverage (TBR).

NOTE Each gml:Feature can reference or contain zero or more metadata documents, which can describe that input feature.

9.3 Operation response

9.3.1 Transform normal response

The normal response to a Transform operation request is at least one GML 3 feature, feature collection, or coverage. More precisely, a response from the Transform operation shall include the parts listed in Table 15. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

Table 15 — Parts of Transform operation response

Name	Definition	Data type	Multiplicity and use
Feature	Transformed feature, feature collection, or coverage	<code>gml:_Feature</code>	Zero or more (Mutually exclusive with Reference)
Reference	Reference to local payload or remote resource containing transformed data	Reference (see Table 13)	Zero or more (Mutually exclusive with Feature)

The following XML Schema fragment specifies the contents and structure of a Transform operation response:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:ows="http://www.opengeospatial.net/ows"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:wcts="http://www.opengeospatial.net/wcts"
  targetNamespace="http://www.opengeospatial.net/wcts"
  elementFormDefault="qualified" xml:lang="en">
  <annotation>
    <appinfo>fragmentTransformResponse.xsd 2005-01-10</appinfo>
    <documentation>This XML Schema encodes the WCTS Transform
operation response messages. </documentation>
  </annotation>
  <!-- =====
includes and imports
===== -->
  <import namespace="http://www.opengis.net/gml"
schemaLocation="../../../gml/3.1.1/base/feature.xsd"/>
  <import namespace="http://www.opengis.net/gml"
schemaLocation="../../../gml/3.1.1/base/grids.xsd"/>
  <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="../../../ows/0.4.0/owsServiceIdentification.xsd"/>
  <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="../../../ows/0.4.0/owsCommon.xsd"/>
  <!-- =====
elements and types
===== -->
  <element name="TransformedData">
    <annotation>
      <documentation>Response to a valid Transform operation request
sent to a WCTS. </documentation>
    </annotation>
    <complexType>
```

```

    <sequence>
      <choice>
        <element ref="gml:_Feature" maxOccurs="unbounded">
          <annotation>
            <documentation>Ordered sequence of the features
and/or feature collections that were transformed, in the order in which
input data was included and/or referenced in the Transform element.
</documentation>
          </annotation>
        </element>
        <element ref="wcts:Reference" maxOccurs="unbounded">
          <annotation>
            <documentation>Ordered sequence of references to
features and/or feature collections that were transformed, in the order
in which input data was included and/or referenced in the Transform
operation request. </documentation>
          </annotation>
        </element>
      </choice>
    </sequence>
  </complexType>
</element>
<!-- ===== -->
<element name="Reference">
  <annotation>
    <documentation>The reference is used to address a local
payload or a remote resource. The Grid is used to indicate the Grid
Geometry of an image format that does not store georeferencing
information (PNG, JPEG). This is useful to convert maps produced by WMS
for example.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="wcts:ReferenceType">
        <sequence>
          <element ref="gml:Grid" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
<!-- ===== -->
<complexType name="ReferenceType">
  <annotation>
    <documentation>Reference to a remote resource or a local
payload. This reference can have a human readable description and a
link to metadata. </documentation>
  </annotation>
  <sequence>
    <element ref="ows:Abstract" minOccurs="0"/>
    <element name="Format" type="ows:MimeType" minOccurs="0">
      <annotation>
        <documentation>The format of the referenced data. This
element is omitted when the mime type is indicated in the http header
of the reference. </documentation>
      </annotation>
    </element>
  </sequence>

```

```

        <element ref="ows:Metadata" minOccurs="0"
maxOccurs="unbounded"/>
        <element name="RequestMessage" type="anyURI" minOccurs="0">
            <annotation>
                <documentation>Reference to the message to be sent as an
operation request, when requesting the resource from a web server using
HTTP Post. This message shall be attached of the same message (using
the cid scheme), or be accessible using a URL. This element shall not
be included for a local payload resource or for requesting the resource
using HTTP Get. </documentation>
            </annotation>
        </element>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup">
        <annotation>
            <documentation>Reference to remote resource or local
payload. A remote resource is typically addressed by a URL. For a local
payload (multipart mime message), the xlink:href must start with the
prefix cid:. The gml:remoteschema attribute is used to indicate the
schema of the remote resource (e.g., XML application schema). In this
use, the xlink:href attribute shall be included. </documentation>
        </annotation>
    </attributeGroup>
</complexType>
</schema>

```

As indicated, the above XML Schema uses the feature and grids portions of GML 3.1.1 [OGC 04-092r4], where a feature may be a feature collection or a coverage (TBR).

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

NOTE Each gml:_Feature can reference or contain zero or more metadata documents, which should describe that output feature and can be somewhat different than the corresponding metadata for the input feature.

9.3.2 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 05-008]. The allowed exception codes shall include those listed in Table 16. For each listed exceptionCode, the contents of the “locator” parameter value shall be as specified in the right column of Table 16.

NOTE The first three values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 16 — 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 feature data was available from the specified source for input data ^b	Value of InputData parameter
InvalidArea	One or more points in InputData are outside the domainOfValidity of the transformation	Value of InputData parameter
TransformException	Used where a computation error occurs while performing the transformation	None, omit "locator" parameter
<p>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.</p> <p>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.</p>		

9.4 Examples

9.4.1 GML feature collection transformation

9.4.1.1 Transform request

If a client wants to transform a GML FeatureCollection 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.opengeospatial.net/wcts"
  xmlns:ex="http://www.opengeospatial.net/examples"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengeospatial.net/wcts
    ../Schemas/wctsTransform.xsd http://www.opengeospatial.net/examples
    exampleCapital.xsd" service="WCTS" version="0.0.0">
  <!-- Primary editor: Arliss Whiteside. Last updated 2005-01-10-->
  <SourceCRS>urn:ogc:def:crs:EPSG:6.0:4326</SourceCRS>
  <TargetCRS>urn:ogc:def:crs:EPSG:6.0:32611</TargetCRS>
  <InputData>
    <gml:FeatureCollection>
      <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>
        <ex:Capital gml:id="AC">
          <ex:cityName>Aguascalientes</ex:cityName>
          <ex:stateName>AGUASCALIENTES</ex: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>
  </ex:Capital>
</gml:featureMember>
</gml:FeatureCollection>
</InputData>
<OutputFormat>text/xml; gmlVersion=3.1.0</OutputFormat>
</Transform>

```

The above example requests coordinate transformation between two well-known CRSs, each referenced using a URI. 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 `<transformation>` element in the example in Subclauses 11.4.2 and 12.4.2 is an example of such a specification.

The above example includes a feature encoded in-line using the simple example GML 3.1.1 Application Schema provided in Subclause F.2. If this same feature were not encoded in-line, it would be XML encoded as:

```

<?xml version="1.0" encoding="UTF-8"?>
<Transform xmlns="http://www.opengeospatial.net/wcts"
  xmlns:ows="http://www.opengeospatial.net/ows"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengeospatial.net/wcts
    ../Schemas/wctsTransform.xsd" service="WCTS" version="0.0.0">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/20 -->
  <SourceCRS>urn:ogc:def:crs:EPSG:6.0:4326</SourceCRS>
  <TargetCRS>urn:ogc:def:crs:EPSG:6.0:32611</TargetCRS>
  <InputData>
    <Reference xlink:href="TBD">
      <ows:Abstract>A state capital</ows:Abstract>
      <ows:Format>text/xml; gmlVersion=3.1.1</ows:Format>
    </Reference>
  </InputData>
  <OutputFormat>text/xml; gmlVersion=3.1.1</OutputFormat>
</Transform>

```

The corresponding XML encoded Transform operation request would then be:

```

<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns="http://www.opengeospatial.net/examples"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengeospatial.net/examples
    exampleCapital.xsd">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/20-->
  <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>

```

where “TBD” must be replaced a URL reference to the XML encoded feature.

9.4.1.2 Transform response

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

```

<?xml version="1.0" encoding="UTF-8"?>
<TransformedData xmlns="http://www.opengeospatial.net/wcts"
  xmlns:ex="http://www.opengeospatial.net/examples"
  xmlns:gml="http://www.opengis.net/gml"
  xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance"
  .. /Schemas/wctsTransform.xsd http://www.opengeospatial.net/examples
  exampleCapital.xsd">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/20-->
  <gml:FeatureCollection>
    <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>
      <ex:Capital gml:id="AC">
        <ex:cityName>Aguascalientes</ex:cityName>
        <ex:stateName>AGUASCALIENTES</ex: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>
      </ex:Capital>
    </gml:featureMember>
  </gml:FeatureCollection>
</TransformedData>

```

The above example includes a feature encoded in-line using the same simple example GML 3.1.1 Application Schema stated in Subclause F.2. If this same feature were not encoded in-line, it would be XML encoded as:

```
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns="http://www.opengeospatial.net/examples"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/examples
exampleCapital.xsd">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/20-->
  <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>
```

The corresponding XML encoded Transform operation response would then be:

```
<?xml version="1.0" encoding="UTF-8"?>
<TransformedData xmlns="http://www.opengeospatial.net/wcts"
xmlns:ex="http://www.opengeospatial.net/examples"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsTransform.xsd">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/11/08 -->
  <Reference xlink:href="TBD">
    <ows:Abstract>A state capital</ows:Abstract>
    <ows:Format>text/xml; gmlVersion=3.1.0</ows:Format>
  </Reference>
</TransformedData>
```

where the “TBD” must be replaced by a URL reference to the XML encoded feature.

9.4.2 Image coordinate transformation

9.4.2.1 Transform request

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

```
http://gws2.pcigeomatics.com/wcts/wcts?service=WCTS&version=0.0.20&requ
est=Transform&InputData=http://foo.com/wcs/arequesttoWCS&RESX=0.03&RESY
=0.03&OutputFormat=image/pix&SourceCRS=
```

```
urn:ogc:def:crs:OGC:1.0:XXXX&targetCRS=
urn:ogc:def:crs:OGC:1.0:YYYY&store=true
```

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

The input image is assumed to be referenced at <http://foo.com/wcs/arequesttoWCS>.

9.4.2.2 Transform response

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"?>
<wcts:TransformedData xmlns:wcts="http://www.opengeospatial.net/wcts"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <wcts:Reference
xlink:href="http://gws2.pcigeomatics.com/transformed/WCTS1097085572213/
transformedData.pix" />
</wcts:TransformedData>
```

In this example, the output image is assumed to be referenced at <http://gws2.pcigeomatics.com/transformed/WCTS1097085572213/transformedData.pix>.

10 IsTransformable operation (optional)

10.1 Introduction

The IsTransformable operation allows WCTS clients to check if coordinate transformation of a specified set of geometry primitives 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 specific coordinate expression that may be submitted by a Transform request.

NOTE In principle, this information could be extracted from the Capabilities document of the WCTS, since the parameters GeometryType, CoverageType, InterpolationMethod, 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 listed and defined in Table 17. This table also specifies the UML model data type, source of

values, and multiplicity of each listed parameter, plus the meaning to servers when each optional parameter is not included in the operation request.

NOTE 1 The first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 05-008].

Table 17 — Parameters in IsTransformable operation request

Name ^a	Definition	Data type and value	Multiplicity and use
service	Service type identifier	Character String, not empty Value is OWS type abbreviation, namely “WCTS”	One (mandatory)
request	Operation name	Character String, not empty Value is operation name, namely “IsTransformable”	One (mandatory)
version	Specification version for operation	Character String, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
IsTransformable Choice	Identification or definition of desired coordinate operation	IsTransformableChoice, see Table 18	One (mandatory)
CoverageType	Identifier of coverage type that will be requested to be transformed	Character string, not empty Unordered list of coverage type identifiers	Zero or more (Optional) ^b Included for transforming coverages
Interpolation Method	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) ^c Included when transforming coverages
GeometryType	Identifier of type of GML 3 geometric primitive that will be requested to be transformed	Character string, not empty Unordered list of geometric primitive types	Zero or more (Optional) ^b Included for transforming specific geometric primitive types
<p>^a Although some values listed in the “Name” column appear to contain spaces, they shall not contain spaces.</p> <p>^b Either at least one CoverageType or at least one GeometryType shall be included.</p> <p>^c One or more Interpolation Methods shall be included only if one or more CoverageTypes are included.</p>			

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 18 — Alternatives in IsTransformableChoice subsection

Name	Definition	Data type	Multiplicity
SourceAndTargetCRSs	Identifiers of SourceCRS and desired TargetCRS	See Table 1	Zero or one (Mutually exclusive) ^a
Transformation	Identification of desired coordinate operation	URI	Zero or one (Mutually exclusive) ^b
Method	Identification of desired operation method	URI	Zero or one (Mutually exclusive) ^c
<p>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.</p> <p>b Included when client is specifying a specific well-known coordinate operation, whose definition is known to the WCTS server</p> <p>c Included when client is considering specifying a specific user-defined coordinate operation that uses a well-known operation method.</p>			

10.2.2 IsTransformable request KVP encoding (optional)

WCTS servers 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 19. The parameters listed in Table 19 shall be as specified in Table 17 and Table 18 above.

Table 19 — 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:coordinateOperation: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
InterpolationMethods=bilinear	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 05-008].		
^b Either Transformation, Method, or both SourceCRS and TargetCRS shall be included.		

10.2.3 IsTransformable request XML encoding (required)

All WCTS servers shall implement HTTP POST transfer of the IsTransformable operation request, using XML encoding only. The following schema fragment specifies the contents and structure of an IsTransformable operation request encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:wcts="http://www.opengeospatial.net/wcts"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://opengeospatial.net/xlink"
xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengeospatial.net/wcts"
elementFormDefault="qualified" xml:lang="en">
  <annotation>
    <appinfo>fragmentIsTransformableRequest.xsd 2004/12/20</appinfo>
```

```

    <documentation>This XML Schema encodes the WCTS IsTransformable
operation request message. </documentation>
  </annotation>
  <!-- =====
    includes and imports
    ===== -->
  <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="owsAdditions.xsd"/>
  <import namespace="http://www.opengis.net/gml"
schemaLocation="..gml/3.1.1/base/coordinateReferenceSystems.xsd"/>
  <!-- =====
    elements and types
    ===== -->
  <element name="IsTransformable">
    <annotation>
      <documentation>Request to a WCTS to perform the
IsTransformable operation. This operation allows clients to check if
transformation of a specific set of geometry and/or coverage types is
possible between two coordinate reference systems. Either the desired
source and target CRSs can be directly identified, or a specific
coordinate transformation between two CRSs can be identified. This
operation will check if the identified geometries are supported and if
there is a valid way (sequence of transformation steps) to transform
the coordinates from the source CRS to the target CRS. (This operation
will not check if this transformation makes any sense.) In this XML
encoding, no "request" parameter is included, since the element name
specifies the specific operation. </documentation>
    </annotation>
    <complexType>
      <sequence>
        <choice>
          <sequence>
            <group ref="wcts:SourceAndTargetCRSs">
              <annotation>
                <documentation>Desired well-known SourceCRS and
TargetCRS, included when client is not specifying a specific coordinate
operation. </documentation>
              </annotation>
            </group>
          </sequence>
          <element name="Transformation" type="anyURI">
            <annotation>
              <documentation>Desired well-known coordinate
operation, included when client is specifying a specific coordinate
operation. </documentation>
            </annotation>
          </element>
          <element name="Method" type="anyURI">
            <annotation>
              <documentation>Desired well-known operation method
that can be used in a user-defined coordinate operation, included when
client is considering specifying a specific user-defined coordinate
operation. </documentation>
            </annotation>
          </element>
        </choice>
      </choice>
    </complexType>
  </element>

```



```

        <element name="GeometryType" type="ows:GeometryTypeType"
maxOccurs="unbounded">
            <annotation>
                <documentation>Unordered list of one or more GML 3
geometric primitive types that a client can request be transformed by a
WCTS server. It is assumed that a WCTS server can also transform the
corresponding geometric complexes and aggregates. </documentation>
            </annotation>
        </element>
    </sequence>
    <element name="CoverageType"
type="ows:CoverageTypeType" maxOccurs="unbounded">
        <annotation>
            <documentation>Unordered list of one or more
GML 3 coverage types that a client can request be transformed by a WCTS
server. </documentation>
        </annotation>
    </element>
    <element name="InterpolationMethod"
type="ows:InterpolationMethodType" minOccurs="0" maxOccurs="unbounded">
        <annotation>
            <documentation>Unordered list of zero or more
interpolation methods that a client can request be performed on a
coverage by a WCTS server. An interpolation is used after coverage
points have been transformed. </documentation>
        </annotation>
    </element>
    <element name="GeometryType"
type="ows:GeometryTypeType" minOccurs="0" maxOccurs="unbounded">
        <annotation>
            <documentation>Unordered list of zero or more
GML 3 geometric primitive types that a client can request be
transformed by a WCTS server. It is assumed that a WCTS server can also
transform the corresponding geometric complexes and aggregates.
</documentation>
        </annotation>
    </element>
</sequence>
</choice>
</sequence>
    <attribute name="service" type="wcts:ServiceType"
use="required" fixed="WCTS"/>
    <attribute name="version" type="wcts:VersionType"
use="required"/>
</complexType>
</element>
<!-- ===== -->
<group name="SourceAndTargetCRSs">
    <annotation>
        <documentation>Group combining SourceCRS and TargetCRS
elements, used by some WCTS operation requests. </documentation>
    </annotation>
    <sequence>
        <element name="SourceCRS" type="anyURI">
            <annotation>
                <documentation>The coordinate reference system (CRS)
used by coordinates input to a Transform operation. This element shall
uniquely identify the desired CRS, but the definition of that CRS need

```

not be known to the WCTS server. This element is normally a reference to that CRS, but can contain the definition of that CRS.

```

</documentation>
    </annotation>
  </element>
  <element name="TargetCRS" type="anyURI">
    <annotation>
      <documentation>The coordinate reference system (CRS)
used by coordinates output from a Transform operation. This element
shall uniquely identify the desired CRS, but the definition of that CRS
need not be known to the WCTS server. This element is normally a
reference to that CRS, but can contain the definition of that CRS.
</documentation>
    </annotation>
  </element>
</sequence>
</group>
<!-- ===== -->
<element name="Transformation"
type="gml:CoordinateOperationRefType">
  <annotation>
    <documentation>A coordinate operation that can transform
coordinates from the sourceCRS to the targetCRS identified in an
operation request. This element is often a reference to a well-known
coordinate operation, but can contain the definition of a coordinate
operation that references a well-known operation method. Alternately,
this element can contain the definition of a ConcatenatedOperation that
combines two or more well-known coordinate operations or coordinate
operations that reference a well-known operation method. The well-known
coordinate operation or operation method can be defined in the CRS
application profile referenced by the WCTS Implementation
Specification. Alternately or in addition, the well-known coordinate
operation or operation method can be defined in the Capabilities XML
document available from this WCTS server. (TBR) </documentation>
  </annotation>
</element>
<!-- ===== -->
<simpleType name="ServiceType">
  <annotation>
    <documentation>Service type identifier, where the string value
is the OGC Web Service type abbreviation. </documentation>
  </annotation>
  <restriction base="string"/>
</simpleType>
<!-- ===== -->
<simpleType name="VersionType">
  <annotation>
    <documentation>The version of the Implementation Specification
(document) to which the requested operation conforms. The value shall
be N.N.N, where each N is a non-negative integer
up to 99. </documentation>
  </annotation>
  <restriction base="string"/>
</simpleType>
<!-- ===== -->
<simpleType name="GeometryTypeType">
  <annotation>

```

```

    <documentation>Type of GML 3 geometric primitive possibly
handled by a WCTS server. The possible values are all names of GML 3
complexTypes (TBR). </documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="Envelope"/>
    <enumeration value="Point"/>
    <enumeration value="LineString"/>
    <enumeration value="Polygon"/>
    <enumeration value="LinearRing"/>
    <enumeration value="Curve"/>
    <enumeration value="LineStringSegment"/>
    <enumeration value="ArcString"/>
    <enumeration value="Arc"/>
    <enumeration value="Circle"/>
    <enumeration value="ArcStringByBulge"/>
    <enumeration value="ArcByBulge"/>
    <enumeration value="ArcByCenterPoint"/>
    <enumeration value="CircleByCenterPoint"/>
    <enumeration value="CubicSpline"/>
    <enumeration value="BSpline"/>
    <enumeration value="Bezier"/>
    <enumeration value="OrientableCurve"/>
    <enumeration value="Surface"/>
    <enumeration value="PolygonPatch"/>
    <enumeration value="Triangle"/>
    <enumeration value="Rectangle"/>
    <enumeration value="Ring"/>
    <enumeration value="OrientableSurface"/>
    <enumeration value="Solid"/>
    <enumeration value="CompositeCurve"/>
    <enumeration value="CompositeSurface"/>
    <enumeration value="CompositeSolid"/>
  </restriction>
</simpleType>
<!-- ===== -->
<simpleType name="CoverageTypeType">
  <annotation>
    <documentation>Type of GML 3 coverage possibly handled by a
WCTS server. For coverages which use specific geometric primitives, a
client should also check if the corresponding geometric primitive types
are supported. </documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="MulitPoint">
      <annotation>
        <documentation>TBD. </documentation>
      </annotation>
    </enumeration>
    <enumeration value="MultiSurface">
      <annotation>
        <documentation>TBD. </documentation>
      </annotation>
    </enumeration>
    <enumeration value="RectifiedGrid">
      <annotation>
        <documentation>TBD. </documentation>
      </annotation>
    </enumeration>
  </restriction>

```

```

        </enumeration>
    </restriction>
</simpleType>
<!-- ===== -->
<simpleType name="InterpolationMethodType">
    <annotation>
        <documentation>Codes that identify interpolation methods. The
meanings of these codes are defined in Annex B of ISO 19123: Geographic
information â€” Schema for coverage geometry and functions.
</documentation>
    </annotation>
    <restriction base="string">
        <enumeration value="nearest neighbor"/>
        <enumeration value="bilinear"/>
        <enumeration value="bicubic"/>
        <enumeration value="lost area"/>
        <enumeration value="barycentric"/>
        <enumeration value="none">
            <annotation>
                <documentation>No interpolation. </documentation>
            </annotation>
        </enumeration>
    </restriction>
</simpleType>
</schema>

```

As indicated, the above XML Schema fragment uses the coordinate reference systems portion of GML 3.1.1 [OGC 04-092r4].

10.3 Operation response

10.3.1 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. The problem value shall be one of the values specified in Table 20.

Table 20 — 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.

More completely, a response from the IsTransformable operation shall include the parameters listed in Table 21. This table also specifies the UML model data type and values, plus the multiplicity and use of each parameter.

Table 21 — Parameters in IsTransformable operation response

Name	Definition	Data type and value	Multiplicity and use
transformable	Indicates whether this WCTS server can perform a transformation from the sourceCRS to the targetCRS identified in the operation request	Boolean type Value shall be "true" or "false"	One (mandatory)
problem	Type of transformation problem detected by WCTS server	Code List type See Table 20	Zero or one (optional) Included when "transformable" is false

The following schema fragment specifies the contents and structure of an IsTransformable operation response encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengeospatial.net/wcts"
xmlns:wcts="http://www.opengeospatial.net/wcts"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
xml:lang="en">
  <annotation>
    <appinfo>fragmentIsTransformableResponse.xsd 2005-01-10</appinfo>
    <documentation>This XML Schema encodes the WCTS IsTransformable
operation message. </documentation>
  </annotation>
  <!-- =====
elements and types
===== -->
  <element name="IsTransformableResponse">
    <annotation>
      <documentation>Response to a valid IsTransformable operation
request sent to a WCTS. </documentation>
    </annotation>
    <complexType>
      <sequence/>
      <attribute name="transformable" type="boolean" use="required">
        <annotation>
          <documentation>Indicates whether this WCTS server can
perform a transformation from the sourceCRS to the targetCRS identified
in the operation request. The value shall be "true" or "false".
</documentation>
        </annotation>
      </attribute>
      <attribute name="problem" type="wcts:ProblemType"
use="optional">
        <annotation>
```

```

        <documentation>Type of transformation problem detected
by WCTS server. This attribute shall be included whenever the
"transformable" attribute is false. </documentation>
        </annotation>
    </attribute>
</complexType>
</element>
<!-- ===== -->
<simpleType name="ProblemType">
    <annotation>
        <documentation>Type of transformation problem by WCTS server.
</documentation>
    </annotation>
    <restriction base="string">
        <enumeration value="source CRS">
            <annotation>
                <documentation>WCTS server cannot transform from
identified source CRS. </documentation>
            </annotation>
        </enumeration>
        <enumeration value="target CRS">
            <annotation>
                <documentation>WCTS server cannot transform to
identified target CRS from identified source CRS. </documentation>
            </annotation>
        </enumeration>
        <enumeration value="geometry type">
            <annotation>
                <documentation>WCTS server cannot transform one or more
identified geometry types. </documentation>
            </annotation>
        </enumeration>
        <enumeration value="coverage type">
            <annotation>
                <documentation>WCTS server cannot transform one or more
identified coverage types. </documentation>
            </annotation>
        </enumeration>
        <enumeration value="interpolation method">
            <annotation>
                <documentation>WCTS server cannot perform one or more
identified interpolation methods. </documentation>
            </annotation>
        </enumeration>
        <enumeration value="other">
            <annotation>
                <documentation>WCTS server cannot perform identified
transformation due to some other problem, including incompatibility
between identified parameters. </documentation>
            </annotation>
        </enumeration>
    </restriction>
</simpleType>
</schema>

```

10.3.2 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 7.4 of [OGC 05-008]. 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?

10.4 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.opengeospatial.net/wcts"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsIsTransformable.xsd" service="WCTS" version="0.0.0">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/20-->
  <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

The “true” response and an example “false” response to the IsTransformable operation request are:

```
<?xml version="1.0" encoding="UTF-8"?>
<IsTransformableResponse xmlns="http://www.opengeospatial.net/wcts"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsIsTransformable.xsd" transformable="true"/>
<!-- Primary editor: Arliss Whiteside. Last updated 2004/11/09 -->

<?xml version="1.0" encoding="UTF-8"?>
<IsTransformableResponse xmlns="http://www.opengeospatial.net/wcts"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsIsTransformable.xsd" transformable="false"
problem="geometry type"/>
<!-- Primary editor: Arliss Whiteside. Last updated 2004/11/09 -->
```

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 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 listed and defined in Table 22. This table also specifies the UML model data type, source of values, and multiplicity of each listed parameter.

NOTE The first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 05-008]. The other parameter listed below is defined in Table 1 in Subclause 7.2 of this document.

Table 22 — Parameters in GetTransformation operation request

Name	Definition	Data type and value	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, namely “WCTS”	One (mandatory)
request	Operation name	Character String type, not empty Value is operation name, namely “GetTransformation”	One (mandatory)
version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
SourceAnd TargetCRSs	Identifiers of SourceCRS and desired TargetCRS	See Table 1	One (mandatory)

11.2.2 GetTransformation request KVP encoding (required)

All WCTS servers that implement the GetTransformation operation shall 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 23. The parameters listed in Table 23 shall be as specified in Table 22 above.

Table 23 — 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 05-008].		

11.2.3 GetTransformation request XML encoding (optional)

WCTS servers may also implement HTTP POST transfer of the GetTransformation operation request, using XML encoding only. The following schema fragment specifies the contents and structure of a GetTransformation operation request encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengeospatial.net/wcts"
xmlns:wcts="http://www.opengeospatial.net/wcts"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
xml:lang="en">
  <annotation>
    <appinfo>fragmentGetTransformationRequest.xsd
    2004/12/20</appinfo>
    <documentation>This XML Schema encodes the WCTS GetTransformation
    operation request message. </documentation>
  </annotation>
  <!-- =====
  includes and imports
  ===== -->
  <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="../../../ows/0.4.0/owsOperationsMetadata.xsd"/>
  <!-- =====
  elements and types
  ===== -->
  <element name="GetTransformation">
    <annotation>
      <documentation>Request to a WCTS to perform the
      GetTransformation operation. This operation allows a client to retrieve
      the descriptions of all server-known transformations from the
      identified source CRS to the target CRS. In this XML encoding, no
      "request" parameter is included, since the element name specifies the
      specific operation. </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="SourceCRS" type="anyURI">
          <annotation>
```

```

        <documentation>The coordinate reference system (CRS)
used by coordinates input to a Transform operation. This element shall
uniquely identify the desired CRS, but the definition of that CRS need
not be known to the WCTS server. </documentation>
        </annotation>
    </element>
    <element name="TargetCRS" type="anyURI">
        <annotation>
            <documentation>The coordinate reference system (CRS)
used by coordinates output from a Transform operation. This element
shall uniquely identify the desired CRS, but the definition of that CRS
need not be known to the WCTS server. </documentation>
        </annotation>
    </element>
</sequence>
<attribute name="service" type="wcts:ServiceType"
use="required" fixed="WCTS"/>
<attribute name="version" type="wcts:VersionType"
use="required"/>
</complexType>
</element>
<!-- ===== -->
<simpleType name="ServiceType">
    <annotation>
        <documentation>Service type identifier, where the string value
is the OGC Web Service type abbreviation. </documentation>
    </annotation>
    <restriction base="string"/>
</simpleType>
<!-- ===== -->
<simpleType name="VersionType">
    <annotation>
        <documentation>The version of the Implementation Specification
(document) to which the requested operation conforms. The value shall
be N.N.N, where each N is a non-negative integer
up to 99. </documentation>
    </annotation>
    <restriction base="string"/>
</simpleType>
</schema>

```

As indicated, the above XML Schema fragment uses the coordinate reference systems portion of GML 3.1.1 [OGC 04-092r4].

11.3 Operation response

11.3.1 GetTransformation normal response

The normal response to a GetTransformation operation request is one or more coordinate transformations, encoded in XML using GML 3.1.1. More precisely, a response from the Transform operation shall include the parts listed in Table 24. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

Table 24 — Parts of GetTransformation operation response

Name	Definition	Data type	Multiplicity and use
Transformation	Definition of coordinate operation that can transform coordinates from the sourceCRS to the targetCRS identified in operation request	Transformation parameters group, see Table 2	Zero or more (mandatory) One for each relevant coordinate operation

The following XML Schema fragment specifies the contents and structure of a GetTransformation operation response:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengeospatial.net/wcts"
xmlns:wcts="http://www.opengeospatial.net/wcts"
xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
xml:lang="en">
  <annotation>
    <appinfo>wctsGetAndDescribeTransformationResponse.xsd 2005-01-
10</appinfo>
    <documentation>
      <description>This XML Schema encodes the Get Transformation
and Describe Transformation packages of the UML model for the OGC Web
Coordinate Transformation Service (WCTS). Those UML packages define the
GetTransformation and DescribeTransformation operation requests and
response XML elements and types. These operations are combined here
because these two operations have the same response. </description>
    </documentation>
  </annotation>
  <!-- =====
includes and imports
===== -->
  <import namespace="http://www.opengis.net/gml"
schemaLocation="../../gml/3.1.1/base/coordinateOperations.xsd"/>
  <!-- =====
elements and types
===== -->
  <element name="TransformDescription">
    <annotation>
      <documentation>Response to a valid GetTransformation or
Describe Transformation operation request sent to a WCTS.
</documentation>
    </annotation>
    <complexType>
      <sequence>
        <element ref="wcts:transformation" minOccurs="0"
maxOccurs="unbounded">
          <annotation>
            <documentation>Unordered list of zero or more
coordinate operations. In the GetTransformation operation response,
these coordinate operations can transform coordinates from the
sourceCRS to the targetCRS identified in the GetTransformation
operation request. In the DescribeTransformation operation response,
these coordinate operations are the ones identified in the
```

DescribeTransformation operation request. In this use, each element shall either:

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

- * Contain a URL from which that definition object can be retrieved, using GML encoding </documentation>

```

    </annotation>
  </element>
</sequence>
</complexType>
</element>
<!-- ===== -->
<element name="transformation">
  <annotation>
    <documentation>A coordinate operation that can transform
coordinates from a sourceCRS to a targetCRS. This element is often a
reference to a well-known coordinate operation, but can contain the
definition of a user-defined coordinate operation that references a
well-known operation method. Alternately, this element can contain the
definition of a ConcatenatedOperation that combines two or more well-
known coordinate operations or coordinate operations that reference a
well-known operation method. When the definition of a coordinate
operation is contained, the element shall also include a xlink:href
value containing the URN that references this definition. Well-known
coordinate operations and operation methods can be defined in documents
referenced by the WCTS Implementation Specification. Alternately or in
addition, well-known coordinate operations and operation methods can be
defined in metadata referenced by or contained in the Contents section
of the Capabilities XML document available from this WCTS server.
</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:CoordinateOperationRefType">
        <sequence minOccurs="0">
          <annotation>
            <documentation>This sourceCRS and targetCRS shall
be included only when the referenced or contained coordinate operation
is a Conversion, which does not identify its sourceCRS and targetCRS.
</documentation>
          </annotation>
          <element name="sourceCRS" type="gml:CRSRefType">
            <annotation>
              <documentation>The coordinate reference system
(CRS) used by coordinates input to a Conversion. This element shall
uniquely identify the desired CRS, but the definition of that CRS need
not be known to the WCTS server. This element is normally a reference
to that CRS, but can contain the definition of a user-defined CRS. When
this element contains the CRS definition object, it shall also contain
a xlink:href containing the URN that references this definition.
</documentation>
            </annotation>
          </element>
          <element name="targetCRS" type="gml:CRSRefType">
            <annotation>
              <documentation>The coordinate reference system
(CRS) used by coordinates output from a Conversion. This element shall

```

uniquely identify the desired CRS, but the definition of that CRS need not be known to the WCTS server. This element is normally a reference to that CRS, but can contain the definition of a user-defined CRS. When this element contains the CRS definition object, it shall also contain a `xlink:href` containing the URN that references this definition.

```
</documentation>
      </annotation>
    </element>
  </sequence>
</extension>
</complexContent>
</complexType>
</element>
</schema>
```

As indicated, the above XML Schema fragment uses the coordinate operations portion of GML 3.1.1 [OGC 04-092r4].

It is possible that a transformation is not feasible. Possible reasons are that the web coordinate transformation service 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.

11.3.2 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 7.4 of [OGC 05-008]. 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?

11.4 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:ogc:
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.opengeospatial.net/wcts"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsGetAndDescribeTransformation.xsd" service="WCTS"
version="0.0.0">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/20-->
```

```

    <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"?>
<TransformDescription xmlns="http://www.opengeospatial.net/wcts"
  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.opengeospatial.net/wcts
    ../Schemas/wctsGetAndDescribeTransformation.xsd
    http://www.opengis.net/gml ../gml/3.1.1/base/coordinateOperations.xsd">
  <!-- Primary editor: Arliss Whiteside. Last updated 2005-01-10-->
  <!-- This XML example is for a specific coordinate operation that
    uses the well-known Transverse Mercator operation method as defined by
    the EPSG, adapted from Subclause E.5 of 03-010r9. -->
  <!-- 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) -->
  <transformation>
    <gml:Conversion gml:id="EPSG19916">
      <gml:coordinateOperationName>Transverse
Mercator</gml:coordinateOperationName>
      <gml:coordinateOperationID>
        <gml:name
codeSpace="urn:ogc:def:coordinateOperation:EPSG:6.0:">19916</gml:name>
        </gml:coordinateOperationID>
        <gml:usesMethod xlink:href="urn:ogc:def:method:EPSG:6.0:9807"
xlink:title="Transverse Mercator"/>
        <gml:usesValue>
          <gml:value
uom="urn:ogc:def:uom:OGC:1.0:degree">49</gml:value>
          <gml:valueOfParameter
xlink:href="urn:ogc:parameter:EPSG:6.0:8801" xlink:title="Latitude of
natural origin"/>
          </gml:usesValue>
          <gml:usesValue>
            <gml:value uom="urn:ogc:def:uom:OGC:1.0:degree">-
2</gml:value>
            <gml:valueOfParameter
xlink:href="urn:ogc:parameter:EPSG:6.0:8802" xlink:title="Longitude of
natural origin"/>
            </gml:usesValue>
            <gml:usesValue>
              <gml:value
uom="urn:ogc:def:uom:OGC:1.0:unity">0.999601272</gml:value>
              <gml:valueOfParameter
xlink:href="urn:ogc:parameter:EPSG:6.0:8805" xlink:title="Scale factor
at natural origin"/>
              </gml:usesValue>
              <gml:usesValue>

```

```

        <gml:value
uom="urn:ogc:def:uom:OGC:1.0:metre">400000</gml:value>
        <gml:valueOfParameter
xlink:href="urn:ogc:parameter:EPSG:6.0:8806" xlink:title="False
Easting"/>
        </gml:usesValue>
        <gml:usesValue>
        <gml:value uom="urn:ogc:def:uom:OGC:1.0:metre">-
100000</gml:value>
        <gml:valueOfParameter
xlink:href="urn:ogc:parameter:EPSG:6.0:8807" xlink:title="False
Northing"/>
        </gml:usesValue>
        </gml:Conversion>
    </transformation>
</TransformDescription>

```

12 DescribeTransformation Operation (optional)

12.1 Introduction

The DescribeTransformation operation allows WCTS clients to retrieve the definitions of one or more identified coordinate operations. The definitions in the operation response are encoded in XML using GML 3.1.1. Implementation of the DescribeTransformation operation is optional for a web coordinate transformation server.

12.2 Operation request

12.2.1 DescribeTransformation request parameters

A request to perform the DescribeTransformation operation shall include the parameters listed and defined in Table 25. This table also specifies the UML model data type, source of values, and multiplicity of each listed parameter.

NOTE The first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 05-008].

Table 25 — Parameters in DescribeTransformation operation request

Name	Definition	Data type and value	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, namely “WCTS”	One (mandatory)
request	Operation name	Character String type, not empty Value is operation name, namely “DescribeTransformation”	One (mandatory)
version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
Transformation	Identifier of desired coordinate operation	URI Values defined in service metadata or in data known to client	One or more (mandatory) One for each coordinate operation desired

12.2.2 DescribeTransformation request KVP encoding (required)

All WCTS servers that implement the DescribeTransformation operation shall implement HTTP GET transfer of the DescribeTransformation operation request, using KVP encoding. The KVP encoding of the DescribeTransformation operation request shall use the parameters specified in Table 26. The parameters listed in Table 26 shall be as specified in Table 25 above.

Table 26 — DescribeTransformation operation request URL parameters

Name and example ^a	Optionality	Definition and format
service=WCTS	Mandatory	Service type identifier
request=DescribeTransformation	Mandatory	Operation name
version=0.0.0	Mandatory	Specification and schema version for this operation
Transformations=urn:ogc:def:coordinateOperation:EPSG:6.3:19916	Mandatory	Identifier URIs of one or more coordinate operations, 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 05-008].		

12.2.3 DescribeTransformation request XML encoding (optional)

WCTS servers may also implement HTTP POST transfer of the DescribeTransformation operation request, using XML encoding only. The following schema fragment specifies the contents and structure of a DescribeTransformation operation request encoded in XML:


```

<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengeospatial.net/wcts"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml"
xmlns:wcts="http://www.opengeospatial.net/wcts"
elementFormDefault="qualified" xml:lang="en">
  <annotation>
    <appinfo>fragmentDescribeTransformationRequest.xsd 2005-01-
10</appinfo>
    <documentation>This XML Schema encodes the WCTS
DescribeTransformation operation request message. </documentation>
  </annotation>
  <!-- =====
includes and imports
===== -->
  <import namespace="http://www.opengis.net/gml"
schemaLocation=" ../gml/3.1.1/base/coordinateReferenceSystems.xsd"/>
  <!-- =====
elements and types
===== -->
  <element name="DescribeTransformation">
    <annotation>
      <documentation>Request to a WCTS to perform the
DescribeTransformation operation. This operation allows a client to
retrieve the definitions of one or more identified Transformations. In
this XML encoding, no "request" parameter is included, since the
element name specifies the specific operation. </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="Transformation" type="anyURI"
maxOccurs="unbounded">
          <annotation>
            <documentation>Unordered list of one or more
Transformation identifiers, usually identifiers listed in the Contents
section of the WCTS Capabilities document.</documentation>
          </annotation>
        </element>
      </sequence>
      <attribute name="service" type="wcts:ServiceType"
use="required" fixed="WCTS"/>
      <attribute name="version" type="wcts:VersionType"
use="required"/>
    </complexType>
  </element>
  <!-- ===== -->
  <simpleType name="ServiceType">
    <annotation>
      <documentation>Service type identifier, where the string value
is the OGC Web Service type abbreviation. </documentation>
    </annotation>
    <restriction base="string"/>
  </simpleType>
  <!-- ===== -->
  <simpleType name="VersionType">
    <annotation>

```

```

        <documentation>The version of the Implementation Specification
(document) to which the requested operation conforms. The value shall
be N.N.N, where each N is a non-negative integer
up to 99. </documentation>
    </annotation>
    <restriction base="string"/>
</simpleType>
</schema>

```

12.3 Operation response

12.3.1 DescribeTransformation normal response

The normal response to a DescribeTransformation operation request is one or more coordinate operation definitions, encoded in XML using GML 3.1.1. More precisely, a response from the DescribeTransformation operation shall include the parts listed in Table 27. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

Table 27 — Parts of DescribeTransformation operation response

Name	Definition	Data type	Multiplicity and use
Transformation	Definition of coordinate operation identified in operation request	Transformation parameter group, see Table 2	One or more (mandatory) One for each identified coordinate operation

The following XML Schema fragment specifies the contents and structure of a DescribeTransformation operation response:

```

<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengeospatial.net/wcts"
xmlns:wcts="http://www.opengeospatial.net/wcts"
xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
xml:lang="en">
  <annotation>
    <appinfo>wctsGetAndDescribeTransformationResponse.xsd 2005-01-
10</appinfo>
    <documentation>
      <description>This XML Schema encodes the Get Transformation
and Describe Transformation packages of the UML model for the OGC Web
Coordinate Transformation Service (WCTS). Those UML packages define the
GetTransformation and DescribeTransformation operation requests and
response XML elements and types. These operations are combined here
because these two operations have the same response. </description>
    </documentation>
  </annotation>
  <!-- =====
includes and imports
===== -->
  <import namespace="http://www.opengis.net/gml"
schemaLocation="../../gml/3.1.1/base/coordinateOperations.xsd"/>
  <!-- =====

```

```

elements and types
===== -->
<element name="TransformDescription">
  <annotation>
    <documentation>Response to a valid GetTransformation or
Describe Transformation operation request sent to a WCTS.
</documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="wcts:transformation" minOccurs="0"
maxOccurs="unbounded">
        <annotation>
          <documentation>Unordered list of zero or more
coordinate operations. In the GetTransformation operation response,
these coordinate operations can transform coordinates from the
sourceCRS to the targetCRS identified in the GetTransformation
operation request. In the DescribeTransformation operation response,
these coordinate operations are the ones identified in the
DescribeTransformation operation request. In this use, each element
shall either:
* Contain the coordinate operation definition object, using GML
encoding, and a xlink:href value containing the URN that references
this definition
* Contain a URL from which that definition object can be retrieved,
using GML encoding </documentation>
        </annotation>
      </element>
    </sequence>
  </complexType>
</element>
<!-- ===== -->
<element name="transformation">
  <annotation>
    <documentation>A coordinate operation that can transform
coordinates from a sourceCRS to a targetCRS. This element is often a
reference to a well-known coordinate operation, but can contain the
definition of a user-defined coordinate operation that references a
well-known operation method. Alternately, this element can contain the
definition of a ConcatenatedOperation that combines two or more well-
known coordinate operations or coordinate operations that reference a
well-known operation method. When the definition of a coordinate
operation is contained, the element shall also include a xlink:href
value containing the URN that references this definition. Well-known
coordinate operations and operation methods can be defined in documents
referenced by the WCTS Implementation Specification. Alternately or in
addition, well-known coordinate operations and operation methods can be
defined in metadata referenced by or contained in the Contents section
of the Capabilities XML document available from this WCTS server.
</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="gml:CoordinateOperationRefType">
        <sequence minOccurs="0">
          <annotation>
            <documentation>This sourceCRS and targetCRS shall
be included only when the referenced or contained coordinate operation

```

is a Conversion, which does not identify its sourceCRS and targetCRS.
 </documentation>

```

    </annotation>
    <element name="sourceCRS" type="gml:CRSRefType">
      <annotation>
        <documentation>The coordinate reference system
        (CRS) used by coordinates input to a Conversion. This element shall
        uniquely identify the desired CRS, but the definition of that CRS need
        not be known to the WCTS server. This element is normally a reference
        to that CRS, but can contain the definition of a user-defined CRS. When
        this element contains the CRS definition object, it shall also contain
        a xlink:href containing the URN that references this definition.
        </documentation>
      </annotation>
    </element>
    <element name="targetCRS" type="gml:CRSRefType">
      <annotation>
        <documentation>The coordinate reference system
        (CRS) used by coordinates output from a Conversion. This element shall
        uniquely identify the desired CRS, but the definition of that CRS need
        not be known to the WCTS server. This element is normally a reference
        to that CRS, but can contain the definition of a user-defined CRS. When
        this element contains the CRS definition object, it shall also contain
        a xlink:href containing the URN that references this definition.
        </documentation>
      </annotation>
    </element>
  </sequence>
</extension>
</complexContent>
</complexType>
</element>
</schema>

```

If a transformation definition is not available for one or more of the identified transformations, the WCTS shall return an exception report message.

12.3.2 Exceptions

When a web coordinate transformation server encounters an error while performing a DescribeTransformation operation, it shall return an exception report message as specified in Subclause 7.4 of the OGC Web Services Common Specification. The allowed exception codes shall include those listed in Table 3 in Subclause 7.3 of this document.

12.4 Examples

12.4.1 DescribeTransformation request

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

```

www.lat-lon.de/transform&service=WCTS&request=DescribeTransformation
&version=0.0.20&transformation=urn:ogc:def:coordinateOperation:EPSG:6.3
:19916&

```

The corresponding DescribeTransformation operation request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<DescribeTransformation xmlns="http://www.opengeospatial.net/wcts"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsGetAndDescribeTransformation.xsd" service="WCTS"
version="0.0.0">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/20-->
  <Transformation>urn:ogc:def:coordinateOperation:EPSG:6.3:19916</Tran
sformation>
</DescribeTransformation>
```

12.4.2 DescribeTransformation response

If no exception occurs, the server will reply to a DescribeTransformation request with a list of one or more transformations embedded in TransformDescription root element. If transformation EPSG 19916 is requested, the response might be:

```
<?xml version="1.0" encoding="UTF-8"?>
<TransformDescription xmlns="http://www.opengeospatial.net/wcts"
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.opengeospatial.net/wcts
../Schemas/wctsGetAndDescribeTransformation.xsd
http://www.opengis.net/gml ../gml/3.1.1/base/coordinateOperations.xsd">
  <!-- Primary editor: Arliss Whiteside. Last updated 2005-01-10-->
  <!-- This XML example is for a specific coordinate operation that
uses the well-known Transverse Mercator operation method as defined by
the EPSG, adapted from Subclause E.5 of 03-010r9. -->
  <transformation>
    <gml:Conversion gml:id="EPSG19916">
      <gml:coordinateOperationName>Transverse
Mercator</gml:coordinateOperationName>
      <gml:coordinateOperationID>
        <gml:name
codeSpace="urn:ogc:def:coordinateOperation:EPSG:6.0:">19916</gml:name>
      </gml:coordinateOperationID>
      <gml:usesMethod xlink:href="urn:ogc:def:method:EPSG:6.0:9807"
xlink:title="Transverse Mercator"/>
      <gml:usesValue>
        <gml:value
uom="urn:ogc:def:uom:OGC:1.0:degree">49</gml:value>
        <gml:valueOfParameter
xlink:href="urn:ogc:def:parameter:EPSG:6.0:8801" xlink:title="Latitude
of natural origin"/>
      </gml:usesValue>
      <gml:usesValue>
        <gml:value uom="urn:ogc:def:uom:OGC:1.0:degree">-
2</gml:value>
        <gml:valueOfParameter
xlink:href="urn:ogc:def:parameter:EPSG:6.0:8802" xlink:title="Longitude
of natural origin"/>
      </gml:usesValue>
```

```

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

```

13 DescribeCRS operation (optional)

13.1 Introduction

The DescribeCRS operation allows WCTS clients to retrieve the definitions of one or more identified coordinate reference systems (CRSs). The definitions in the operation response are encoded in XML using GML 3.1.1. Implementation of the DescribeCRS operation is optional for a web coordinate transformation server.

13.2 Operation request

13.2.1 DescribeCRS request parameters

A request to perform the DescribeCRS operation shall include the parameters listed and defined in Table 28. This table also specifies the UML model data type, source of values, and multiplicity of each listed parameter, plus the meaning to servers when each optional parameter is not included in the operation request.

NOTE The first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 05-008].

Table 28 — Parameters in DescribeCRS operation request

Name	Definition	Data type and value	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, namely “WCTS”	One (mandatory)
request	Operation name	Character String type, not empty Value is operation name, namely “DescribeCRS”	One (mandatory)
version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
CRS	Identifier of desired coordinate reference system	URI Values defined in service metadata or in data known to client	One or more (mandatory) One for each CRS desired

13.2.2 DescribeCRS request KVP encoding (required)

All WCTS servers that implement the DescribeCRS operation shall implement HTTP GET transfer of the DescribeCRS operation request, using KVP encoding. The KVP encoding of the DescribeCRS operation request shall use the parameters specified in Table 29. The parameters listed in Table 29 shall be as specified in Table 28 above.

Table 29 — DescribeCRS operation request URL parameters

Name and example	Optionality	Definition and format
service=WCTS	Mandatory	Service type identifier
request=DescribeCRS	Mandatory	Operation name
version=0.0.0	Mandatory	Specification and schema version for this operation
CRSs= urn:ogc:def:crs:EPSG:6.3:4277	Mandatory	Identifier URIs of one or more desired coordinate reference systems, comma separated list

13.2.3 DescribeCRS request XML encoding (optional)

WCTS servers may also implement HTTP POST transfer of the DescribeCRS operation request, using XML encoding only. The following schema fragment specifies the contents and structure of a DescribeCRS operation request encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengeospatial.net/wcts"
xmlns:wcts="http://www.opengeospatial.net/wcts"
xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
xml:lang="en">
  <annotation>
    <appinfo>fragmentDescribeCRSRequest.xsd 2004/12/03</appinfo>
```

```

    <documentation>This XML Schema encodes the WCTS DescribeCRS
operation request message. </documentation>
  </annotation>
  <!-- =====
    includes and imports
    ===== -->
  <import namespace="http://www.opengis.net/gml"
schemaLocation=" ../gml/3.1.1/base/referenceSystems.xsd"/>
  <!-- =====
    elements and types
    ===== -->
  <element name="DescribeCRS">
    <annotation>
      <documentation>Request to a WCTS to perform the DescribeCRS
operation. This operation allows a client to retrieve the definitions
of one or more identified coordinate reference systems (CRSs). In this
XML encoding, no "request" parameter is included, since the element
name specifies the specific operation. </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="CRS" type="anyURI" maxOccurs="unbounded">
          <annotation>
            <documentation>Unordered list of one or more CRS
identifiers. These CRS identifiers are usually listed in the Contents
section of the WCTS Capabilities document.</documentation>
          </annotation>
        </element>
      </sequence>
      <attribute name="service" type="wcts:ServiceType"
use="required" fixed="WCTS"/>
      <attribute name="version" type="wcts:VersionType"
use="required"/>
    </complexType>
  </element>
  <!-- ===== -->
  <simpleType name="ServiceType">
    <annotation>
      <documentation>Service type identifier, where the string value
is the OGC Web Service type abbreviation. </documentation>
    </annotation>
    <restriction base="string"/>
  </simpleType>
  <!-- ===== -->
  <simpleType name="VersionType">
    <annotation>
      <documentation>The version of the Implementation Specification
(document) to which the requested operation conforms. The value shall
be N.N.N, where each N is a non-negative integer
up to 99. </documentation>
    </annotation>
    <restriction base="string"/>
  </simpleType>
</schema>

```


13.3 Operation response

13.3.1 DescribeCRS normal response

The normal response to a DescribeCRS operation request is one or more CRS definitions, encoded in XML using GML 3.1.1. More precisely, a response from the DescribeCRS operation shall include the parts listed in Table 30. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

Table 30 — Parts of DescribeCRS operation response

Name	Definition	Data type	Multiplicity and use
crs	Definition of CRS identified in operation request	SC_CRS ^a	One or more (mandatory) One for each identified CRS
^a In this use, each SC_CRS shall contain the CRS definition object, with the identifier used in the operation request.			

The following schema fragment specifies the contents and structure of a DescribeCRS operation response, always encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:ows="http://www.opengeospatial.net/ows"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:wcts="http://www.opengeospatial.net/wcts"
  targetNamespace="http://www.opengeospatial.net/wcts"
  elementFormDefault="qualified" xml:lang="en">
  <annotation>
    <appinfo>fragmentDescribeCRSResponse.xsd 2004/12/20</appinfo>
    <documentation>This XML Schema encodes the WCTS DescribeCRS
operation response message. </documentation>
  </annotation>
  <!-- =====
    includes and imports
  ===== -->
  <import namespace="http://www.opengis.net/gml"
schemaLocation="../../../gml/3.1.1/base/referenceSystems.xsd"/>
  <import namespace="http://www.opengeospatial.net/ows"
schemaLocation="../../../ows/0.4.0/owsOperationsMetadata.xsd"/>
  <!-- =====
    elements and types
  ===== -->
  <element name="CRSDescription">
    <annotation>
      <documentation>Response to a DescribeCRS operation request.
For each identified CRS, contains the definition of that CRS encoded in
GML 3.1. </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="crs" type="gml:CRSRefType"
maxOccurs="unbounded">
          <annotation>
```

```

        <documentation>Ordered list of CRS definition
objects, in the same order as in the operation request. In this use,
each element shall contain the CRS definition object, and the
xlink:href value shall be the identifier used in the operation request.
</documentation>
        </annotation>
    </element>
</sequence>
</complexType>
</element>
</schema>

```

If a CRS definition is not available for one or more of the identified CRSs, the WCTS shall return an exception report message.

13.3.2 DescribeCRS exceptions

When a WCTS server encounters an error while performing a DescribeCRS operation, it shall return an exception report message as specified in Subclause 7.4 of [OGC 05-008]. The allowed standard exception codes shall include those listed in Table 3 in Subclause 7.3 of this document.

13.4 Examples

13.4.1 DescribeCRS request

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

```
http://www.lat-lon.de/transform&service=WCTS&request=DescribeCRS
&version=0.0.20&CRSs=urn:ogc:def:crs:EPSG:6.3:4277&
```

The corresponding DescribeCRS operation request encoded in XML is:

```

<?xml version="1.0" encoding="UTF-8"?>
<DescribeCRS xmlns="http://www.opengeospatial.net/wcts"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsDescribeCRS.xsd" service="WCTS" version="0.0.0">
    <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/20-->
    <CRS>urn:ogc:def:crs:EPSG:6.3:4277</CRS>
</DescribeCRS>

```

13.4.2 DescribeCRS response

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

```

<?xml version="1.0" encoding="UTF-8"?>
<CRSDescription xmlns="http://www.opengeospatial.net/wcts"
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.opengeospatial.net/wcts
../Schemas/wctsDescribeCRS.xsd http://www.opengis.net/gml
../gml/3.1.1/base/coordinateReferenceSystems.xsd">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/20-->
  <crs>
    <gml:GeographicCRS gml:id="EPSG4277">
      <gml:srsName>OSGB 1936</gml:srsName>
      <gml:srsID>
        <gml:name
codeSpace="urn:ogc:def:crs:EPSG:6.0:">4277</gml:name>
        </gml:srsID>
        <gml:validArea>
          <gml:description>United Kingdom (UK) - Great Britain -
England Scotland Wales - onshore; Isle of Man. </gml:description>
        </gml:validArea>
        <gml:usesEllipsoidalCS>
          <gml:EllipsoidalCS gml:id="EPSG6402">
            <gml:csName>ellipsoidal</gml:csName>
            <gml:csID>
              <gml:name
codeSpace="urn:ogc:def:ellipsoid:EPSG:6.0:">6402</gml:name>
            </gml:csID>
            <gml:usesAxis>
              <gml:CoordinateSystemAxis gml:id="EPSG9901"
gml:uom="urn:ogc:uom:OGC:1.0:degree">
                <gml:name>Geodetic latitude</gml:name>
                <gml:axisID>
                  <gml:name
codeSpace="urn:ogc:def:axis:EPSG:6.0:">9901</gml:name>
                  </gml:axisID>
                  <gml:axisAbbrev>Lat</gml:axisAbbrev>
                  <gml:axisDirection
codeSpace="urn:ogc:def:axisDirection:OGC:1.0:">north</gml:axisDirection
                  >
                    </gml:CoordinateSystemAxis>
                  </gml:usesAxis>
                <gml:usesAxis>
                  <gml:CoordinateSystemAxis gml:id="EPSG9902"
gml:uom="urn:ogc:uom:OGC:1.0:degree">
                    <gml:name>Geodetic longitude</gml:name>
                    <gml:axisID>
                      <gml:name
codeSpace="urn:ogc:def:axis:EPSG:6.0:">9902</gml:name>
                      </gml:axisID>
                      <gml:axisAbbrev>Lon</gml:axisAbbrev>
                      <gml:axisDirection
codeSpace="urn:ogc:def:axisDirection:OGC:1.0:">east</gml:axisDirection>
                      </gml:CoordinateSystemAxis>
                    </gml:usesAxis>
                  </gml:EllipsoidalCS>
                </gml:usesEllipsoidalCS>
                <gml:usesGeodeticDatum>
                  <gml:GeodeticDatum gml:id="EPSG6277">
                    <gml:datumName>OSGB 1936</gml:datumName>
                    <gml:datumID>
                      <gml:name
codeSpace="urn:ogc:def:datum:EPSG:6.0:">6277</gml:name>

```

```

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

```

14 DescribeMethod operation (optional)

14.1 Introduction

The DescribeMethod operation allows WCTS clients to retrieve the definitions of one or more identified operation methods, primarily for methods than can be used in user-defined coordinate transformations. The definitions in the operation response are encoded in XML using a GML application schema defined in Subclause 14.3.2. Implementation of the DescribeMethod operation is optional for a web coordinate transformation server.

14.2 Operation request

14.2.1 DescribeMethod request parameters

A request to perform the DescribeMethod 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, plus the meaning to servers when each optional parameter is not included in the operation request.

NOTE The first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 05-008].

Table 31 — Parameters in DescribeMethod operation request

Name ^a	Definition	Data type and value	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, namely “WCTS”	One (mandatory)
request	Operation name	Character String type, not empty Value is operation name, namely “DescribeMethod”	One (mandatory)
version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
Method	Identifier of desired operation method	URI Values defined in service metadata or in data known to client	One or more (mandatory) One for each identifier input
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].			

14.2.2 DescribeMethod request KVP encoding (required)

All WCTS servers that implement the DescribeMethod operation shall implement HTTP GET transfer of the DescribeMethod operation request, using KVP encoding. The KVP encoding of the DescribeMethod operation request shall use the parameters specified in Table 32. The parameters listed in Table 32 shall be as specified in Table 31 above.

Table 32 — DescribeMethod operation request URL parameters

Name and example ^a	Optionality	Definition and format
service=WCTS	Mandatory	Service type identifier
request= DescribeMethod	Mandatory	Operation name
version=0.0.20	Mandatory	Specification and schema version for this operation
methods=urn:ogc:def:method:EPSG:6.3:9807	Mandatory	Identifier URIs of one or more desired operation methods, comma separated
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 05-008].		

14.2.3 DescribeMethod request XML encoding (optional)

WCTS servers may also implement HTTP POST transfer of the DescribeMethod operation request, using XML encoding only. The following schema fragment specifies the contents and structure of a DescribeMethod operation request encoded in XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengeospatial.net/wcts"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:wcts="http://www.opengeospatial.net/wcts"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:wcs="http://www.opengeospatial.net/wcs"
elementFormDefault="qualified" xml:lang="en">
  <annotation>
    <appinfo>describeMethodRequest.xsd 2004/12/06</appinfo>
    <documentation>
      <description>This XML Schema encodes the WCTS DescribeMethod
operation request XML elements and types. </description>
    </documentation>
  </annotation>
  <!-- =====
elements and types
===== -->
  <element name="DescribeMethod">
    <annotation>
      <documentation>Request to a WCTS to perform the DescribeMethod
operation. This operation allows a client to retrieve the definitions
of one or more identified transformation methods. In this XML encoding,
no "request" parameter is included, since the element name specifies
the specific operation. </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element name="Method" type="anyURI" maxOccurs="unbounded">
          <annotation>
            <documentation>Unordered list of one or more
operation method identifiers. These operation method identifiers are
usually listed in the Contents section of the WCTS Capabilities
document.</documentation>
          </annotation>
        </element>
      </sequence>
      <attribute name="service" type="wcts:ServiceType"
use="required" fixed="WCTS"/>
      <attribute name="version" type="wcts:VersionType"
use="required"/>
    </complexType>
  </element>
  <!-- ===== -->
  <simpleType name="ServiceType">
    <annotation>
      <documentation>Service type identifier, where the string value
is the OGC Web Service type abbreviation. </documentation>
    </annotation>
    <restriction base="string"/>
  </simpleType>
  <!-- ===== -->
  <simpleType name="VersionType">
    <annotation>
      <documentation>The version of the Implementation Specification
(document) to which the requested operation conforms. The value shall
be N.N.N, where each N is a non-negative integer
up to 99. </documentation>

```

```

    </annotation>
    <restriction base="string"/>
  </simpleType>
</schema>

```

14.3 Operation response

14.3.1 DescribeMethod normal response

The normal response to a DescribeMethod operation request is one or more operation method definitions, encoded in XML using GML 3.1.1. More precisely, a response from the DescribeMethod operation shall include the parts listed in Table 33. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

Table 33 — Parts of DescribeMethod operation response

Name	Definition	Data type	Multiplicity and use
method	Definition of operation method identified in operation request	CC_OperationMethod ^a	One or more (mandatory) One for each identified method
a In this use, each CC_OperationMethod shall contain the definition object with the identifier used in the operation request.			

The following schema fragment specifies the contents and structure of a DescribeMethod operation response, always encoded in XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:ows="http://www.opengeospatial.net/ows"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:wcts="http://www.opengeospatial.net/wcts"
  xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.opengeospatial.net/wcts"
  elementFormDefault="qualified" xml:lang="en">
  <annotation>
    <appinfo>describeMethodResponse.xsd 2004/12/20</appinfo>
    <documentation>
      <description>This XML Schema encodes the WCTS DescribeMethod
operation response XML elements. </description>
    </documentation>
  </annotation>
  <!-- =====
    includes and imports
  ===== -->
  <include schemaLocation="wctsOperationParameter.xsd"/>
  <import namespace="http://www.opengis.net/gml"
schemaLocation="../gml/3.1.1/base/coordinateOperations.xsd"/>
  <!-- =====
    elements and types
  ===== -->
  <element name="MethodDescription">
    <annotation>
      <documentation>Response to a DescribeMethod operation request.
For each identified method, contains the definition of that operation
method encoded in GML 3.1. </documentation>

```

```

        </annotation>
        <complexType>
          <sequence>
            <element name="method" type="gml:OperationMethodRefType"
maxOccurs="unbounded">
              <annotation>
                <documentation>Ordered list of Operation Method
definitions, in the same order as in the operation request. In this
use, each element shall contain the method definition object, and the
xlink:href value shall be the identifier used in the operation request.
</documentation>
              </annotation>
            </element>
          </sequence>
        </complexType>
      </element>
</schema>

```

If an operation method definition is not available for one or more of the identified methods, the WCTS shall return an exception report message.

14.3.2 GML Application Schema

The above XML Schema fragment uses a GML Application Schema specifically 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 XML Schema file named `wctsOperationParameter.xsd`. This schema adds three XML elements to the definition of an `OperationParameter`, namely the:

- a) Optional `allowedValues` element, allowing specification of the allowed values or range(s) of this parameter. This element can and should include the data type of this parameter, plus the reference system or units of measure.
- b) Optional `defaultValue` element, allowing specification of any default value of this parameter. This element can also include the data type of this parameter, plus the reference system or units of measure.
- c) Optional list of `supportedUOM` elements, allowing referencing the supported units of measure for values of this parameter.

This GML Application Schema uses a separate XML Schema for defining allowed values, named `owsDomainType.xsd`, also attached to this document, and adapted from `values.xsd` in WCS 1.0. Since this `owsDomainType.xsd` is expected to be used by more than one OWS, it has been defined in the “ows” namespace.

14.3.3 DescribeMethod exceptions

When a WCTS server encounters an error while performing a DescribeMethod operation, it shall return an exception report message as specified in Subclause 7.4 of [OGC 05-008]. The allowed standard exception codes shall include those listed in Table 3 in Subclause 7.3 of this document.

14.4 Examples

14.4.1 Transverse Mercator example

14.4.1.1 DescribeMethod request

A DescribeMethod operation request for the EPSG 9807 operation method can look like this encoded in KVP:

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

The corresponding DescribeMethod operation request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<DescribeMethod xmlns="http://www.opengeospatial.net/wcts"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsDescribeMethod.xsd" service="WCTS" version="0.0.0">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/20-->
  <Method>urn:ogc:def:method:EPSG:6.3:9807</Method>
</DescribeMethod>
```

14.4.1.2 DescribeMethod response

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

```
<?xml version="1.0" encoding="UTF-8"?>
<MethodDescription xmlns="http://www.opengeospatial.net/wcts"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsDescribeMethod.xsd http://www.opengis.net/gml
../gml/3.1.1/base/coordinateOperations.xsd">
  <!-- Primary editor: Arliss Whiteside. Last updated 2004/12/03 -->
  <method>
    <gml:OperationMethod gml:id="EPSG9807">
      <gml:methodName>Transverse Mercator</gml:methodName>
      <gml:methodID>
        <gml:name
codeSpace="urn:ogc:def:method:EPSG:6.0:">9807</gml:name>
      </gml:methodID>
      <gml:remarks>TBD</gml:remarks>
```

```

        <gml:methodFormula>See Section 1.4.6 "Transverse Mercator" of
        EPSG Guidance Note 7, December 2000. </gml:methodFormula>
        <gml:sourceDimensions>2</gml:sourceDimensions>
        <gml:targetDimensions>2</gml:targetDimensions>
        <gml:usesParameter>
            <OperationParameter gml:id="EPSG8801">
                <gml:parameterName>Latitude of natural
origin</gml:parameterName>
                <gml:parameterID>
                    <gml:name
codeSpace="urn:ogc:def:parameter:EPSG:6.0:">8801</gml:name>
                    </gml:parameterID>
                </OperationParameter>
            </gml:usesParameter>
            <gml:usesParameter>
                <OperationParameter gml:id="EPSG8802">
                    <gml:parameterName>Longitude of natural
origin</gml:parameterName>
                    <gml:parameterID>
                        <gml:name
codeSpace="urn:ogc:def:parameter:EPSG:6.0:">8802</gml:name>
                        </gml:parameterID>
                    </OperationParameter>
                </gml:usesParameter>
            </gml:usesParameter>
            <OperationParameter gml:id="EPSG8805">
                <gml:parameterName>Scale factor at natural
origin</gml:parameterName>
                <gml:parameterID>
                    <gml:name
codeSpace="urn:ogc:def:parameter:EPSG:6.0:">8805</gml:name>
                    </gml:parameterID>
                </OperationParameter>
            </gml:usesParameter>
            <gml:usesParameter>
                <OperationParameter gml:id="EPSG8806">
                    <gml:parameterName>False Easting</gml:parameterName>
                    <gml:parameterID>
                        <gml:name
codeSpace="urn:ogc:def:parameter:EPSG:6.0:">8806</gml:name>
                        </gml:parameterID>
                    </OperationParameter>
                </gml:usesParameter>
            </gml:usesParameter>
            <OperationParameter gml:id="EPSG8807">
                <gml:parameterName>False Northing</gml:parameterName>
                <gml:parameterID>
                    <gml:name
codeSpace="urn:ogc:def:parameter:EPSG:6.0:">8807</gml:name>
                    </gml:parameterID>
                </OperationParameter>
            </gml:usesParameter>
        </gml:OperationMethod>
    </method>
</MethodDescription>

```

14.4.2 User defined orthorectification method example

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

14.4.2.1 DescribeMethod request

A DescribeMethod operation request for a user defined orthorectification operation method can look like this encoded in KVP:

```
http://www.lat-lon.de/transform&service=WCTS&request=DescribeMethod
&version=0.0.0&methods=urn:ogc:def:method:OGC:0.0:XXXX&
```

The corresponding DescribeMethod operation request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<DescribeMethod xmlns="http://www.opengeospatial.net/wcts"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsDescribeMethod.xsd" service="WCTS" version="0.0.0">
  <Method>urn:ogc:def:method:OGC:0.0:XXXX</Method>
</DescribeMethod>
```

14.4.2.2 DescribeMethod response

If no exception occurs, the server will reply to a DescribeMethod operation request with a list of one or more operation method definitions. If a user defined orthorectification operation method is requested, the response might be:

```
<?xml version="1.0" encoding="UTF-8"?>
<MethodDescription xmlns="http://www.opengeospatial.net/wcts"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:ows="http://www.opengeospatial.net/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.net/wcts
../Schemas/wctsDescribeMethod.xsd http://www.opengis.net/gml
../gml/3.1.1/base/coordinateOperations.xsd">
  <method>
    <gml:OperationMethod gml:id="_SPOT_SRIT">
      <gml:methodName>SPOT SRIT
(Orthorectification)</gml:methodName>
      <gml:methodID>
        <gml:name
codeSpace="http://www.pcigeomatics.com/ontology/opengis/services/wcts/w
arper#">SPOT_SRIT</gml:name>
        <gml:version>1.0.0</gml:version>
      </gml:methodID>
      <gml: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.

```

</gml:remarks>
  <gml:methodFormula>CCRS SRIT Model</gml:methodFormula>
  <gml:sourceDimensions>2</gml:sourceDimensions>
  <gml:targetDimensions>2</gml:targetDimensions>
  <gml:usesParameter
xlink:href="urn:pcigeomatics:def:parameter:pci:1.0.0:BANDS">
    <OperationParameter gml:id="_bands">
      <gml:parameterName>List of channels</gml:parameterName>
      <gml:parameterID>
        <gml:name
codeSpace="http://www.pcigeomatics.com/ontology/opengis/services/wcts/w
arper#">bands</gml:name>
        <gml:version>1.0.0</gml:version>
      </gml:parameterID>
      <gml:remarks>Comma delimited list of channel number
(index starting at 1)</gml:remarks>
      <allowedValues
datatype="http://www.opengis.org/gml#integerList">
        <ows:Range>
          <ows:MinimumValue>1</ows:MinimumValue>
        </ows:Range>
      </allowedValues>
    </OperationParameter>
  </gml:usesParameter>
  <gml:usesParameter
xlink:href="urn:pcigeomatics:def:parameter:pci:1.0.0:DIMAP_URL">
    <OperationParameter gml:id="_DIMAP_URL">
      <gml:parameterName>DIMAP URL</gml:parameterName>
      <gml:parameterID>
        <gml:name
codeSpace="http://www.pcigeomatics.com/ontology/opengis/services/wcts/w
arper#">DIMAP_URL</gml:name>
        <gml:version>1.0.0</gml:version>
      </gml:parameterID>
      <gml:remarks>DIMAP Metadata URL for SPOT Imagery
typically metadata.dim file</gml:remarks>
      <allowedValues
datatype="http://www.w3.org/2001/XMLSchema#anyURI">
        <ows:Value>TBD</ows:Value>
      </allowedValues>
    </OperationParameter>
  </gml:usesParameter>
  <gml:usesParameter
xlink:href="urn:pcigeomatics:def:parameter:pci:1.0.0:pixel_size">
    <OperationParameter gml:id="_pixel-size">
      <gml:parameterName>Pixel X, Y output
size</gml:parameterName>
      <gml:parameterID>
        <gml:name
codeSpace="http://www.pcigeomatics.com/ontology/opengis/services/wcts/w
arper#">pixel-size</gml:name>
        <gml:version>1.0.0</gml:version>
      </gml:parameterID>
      <gml: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</gml:remarks>

```

```

        <allowedValues
datatype="http://www.opengis.org/gml#floatList">
        <ows:Range>
            <ows:MinimumValue>0.001</ows:MinimumValue>
        </ows:Range>
        </allowedValues>
    </OperationParameter>
</gml:usesParameter>
<gml:usesParameter
xlink:href="urn:pcigeomatics:def:parameter:pci:1.0.0:DEM_URL">
    <OperationParameter gml:id="_DEM_URL">
        <gml:parameterName>Elevation (DEM) datafile
URL</gml:parameterName>
        <gml:parameterID>
            <gml:name
codeSpace="http://www.pcigeomatics.com/ontology/opengis/services/wcts/w
arper#">DEM_URL</gml:name>
            <gml:version>1.0.0</gml:version>
        </gml:parameterID>
        <gml:remarks>Digital elevation model (DEM) url used for
the orthorectification process</gml:remarks>
        <allowedValues
datatype="http://www.w3.org/2001/XMLSchema#anyURI">
            <ows:Range/>
        </allowedValues>
    </OperationParameter>
</gml:usesParameter>
</gml:OperationMethod>
</method>
</MethodDescription>

```

NOTE The orthorectified image resolution in each direction must 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 levels 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

Both KVP and XML encoding of the GetCapabilities operation request shall be tested, if the server implements XML encoding of the GetCapabilities request. 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.

Both XML and KVP encoding of the Transform operation request shall be tested, if the server implements KVP encoding of the Transform request. 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, especially 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.

Both XML and KVP encoding of the IsTransformable operation request shall be tested, if the server implements KVP encoding of the IsTransformable request. 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.

Both KVP and XML encoding of the GetTransformation operation request shall be tested, if the server implements XML encoding of the GetTransformation request. 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 DescribeTransformation operation testing

If implemented, the DescribeTransformation 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 identifier of each selected transformation.

Both KVP and XML encoding of the DescribeTransformation operation request shall be tested, if the server implements XML encoding of the DescribeTransformation request. 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 DescribeTransformation 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.8 DescribeCRS operation testing

If implemented, the DescribeCRS operation shall be tested by using that operation to retrieve a variety of CRS definitions, singly and in groups. These examples shall be selected from the sets of the source and target CRSs implemented by the server, using the identifier of each selected CRS.

Both KVP and XML encoding of the DescribeCRS operation request shall be tested, if the server implements XML encoding of the DescribeCRS request. 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 DescribeCRS operation requested shall be checked for the proper correspondence to the operation inputs, depending on the specific source and target CRSs implemented. Each response shall also be checked to ensure that it is valid with respect to the XML Schema definition of that response.

A.9 DescribeMethod operation testing

If implemented, the DescribeMethod operation shall be tested by using that operation to retrieve a variety of operation method definitions, singly and in small groups. These examples shall be selected from the set of the operation methods which can be used in user-defined transformations that are implemented by the server, using the identifier of each selected method.

Both KVP and XML encoding of the DescribeMethod operation request shall be tested, if the server implements XML encoding of the DescribeMethod request. 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 DescribeMethod operation requested shall be checked for the proper correspondence to the operation inputs, depending on the specific operation methods 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)

XML schemas

In addition to this document, this specification includes normative XML Schema files. These XML Schema files are bundled in a zip file with the present document.

The coordinate transformation abilities now specified in this document use 12 specified XML Schemas included in the zip file with this document. These XML Schemas combine the XML Schema fragments listed in various subclauses of this document, eliminating duplications. These XML Schema files roughly match the nine UML packages defined in Annex E, and are named:

- a) wctsGetCapabilities.xsd
- b) wctsTransform.xsd
- c) wctsIsTransformable.xsd
- d) wctsGetAndDescribeTransformation.xsd
- e) wctsDescribeCRS.xsd
- f) wctsDescribeMethod
- g) wctsOperationMethod
- h) wctsContents.xsd
- i) wctsCommon.xsd
- j) wctsAll.xsd
- k) owsAdditions.xsd
- l) owsDomainType.xsd
- m) owsReferenceTypeUsingGML.xsd

The last three XML Schemas listed above are used by multiple OWS interface specifications, and are thus specified in the “ows” namespace.

These XML Schemas use and build on the OWS common XML Schemas listed in Annex A of [OGC 05-008], specified in the “ows” namespace and named:

- a) ows19115subset.xsd
- b) owsCommon.xsd
- c) owsExceptionReport.xsd
- d) owsGetCapabilities.xsd
- e) owsOperationsMetadata.xsd

- f) owsServiceIdentification.xsd
- g) owsServiceProvider.xsd

Many of these XML Schemas use the six CRS-related XML Schemas for GML 3.1.1, current drafts in [OGC 04-092r4], named

- a) coordinateReferenceSystems.xsd
- b) coordinateSystems.xsd
- c) datums.xsd
- d) coordinateOperations.xsd
- e) dataQuality.xsd
- f) referenceSystems.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 05-008].

Annex C (informative)

Use cases

C.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

C.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 C.1 shows a diagram for this use case.

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

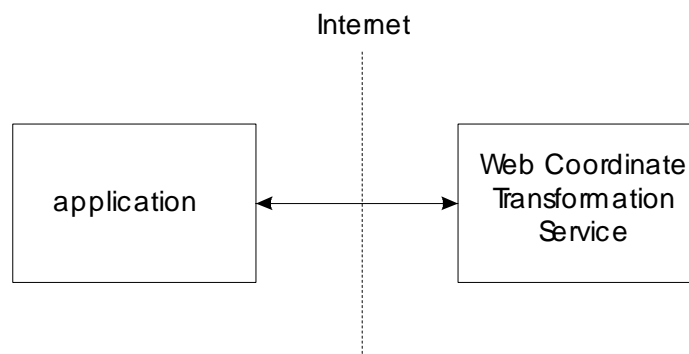


Figure C.1 — Desktop Application use case: application - WCTS interaction

The following steps have to be performed in this use case:

- 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 using the transformed data.

C.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 C.2 shows a diagram for this use case.

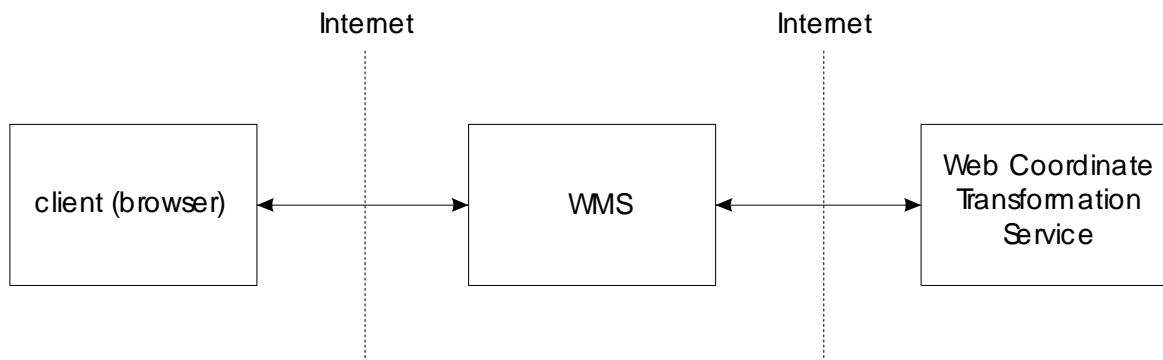


Figure C.2 — WMS use case: client - WMS - WCTS interaction

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)

C.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 C.3 shows a diagram for this use case.

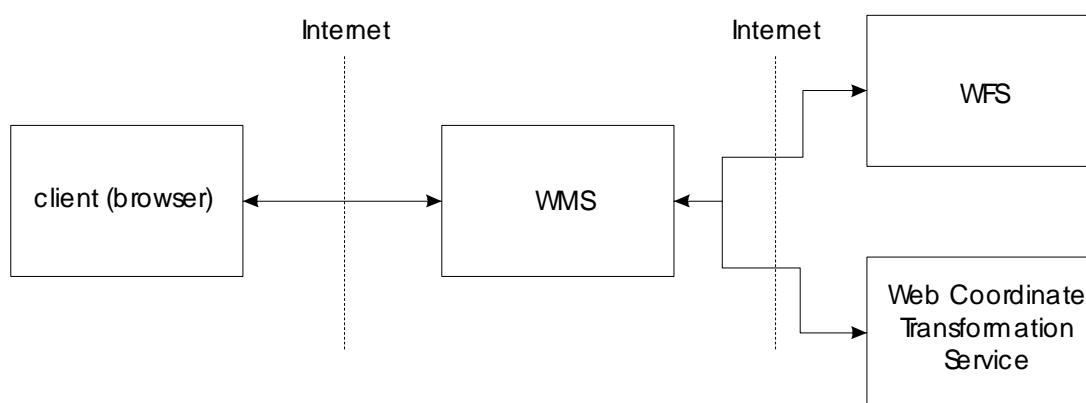


Figure C.3 — Service Chain use case: client - WMS - WFS - WCTS interaction

In this use case, the WMS sends a request to the WCTS, providing the input data as a getFeature-Request for 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 C.4 shows a diagram for this use case.

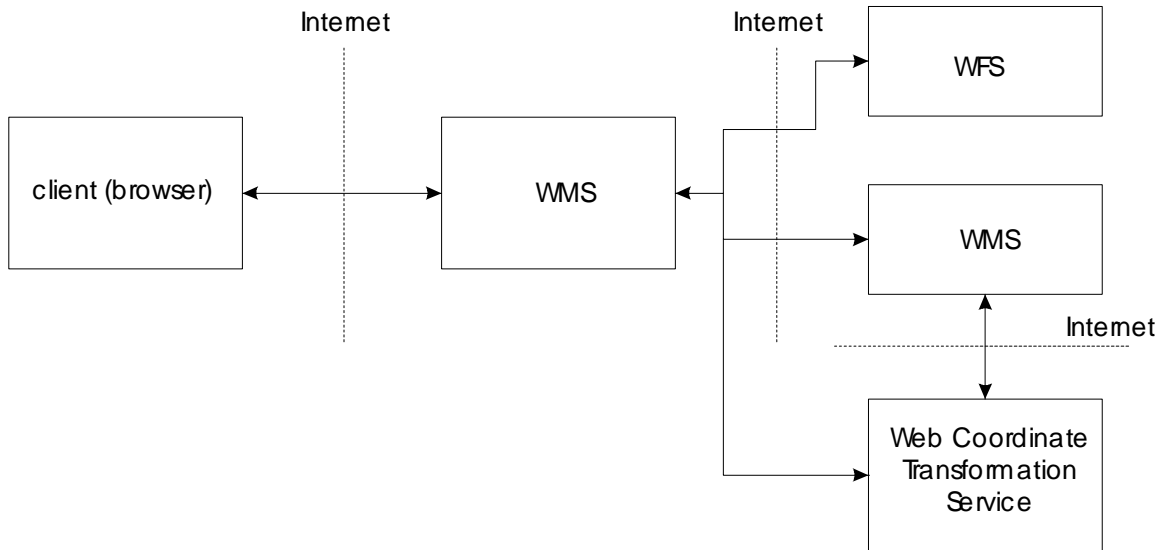


Figure C.4 — Service Chain use case two4: service chain

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).

C.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 a rational function transformation, an orthorectification of imagery, which is a complex math transform by taking in account sensor information, digital Elevation Model (DEM) and maybe ground control points.

The operation method can publish the list of parameters as well as its valid values (for example URI for DEM or SensorML resource, coefficient of equations etc....) using the DescribeMethod operation.

Flow of events

- A user wants to perform a custom transformation based on a well-known method.
- The user performs a GetCapabilities request to the WCTS to retrieve the list of supported methods.
- Based on the brief descriptions of the method or its well-known identifiers, the user selects the operation method of interest.
- The user invokes describeMethod operation to the WCTS by providing the methodID published in the Contents section of the capability.

- 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.
- The parameters description is used to build a form on the client side, which the parameters value to validate the user inputs.
- The client build 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.
- The Server process the query and performs the transformation.
- The WCTS sends the response back.

Annex D (informative)

Example XML documents

D.1 Introduction

This annex provides more example XML documents than given in the body of this document. **TBD**

D.2 TBD

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 [05-008].

Figure E.1 is a simple UML diagram summarizing the WCTS interface. This class diagram shows that the WCTService class inherits the getCapabilities operation from the OGCWebService interface class, and adds the “transform”, isTransformable, GetTransformation, DescribeTransformation, DescribeCRS, and DescribeMethod operations. (The capitalization of names uses the OGC/ISO profile of UML.)

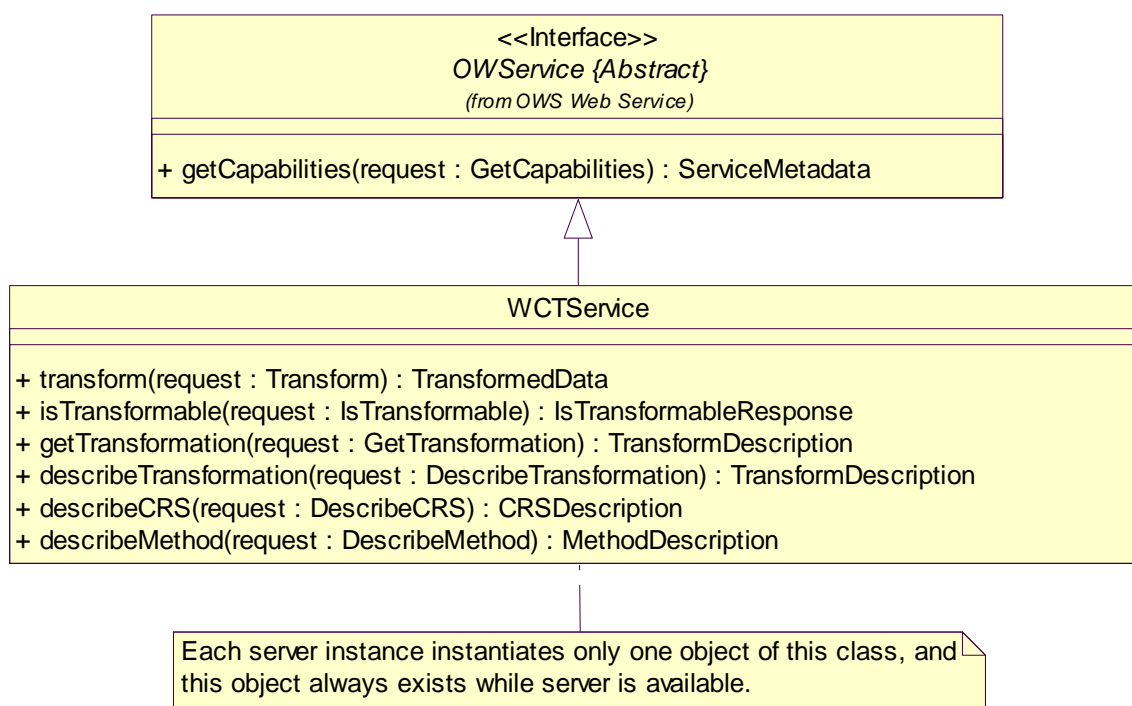


Figure E.1 — WCTS interface UML diagram

Each of the seven operations uses a request and a response data structure, each of which can be 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 Clauses 6 through 14.

E.2 UML packages

The WCTS interface UML model is organized in ten packages, as shown in the package diagram in Figure E.2. These WCTS-specific packages make use of six non-WCTS-specific packages, named OWS Web Service, OWS Operations Metadata, OWS Service Identification, OWS Service Provider, ISO 19115 Subset, and GML Subset. 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 04-046r3], here considered part of the GML Subset package. This package diagram shows the dependencies among the various packages shown.

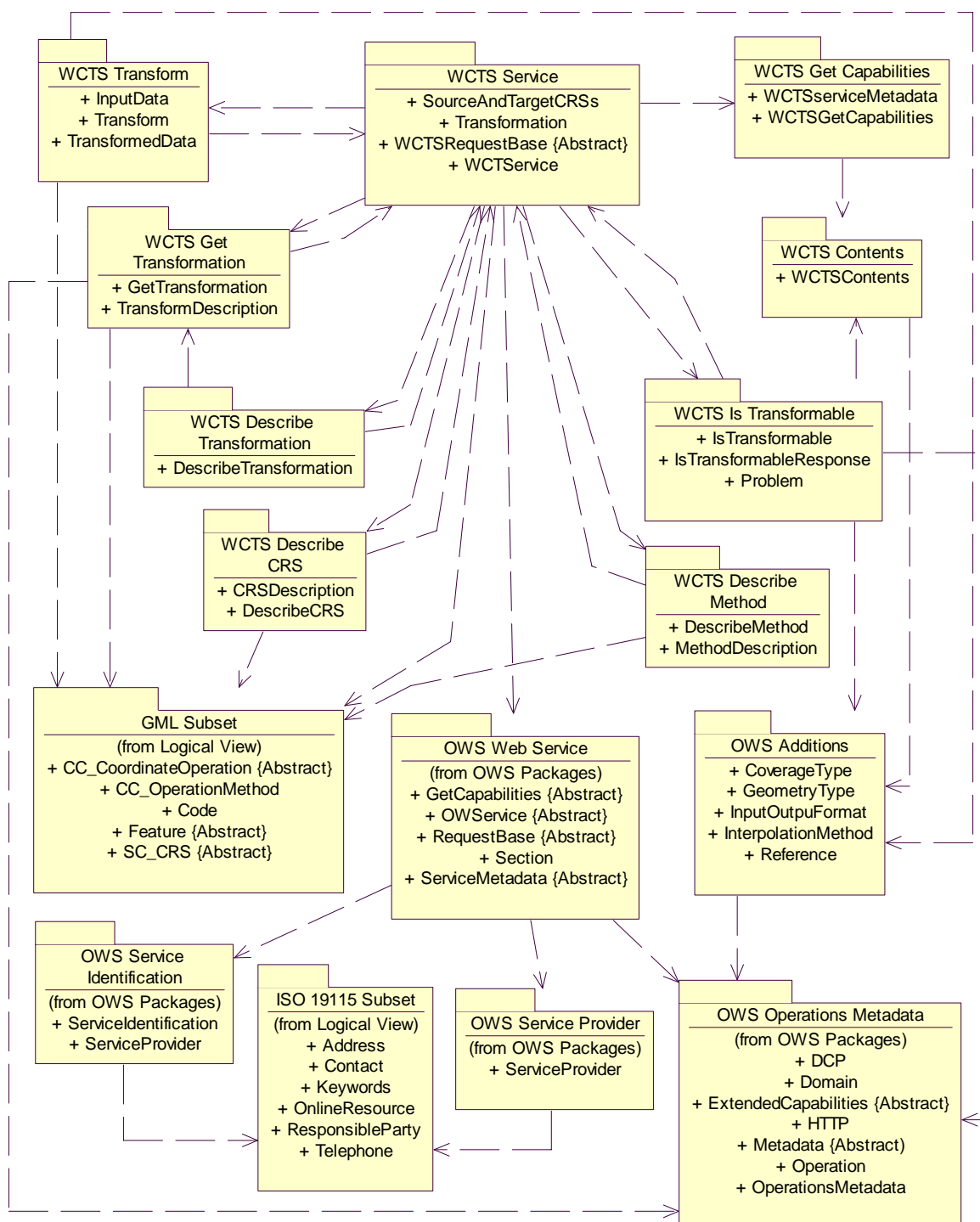


Figure E.2 — WCTS interface package diagram

Each of the WCTS-specific packages shown in Figure E.2 is described in the following subclauses. The OWS Web Service, OWS Operations Metadata, OWS Service Identification, OWS Service Provider, and ISO 19115 Subset packages are described in Annex B of [OGC 05-008]. Most of the GML Subset package is described in the UML model in Clauses 7-13 of [OGC 04-046r3]. This package diagram does not show the

WCTS Operation Method and Parameter Values packages. The WCTS Describe Method package also depends on the WCTS Operation Method package, which depends on the Parameter Values package.

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 seven operation requests and responses, which are shown (with part of this package) in the Transform, Is Transformable, Get Transformation, Describe Transformation, Describe CRS, Describe Method, 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 two used classes from the GML Subset package. The SourceAndTargetCRSs and Transformation classes introduced by this package are further defined by Table 1 and Table 2 in this document.

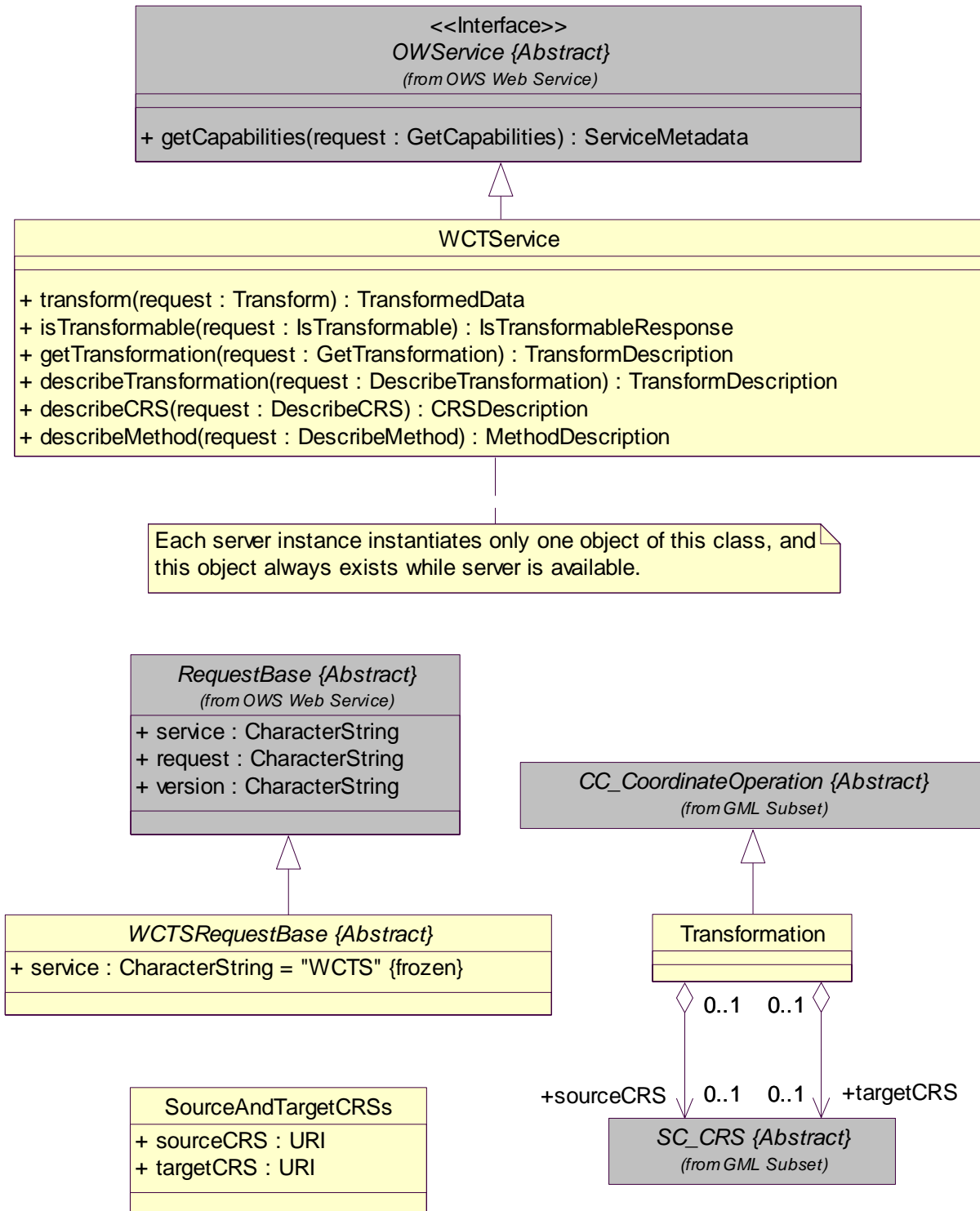


Figure E.3 — WCTS Service package class diagram

E.4 WCTS Transform package

The Transform package is shown in the class diagram in Figure E.4. This diagram also shows two classes of the WCTS Service package plus several classes from the OWS Web Service and GML Subset packages. The Transform and TransformedData classes introduced by this package are further defined by Table 12 and Table 15 in this document.

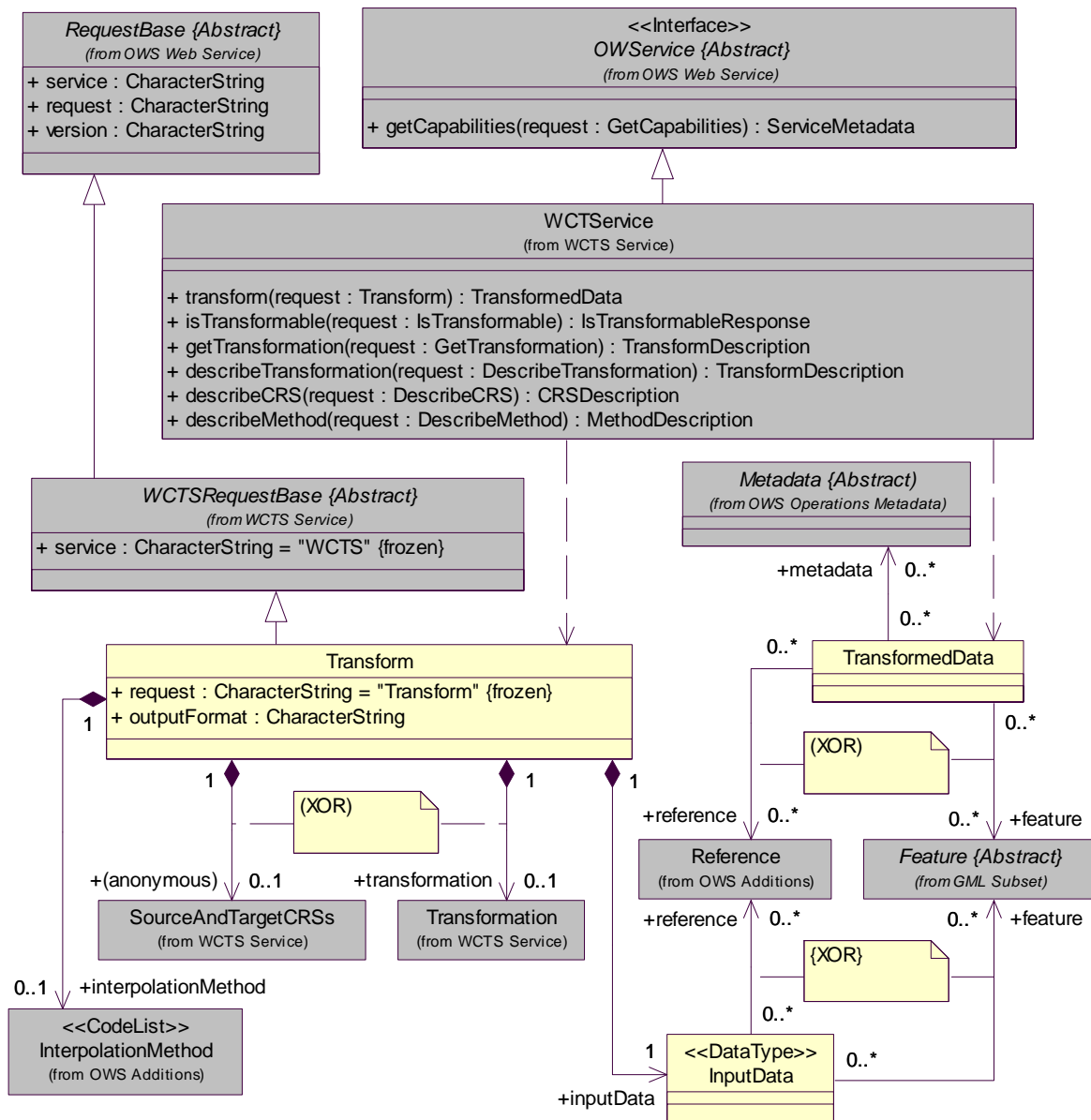


Figure E.4 — WCTS Transform package class diagram

E.5 WCTS Is Transformable package

The Is Transformable package is shown in the class diagram in Figure E.5. This diagram also shows two classes of the WCTS Service package plus several used classes from the OWS Web Service and GML Subset packages. The three classes introduced by this package are further defined in Table 17 and Table 21 in this document.

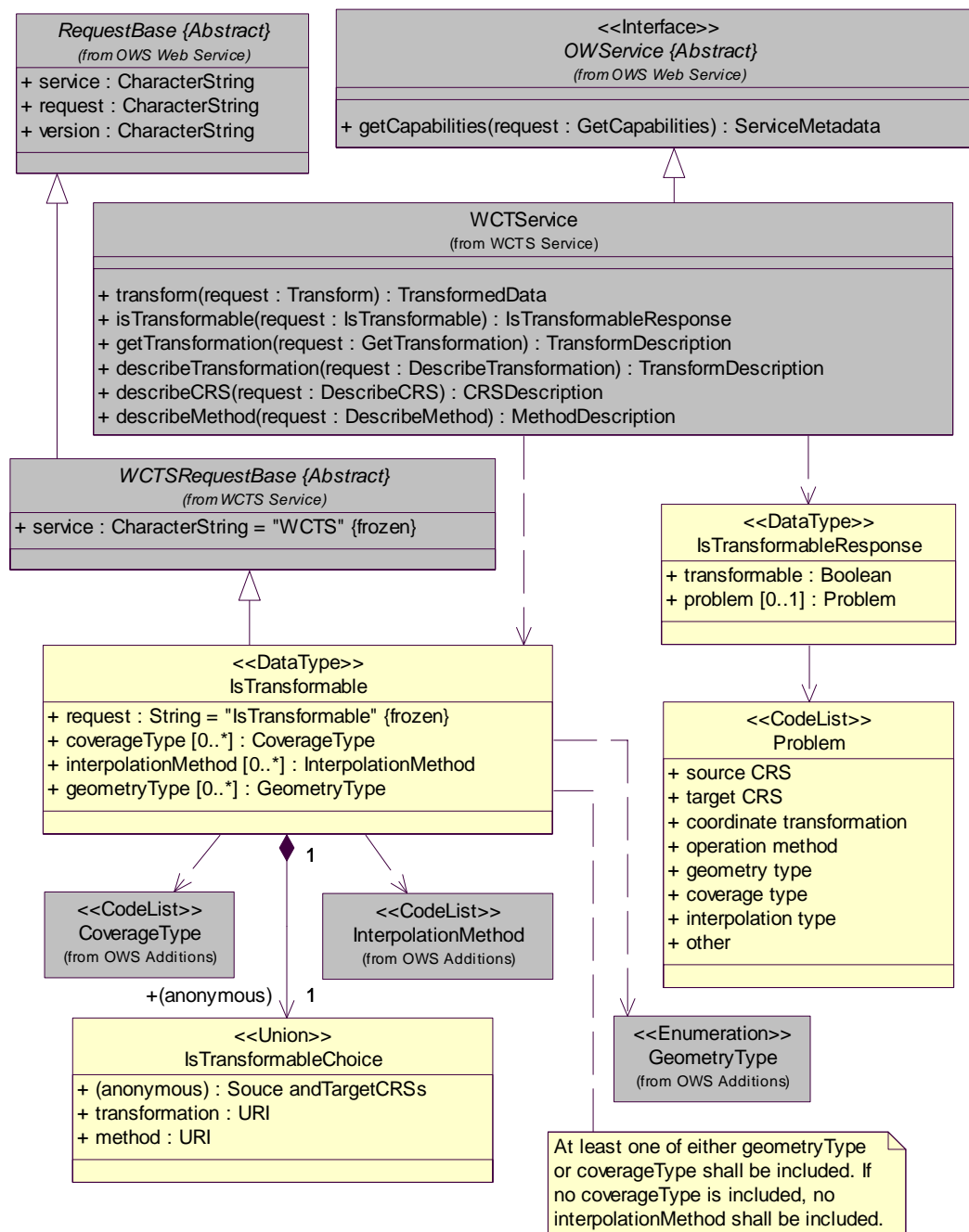


Figure E.5 — WCTS Is Transformable package class diagram

E.6 WCTS Get Transformation package

The Get Transformation 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 the OWS Web Service and GML Subset packages. The GetTransformation and TransformDescription classes introduced by this package are further defined by Table 22 and Table 24 in this document.

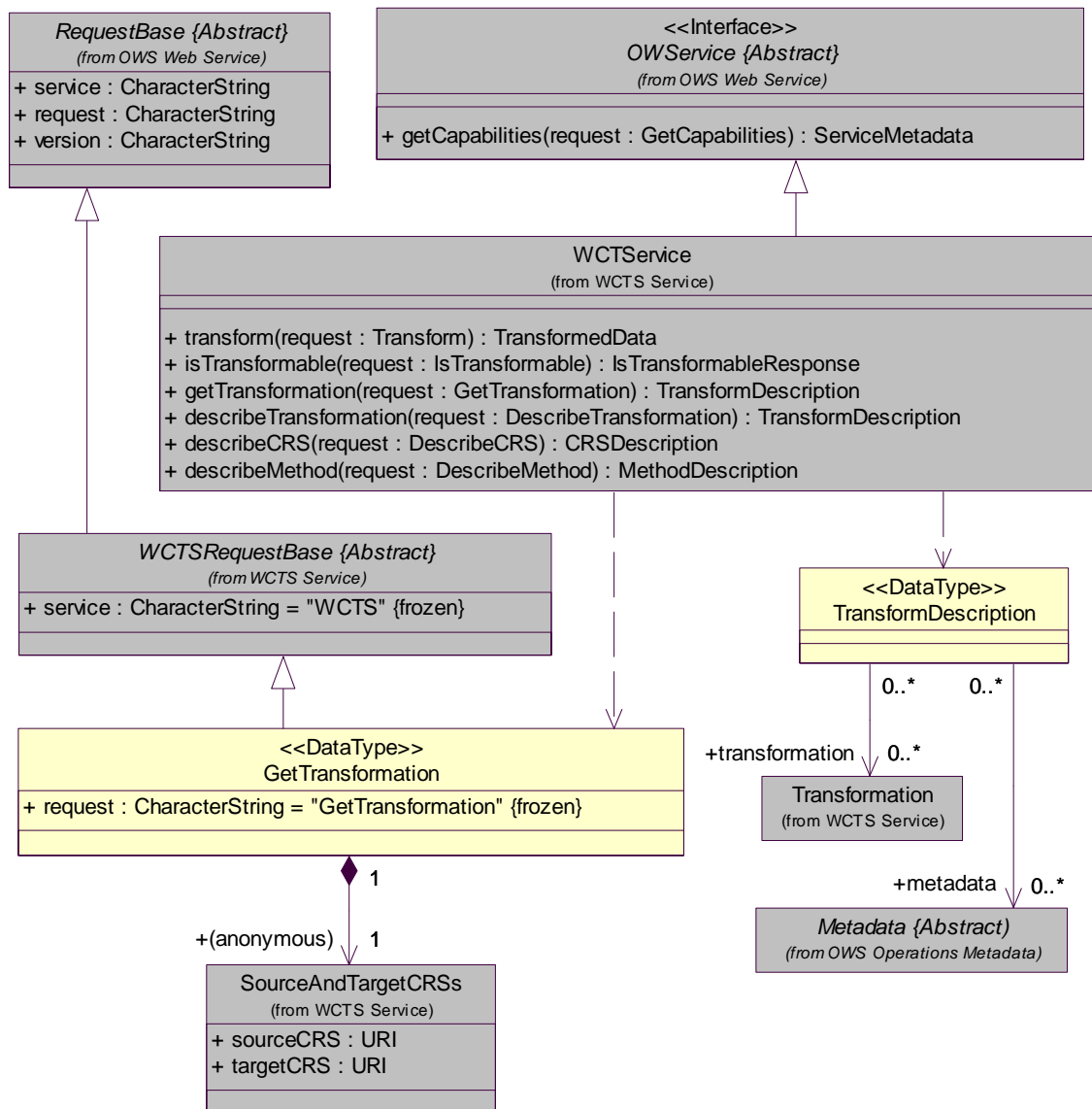


Figure E.6 — WCTS Describe Transformation package class diagram

E.7 WCTS Describe Transformation package

The Describe 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 the OWS Web Service and GML Subset packages. The DescribeTransformation class introduced by this package is further defined by Table 25 in this document.

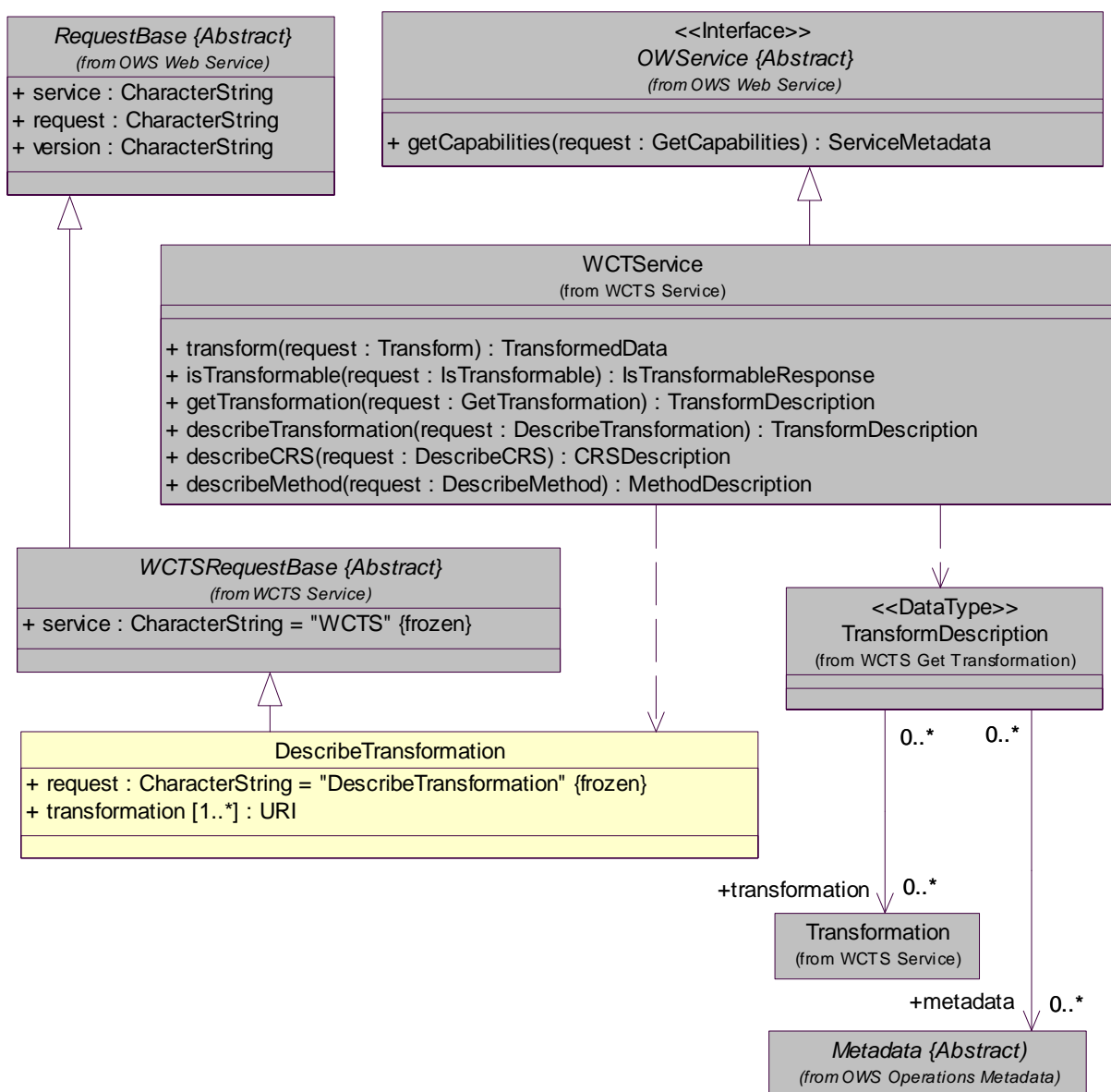


Figure E.7 — WCTS Describe Transformation package class diagram

E.8 WCTS Describe CRS package

The Describe CRS package is shown in the class diagram in Figure E.8. This diagram also shows two classes of the WCTS Service package plus several used classes from the OWS Web Service and GML Subset packages. The DescribeCRS and CRSDescription classes introduced by this package are further defined by Table 28 and Table 30 in this document.

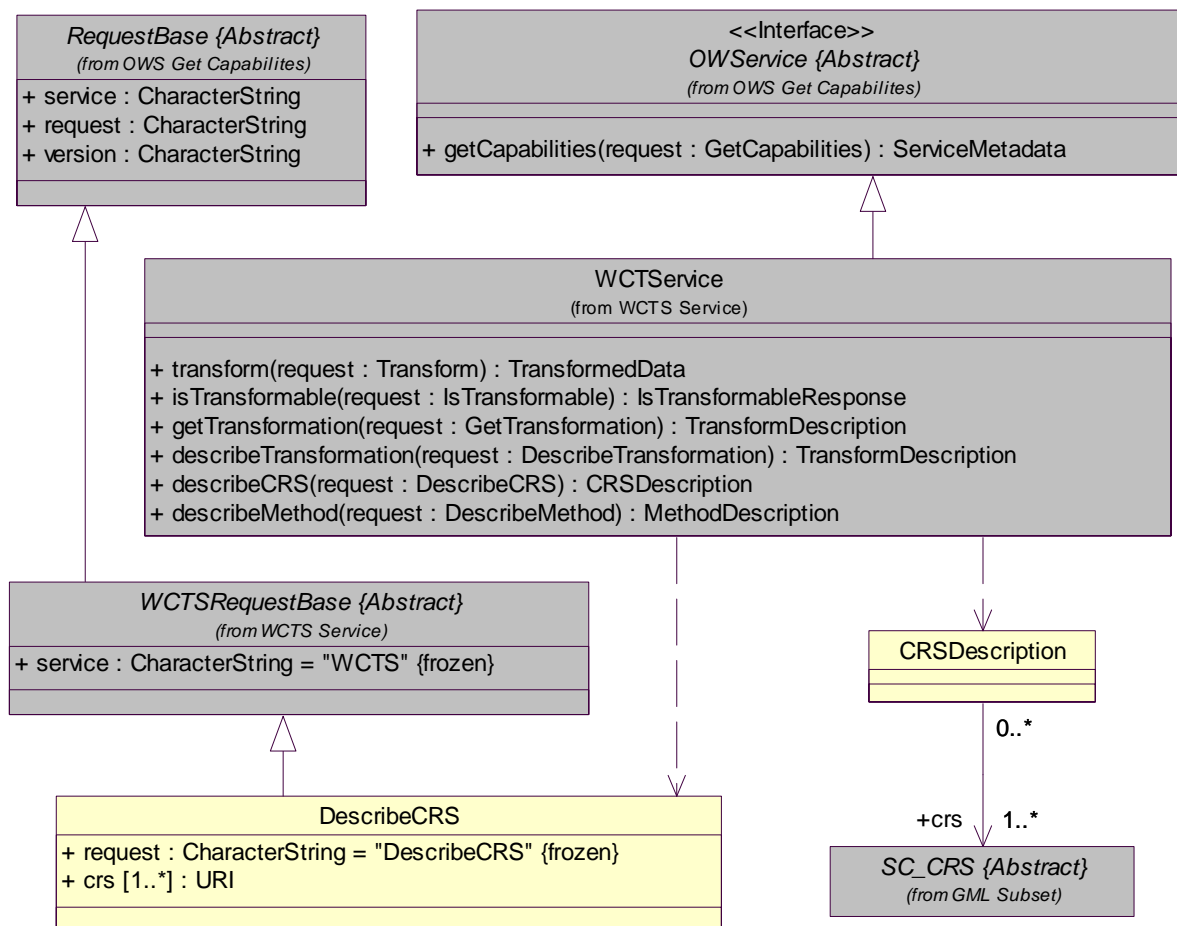


Figure E.8 — WCTS Describe CRS package class diagram

E.9 WCTS Describe Method package

The Describe Method package is shown in the class diagram in Figure E.9. This diagram also shows two classes of the WCTS Service package plus the OperationMethod class used from the WCTS Operation Method package, described in Subclause E.13. The DescribeMethod and MethodDescription classes introduced by this package are further defined by Table 31 and Table 33 in this document.

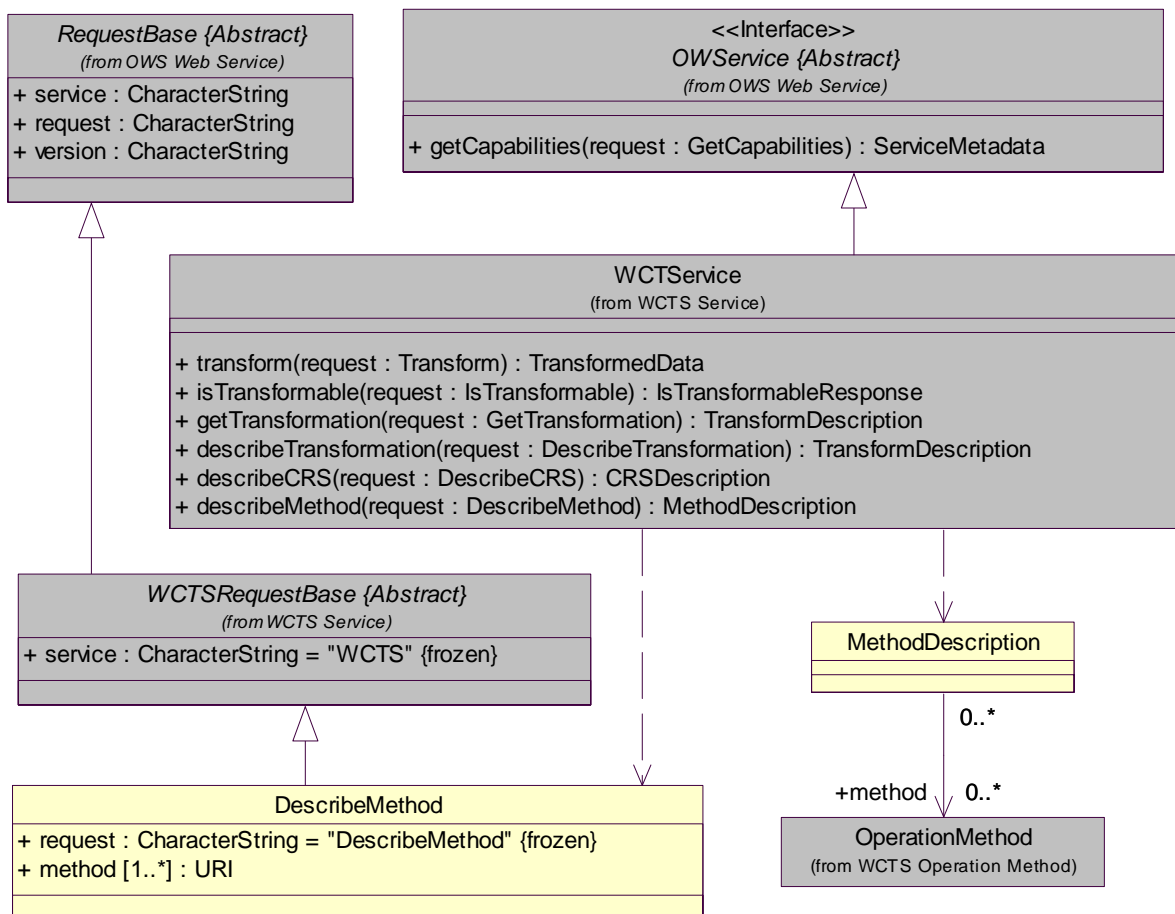


Figure E.9 — WCTS Describe Method package class diagram

E.10 WCTS Get Capabilities package

The WCTS Get Capabilities package is shown in the class diagram in Figure E.10. 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 05-008]. This diagram also shows the WCTSGetCapabilities and WCTSServiceMetadata classes of the WCTS Service package.

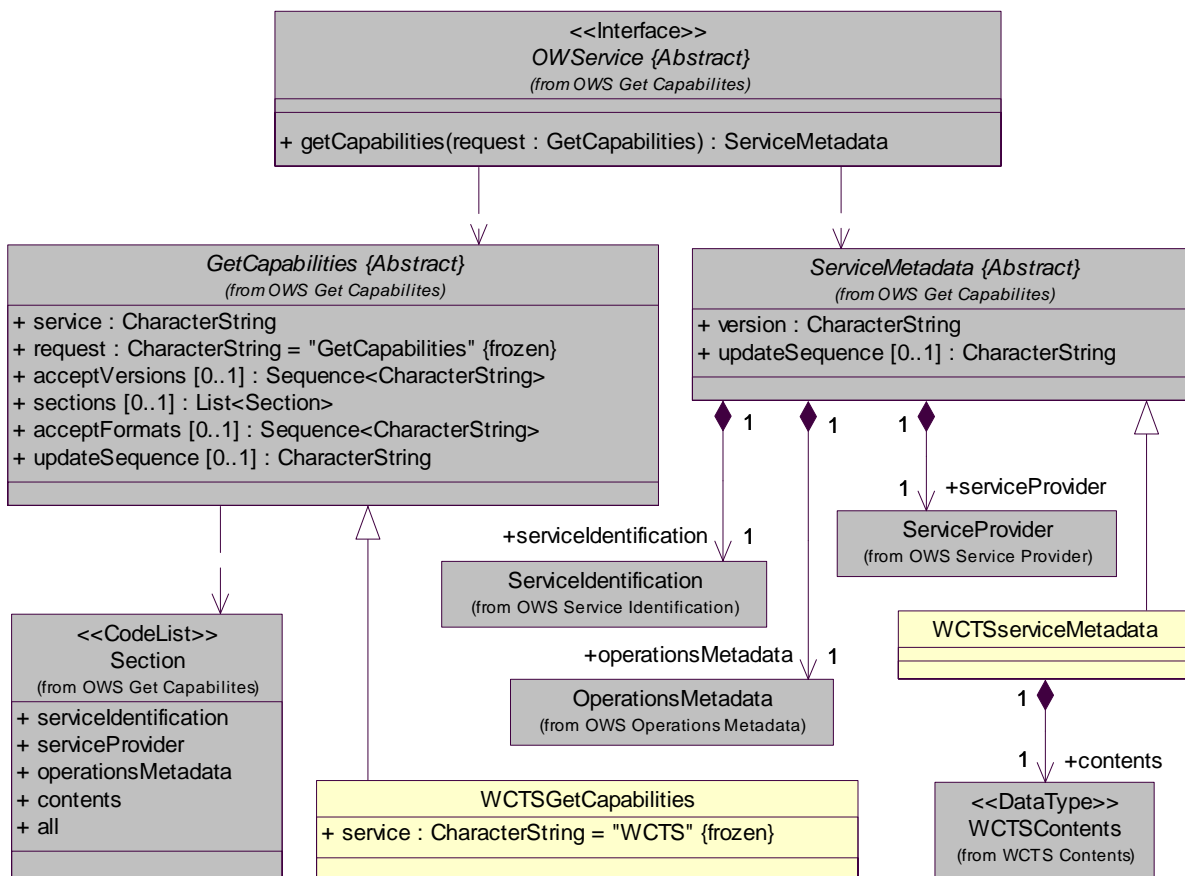


Figure E.10 — WCTS Get Capabilities package class diagram

E.11 WCTS Contents package

The WCTS Contents package is shown in the class diagram in Figure E.11. This diagram also shows one class from the WCTS Get Capabilities Contents package. The WCTSContents, CoverageAbilities, and FeatureAbilities classes introduced by this package are further defined by Table 8 through Table 11 in this document. This package uses four classes that are specified in the OWS Additions package that is described in the following subclause.

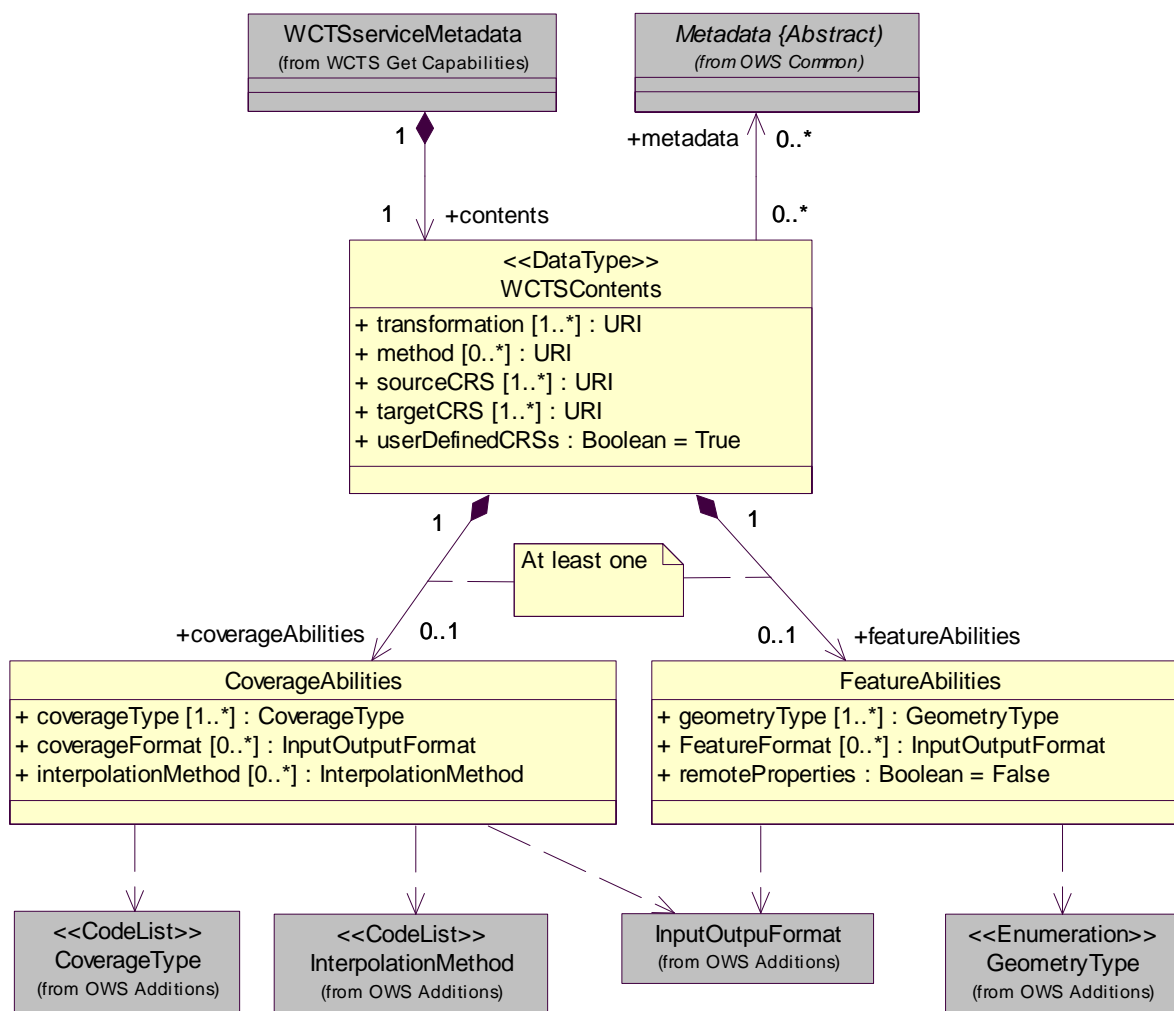


Figure E.11 — WCTS Contents package class diagram

E.12 OWS Additions package

The OWS Additions package is shown in the class diagram in Figure E.12. This package is defined separately since all its classes are expected to be used by multiple OGC Web Services, and are thus expected to be considered for addition to the OWS Common Specification.

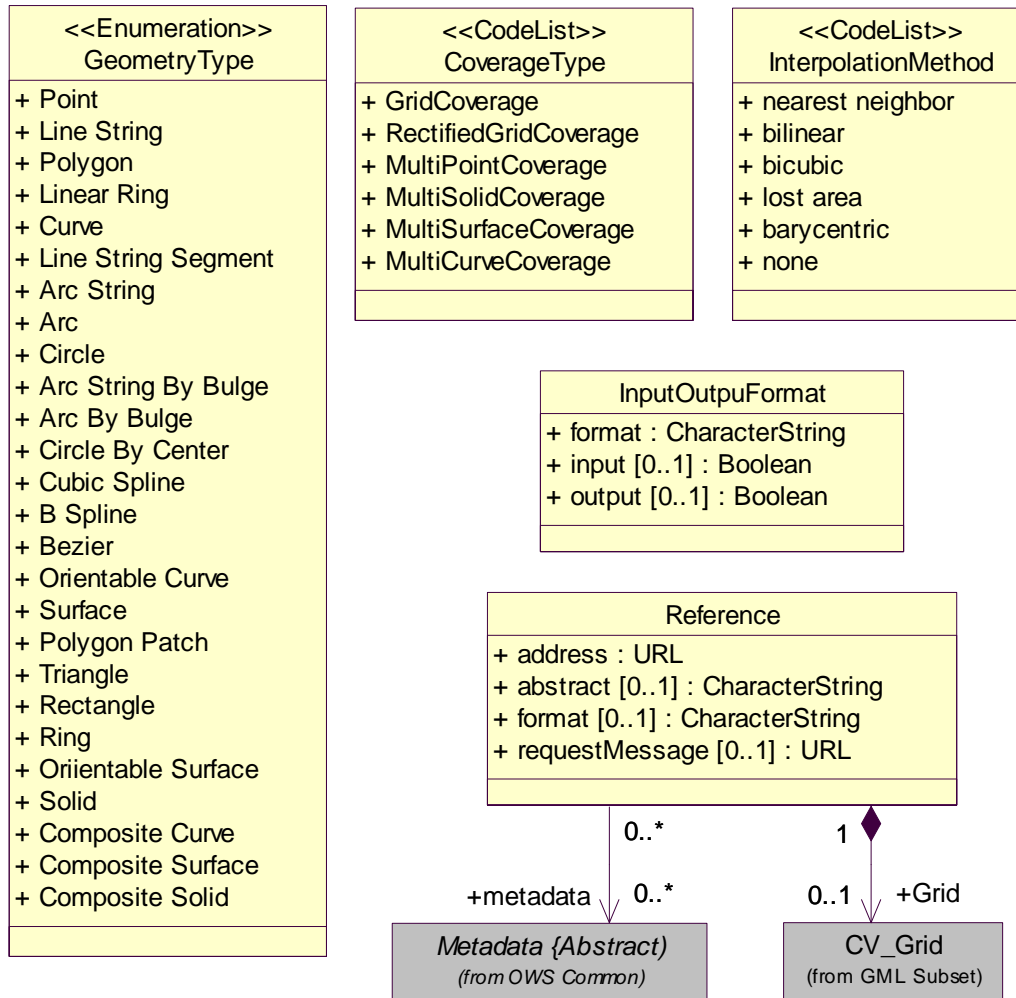


Figure E.12 — OWS Additions package class diagram

EDITOR'S NOTE The listed values for the CoverageType <<CodeList>> were derived from GML 3, and are not fully consistent with ISO 19123. This difference needs to be resolved.

E.13 WCTS Operation Method package

The WCTS Operation Method package is shown in the class diagram in Figure E.13. This package builds on the CC Coordinate Operation package of OGC Abstract Specification Topic 2 [OGC 04-046r3], which is included in the GML Subset package. This package uses the UnNamedDomain and Value classes from the Parameter Values package, which is described in the following subclause.

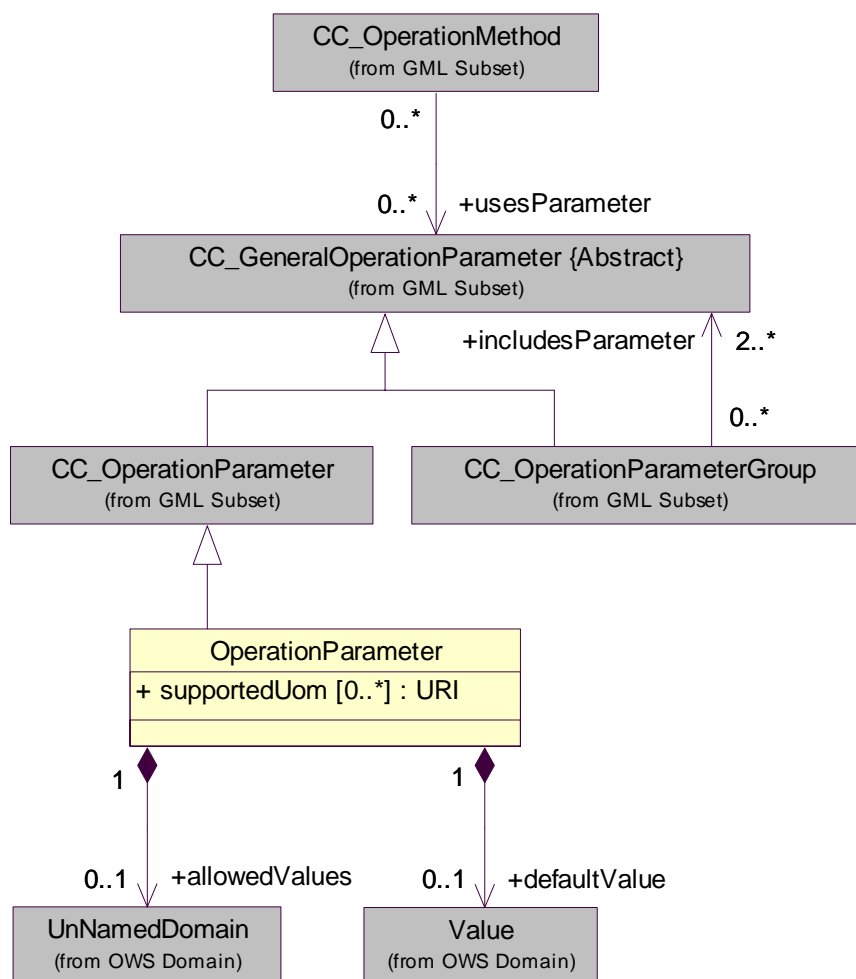


Figure E.13 — WCTS Operation Method package class diagram

E.14 Domain package

The Domain package is shown in the class diagram in Figure E.14. This package is similar to the Values package in the WCS Implementation Specification version 1.0 [OGC 03-065r6].

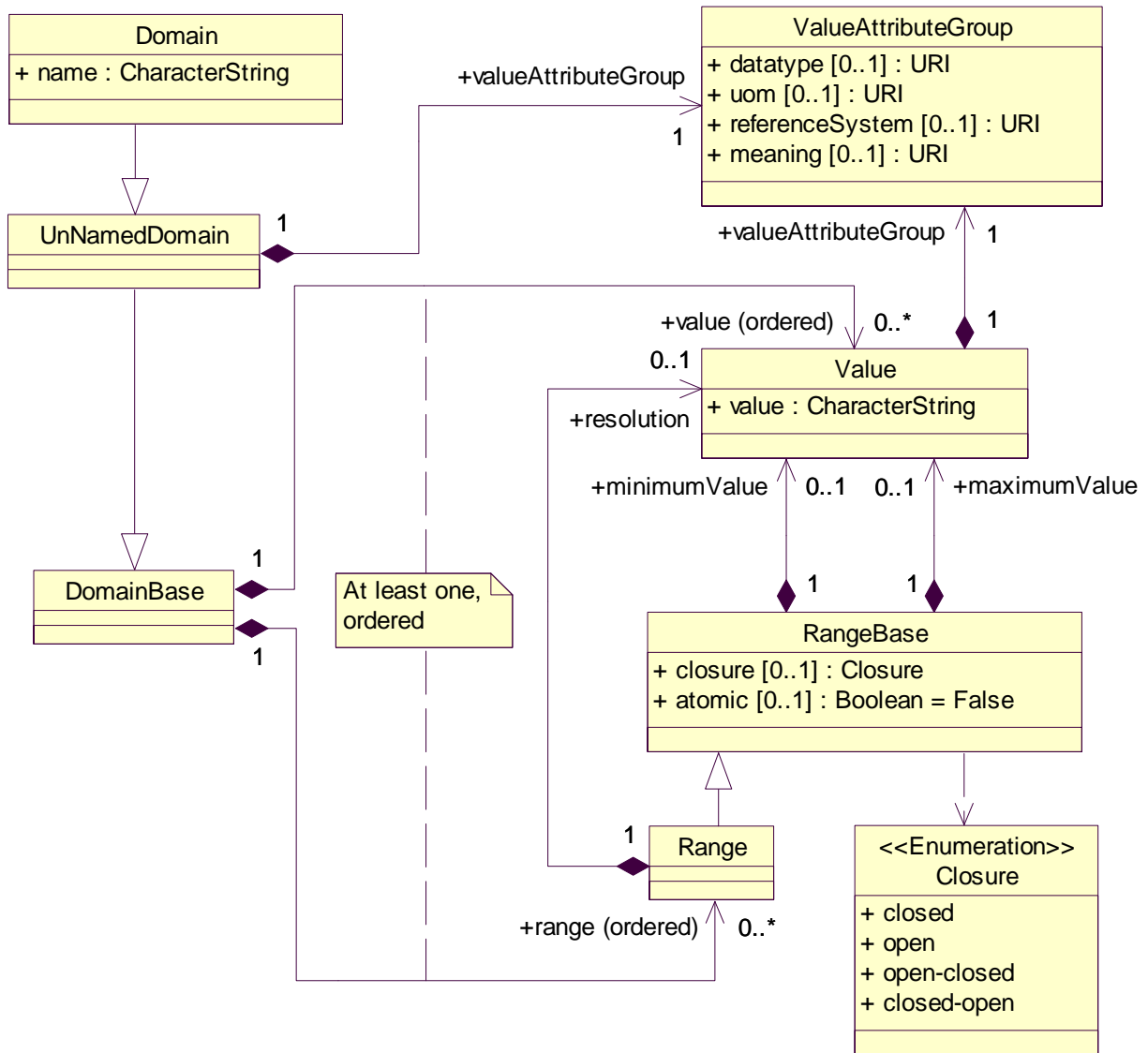


Figure E.14 — Domain package class diagram

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 9.4.1, 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 9.4.1 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.opengeospatial.net/wcts"
xmlns:wcts="http://www.opengeospatial.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>
      <description>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
2004/12/20</description>
    </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.opengeospatial.net/wcts"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengeospatial.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 must 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 (informative)

Design decisions

G.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 Needs to be updated based on recent changes to this document.

G.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) Use specified CRS Application Schemas for some well-known coordinate operations.
- f) 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.
- g) 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.)
- h) When the IsTransformable operation returns "false", return a list of one or more specified problem identifier text strings, where identifiers are specified for the common error conditions.
- i) Support one or more of either Feature, File, or Remote OWS (e.g., WFS) inputs in one Transform operation request.

- j) 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.
- k) Include a "service" parameter in all operation requests, with a fixed string value of "WCTS".
- l) Include a "version" parameter of string type in all operation requests, except in GetCapabilities where it is optional, with the value being the version of the Implementation Specification (document) to which the requested operation conforms.
- m) Include a "request" parameter in all operation requests, with fixed string values corresponding to the operation names.
- n) Include an optional "section" parameter (or equivalent) in the GetCapabilities operation request, and specify several standard values and meanings of this parameter for retrieving major sections of the complete Capabilities document. (One purpose of this flexibility is to limit the size of capability documents for WCTS servers supporting a lot of coordinate systems and coordinate operations.)

G.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 allow transforming geometry elements separate from features.
- j) Not provide special abilities for transforming a single point or few points, not contained in a GML geometry or feature.

- k) 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.

G.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.

G.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 Profile. Normatively reference the applicable parts of GML 3.1.1 (04-092r4).
- b) Specify standard URN values for sets of well-known coordinate operations, coordinate reference systems, units, and operation methods with corresponding operation parameters. Specify these standard URN values in the Application Profile and perhaps in the Implementation Specification.
- c) Specify standardized XML documents for selected sets of “well-known” coordinate operations, operation methods, CRSs, and units, in the Application Profile and perhaps in the Implementation Specification.
- d) 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.

- e) Specify a “limited” WCTS profile that is easy to implement.
- f) Normatively reference parts of OWS Common Specification [05-008], instead of copying this material.

Bibliography

- [1] ISO 19111:2003, Geographic Information – Spatial Referencing by Coordinates
- [2] ISO 19118, Geographic information – Encoding
- [3] OGC 99-107, Abstract Specification Topic 7: The Earth Imagery Case
- [4] OGC 00-045r1, Draft RFC on Image Coordinate Transformations
- [5] OGC 01-004, Implementation Specification: Grid Coverage
- [6] OGC 01-009, Implementation Specification: Coordinate Transformation Services
- [7] OGC 01-013r1, High-Level Ground Coordinate Transformation Interface
- [8] OGC 02-058, Web Feature Service Implementation Specification, version 1.0.0, September 2002