OpenGIS Project Document 01-013r1

TITLE:	**High-Level Ground Coordinate Transformation Interface**	
AUTHOR:	**Name:**	**Arliss Whiteside**
	Address:	**BAE SYSTEMS Mission Solutions** **16250 Technology Drive** **San Diego, CA 92127**
	Phone:	**858-592-1608**
	Email:	**Arliss.Whiteside@baesystems.com**
DATE:	**February 27, 2001**	
CATEGORY:	**Discussion Paper**	

The attached document specifies a "high-level" interface that is an alternate to the current "low-level" Coordinate Transformation Services (CT) Implementation Specification, OGC document 01-009. This document ~~is a~~ was reviewed by the CT RWG in their February 2001 meeting, and has been revised based on suggestions received from meeting attendees Roel Nicolai, John Bobbitt, and Martin Daly.

This revision of this document is identified as a discussion paper instead of a change proposal as directed by the CT RWG, to reflect other possible uses. Those other possible uses include use in the OGC Interoperability Program and inclusion in a possible OGC Request For Comment (RFC). This revision includes a few but not all of the XML changes now being made in OGC document 01-014. This revision also includes two additional items in Annex H.~~change proposal for the Coordinate Transformation Services (CT) Implementation Specification, OGC document 00-007r4. This document proposes adding an alternate "high-level" interface to the current "low-level" CT interface now specified. This change proposal is intended for consideration by the CT RWG.~~

Those familiar with the format of OGC documents will notice that the attached ~~proposal~~ document uses a different format. This ~~proposal~~ document uses the tentative new format for OGC Implementation Specifications that is similar to the ISO document format. I think Mike Gould is accepting comments on this format.

~~This~~ The ~~change proposal~~ specifie~~ds a~~ high-level interface to Coordinate Transformation Service~~s, which~~ supports almost the same server functions as the current low-level interface. This high-level interface is estimated to make client software significantly easier to program, compared to the current low-level interface. This high-level interface is independent of the current low-level interface, except that almost the same XML and WKT data formats should be used by both interface levels.

~~Most of the material in this proposal was copied from OGC document 00-045: Draft RFC on Image Coordinate transformations (Implementation Specification). This proposal includes only the "ground coordinate transformation service" and related portions of document 00-045. Some of the material in this proposal was copied from OGC document~~

~~01-014: Recommended Definition Data for Coordinate Reference Systems and Coordinate Transformations. The material copied should be changed when document 01-014 is changed or superseded.~~

This ~~proposal~~ draft Implementation Specification is not yet complete. For example, this draft includes a CORBA profile of the general DCP-independent interface. ~~The current plan is to also include a DCOM profile of the general DCP-independent interface. However, that profile has not yet been generated.~~ Additional profiles could be developed by interested OGC members, but have not yet been generated. For example, DCOM and/or Web profiles could be developed.

This document is a revision of OGC document 00-049. This revision includes updated XML in Annexes F and G, plus updated term definitions in Clause 8, both based on ~~revised~~ OGC document 01-014. Note that OGC document 01-014 is still changing.

Open GIS Consortium Inc.

Date: 2001-02-27~~01-15~~

Reference number of this OpenGIS© project document: **OGC 01-013r1**

Version: n.n.~~3~~2

Category: OpenGIS© ~~Implementation Specification Change Proposal~~Discussion Paper

Author: Arliss Whiteside

OpenGIS© Interface — Coordinate Transformation Services — High Level Interface

Copyright notice

Warning

This paper presents a discussion of technology issues considered in a Special Interest Group of the Open GIS Consortium Technical Committee. The content of this paper is presented to create discussion in the geospatial information industry on this topic; the content of this paper is not to be considered an adopted specification of any kind. This paper does not represent the official position of the Open GIS Consortium nor of the OGC Technical Committee. ~~This 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.~~

Document type: OpenGIS© Publicly Available ~~Standard~~Discussion Paper
Document subtype: if applicable
Document stage: Draft
Document language: English

File name: 01-013r1.doc~~01-013r1 High Level CT.doc~~~~01-013 High Level CT.doc~~

~~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.~~

Contents

i. Preface

This ~~document is submitted to the OGC as a~~ discussion paper specifies a "high-level" interface that is an alternative to the current "low-level" Coordinate Transformation Services (CT) Implementation Specification, OGC document 01-009. ~~change proposal to the current "low-level" Coordinate Transformation Services (CT) Implementation Specification, now in OGC document 00-007r4.~~ This ~~document specifies an alternate "high-level" interface to be added to the current low-level CT interface. This change proposal is intended for consideration by the Coordinate Transformation (CT) Revision Working Group (RWG).~~

~~This change proposal specifies a~~ high-level interface ~~to Coordinate Transformation Services, which~~ supports almost the same server functions as the current low-level interface. This high-level interface is estimated to make client software significantly easier to program, compared to the current low-level interface. This high-level interface is independent of the current low-level interface, except that almost the same XML data formats should be used by both interface levels.

This revision of this document is a discussion paper because this draft Implementation Specification could be used in several ways. Those possible uses include use in the OGC Interoperability Program, inclusion in a possible OGC Request For Comment (RFC), or use as a change proposal to the current "low-level" Coordinate Transformation Services (CT) Implementation Specification, now specified in OGC document 01-009.

This document does not now include two-letter prefixes to UML, XML, and CORBA names, partially because it is not clear what prefixes to use. Obviously, two-letter prefixes can be added to selected UML, XML, and CORBA names if considered desirable, after selecting specific prefixes to be used.

ii. Author

Name:	Arliss Whiteside
Address:	BAE SYSTEMS Mission Solutions 16250 Technology Drive San Diego, CA 92127
Phone:	858-592-1608
Email:	Arliss.Whiteside@baesystems.com

Foreword

Many of the geospatial concepts used in this specification are from Topic volume 2 of the Open GIS Consortium (OGC) Abstract Specification (OGC document 01-102).

NOTE This specification assumes that OGC Abstract Specification topic 2 (OGC document 99-102r1) is expanded to include the accepted additions in OGC document 00-026 and accepted editing in OGC document 00-046r1.

Selected portions of this document ~~are also~~could be provided in separate files for ease of use. The possible subjects, sources, and names of these files ~~are~~include:

a) CORBA profile IDL (extracted from Annex D), provided in file named GroundCoordinateTransformation.idl

~~b)DCOM profile IDL (extracted from Annex E), provided in file named TBD~~

~~c)~~b) XML DTDs (extracted from Annex F), provided in files named TBD

~~d)~~c) XML descriptions of parameters for all parameterized transformations (extracted from Annex G), provided in files named TBD

This document contains 9 annexes. Annexes A through F are normative, and annexes G through I are informative.

Attention is drawn to the possibility that some elements of this document may be the subject of patent rights. The Open GIS Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

Introduction

This specification defines standard software interfaces to ground coordinate transformation services. The primary function of these services is to transform the position coordinates of points between different ground (or earth referenced) spatial coordinate reference systems.

These standard software interfaces are intended for implementation by service software that performs coordinate transformations for client software. The OGC expects that multiple vendors will implement these standard interfaces in Commercial Off The Shelf (COTS) software packages. This specification defines a general ground coordinate transformation service interface plus TBD profiles of that interface. These interface profiles are for use by service software that communicates with client software using different distributed computing platforms (or environments). Profiles are specified herein for use in the CORBA, DCOM, and TBD distributed computing platforms.

The position or location of a point can be described using coordinates. Such coordinates are unambiguous only when the coordinate reference system on which those coordinates are based is fully defined. Each position should be described by a set of coordinates based on a specified coordinate reference system. Coordinates are often used in datasets in which all coordinates belong to the same coordinate reference system.

The same point position will usually have different coordinates in different coordinate reference systems. There are a large number of different coordinate reference systems in current use. Coordinates from different datasets will thus often have different coordinate reference systems. In order to use together positions from different coordinate reference systems, known point coordinates often must be transformed into the corresponding coordinates in a different coordinate reference system. Such coordinate transformations are the primary function of these ground coordinate transformation services.

Most of the geospatial concepts used in this interface specification are from the current OGC abstract model of Spatial Reference Systems (Topic 2, document 01-102). The OGC Spatial Reference Systems concepts are largely consistent with ISO/TC 211 DIS 19111 ~~(N 934)~~, although different terms are sometimes used for the same concepts.

OpenGIS© Interface — Coordinate Transformation Services — High Level Interface

1 Scope

This (draft) OpenGIS© Implementation Specification defines standard software interfaces to coordinate transformation services. These services transform point positions from one coordinate reference system to another, usually both ground (or earth-referenced) coordinate systems. Image to image coordinate transformation can also be handled, between different versions of the same original image. Concatenated coordinate transformations are supported, that sequentially apply two or more coordinate transformations. The specified interface include both single point and multiple point coordinate transformations, performed as one operation.

The current scope of this interface does not include the functions:

a) Compute partial derivatives between different coordinates

b) Position error estimate computations

c) Ground to image coordinate transformations

d) Image to ground coordinate transformations

e) Image to image coordinate transformations between different original images, performed as one operation or function

This OpenGIS© standard specifies standard interfaces to be implemented by service software that performs coordinate transformation services for somewhat separate client software. This specification defines a general coordinate transformation service interface plus TBD profiles of that interface. These interface profiles are for use by service software that communicates with client software using different distributed computing platforms (or environments). Profiles are defined herein for use in the CORBA, DCOM, and TBD distributed computing platforms.

2 Conformance

Conformance with this specification shall be checked using all the relevant tests specified in Annex A (normative). The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance, are specified in ISO 19105: Geographic information — Conformance and Testing.

In order to conform to this OpenGIS© interface standard, a software implementation shall choose to implement:

a) Any one of the conformance levels specified in Annex B (normative).

b) Any one of the Distributed Computing Platform (DCP) profiles specified in Annexes D through E (normative).

3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this specification are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

OGC 99-102r2, The OpenGIS™ Abstract Specification, Topic 2: Spatial Reference Systems.

NOTE This specification assumes that OGC Abstract Specification topic 2 (OGC document 99-102r1) is expanded to include the accepted additions in OGC document 00-026) and accepted editing in OGC document 00-046r1.

ISO/TC 211 ~~CD~~ DIS 19111 ~~(N 934)~~, Geographic information – Spatial referencing by coordinates.

European Petroleum Survey Group: EPSG Geodesy Parameters V 5.~~0~~1, available at http://www.epsg.org~~http://www.petroconsultants.com/products/geodetic~~.

ISO/IEC 14750, Information technology – Open Distributed Processing – Interface Definition Language, 1999-03-15

4 Conventions

4.1 Symbols (and abbreviated terms)

API	Application Program Interface
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off The Shelf
CRS	Coordinate Reference System
CT	Coordinate Transformation
DCE	Distributed Computing Environment
DCP	Distributed Computing Platform
DCOM	Distributed Component Object Model

DTD	Document Type Definition
IDL	Interface Definition Language
ISO	International Organization for Standardization
OGC	Open GIS Consortium
TBD	To Be Determined
TBR	To Be Reviewed
UML	Unified Modeling Language
WKT	Well Known Text
XML	eXtensible ~~eXtended~~ Markup Language
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional

4.2 UML Notation

The diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram. The UML notations used in this standard are described in the diagram below.

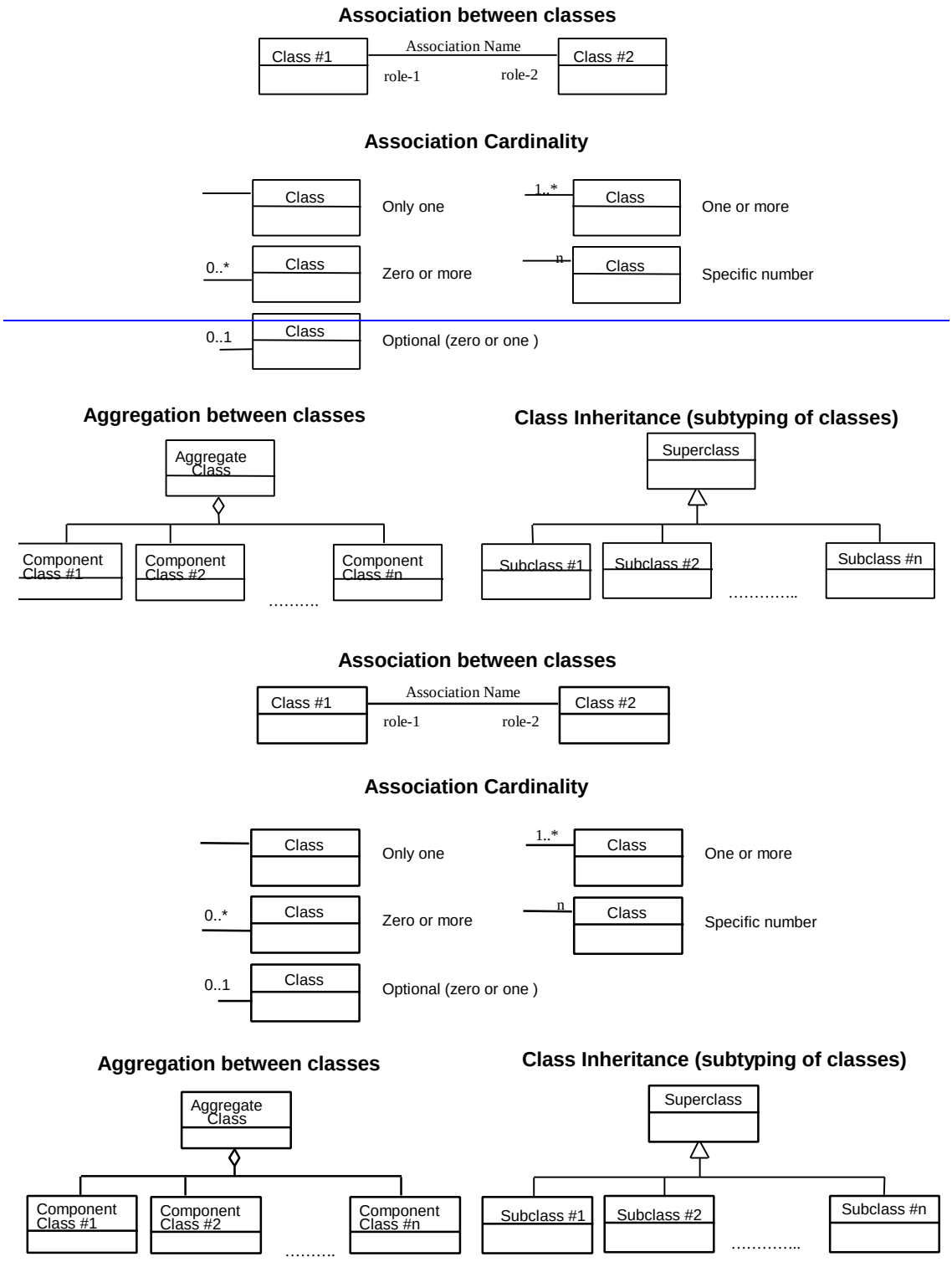

Figure 1 — UML notation

In this standard, the following three stereotypes of UML classes are used:

a) <<Interface>> A definition of a set of operations that is supported by objects having this interface. An Interface class cannot contain any attributes.

b) <<DataType>> A descriptor of a set of values that lack identity (independent existence and the possibility of side effects). A DataType is a class with no operations, whose primary purpose is to hold the information.

c) <<CodeList>> is a flexible enumeration that uses string values for expressing a list of potential values.

In the UML models in this standard, the following standard data types are used:

a) Double – A double precision floating point number

b) CharacterString – A sequence of characters

5 Interface architecture

5.1 Introduction

This clause describes the architecture of the ground coordinate transformation interface. This interface architecture is independent of any specific Distributed Computing Platform (DCP) by which separate software components communicate. This interface architecture is described in three parts:

a) How coordinate reference systems are defined and represented

b) How coordinate transformations are defined and represented

c) Interface for ground coordinate transformations

The details of this general coordinate transformation interface are specified in Annex C (normative).

5.2 Coordinate reference systems

5.2.1 Introduction

The coordinates of points on or near the surface of the earth use coordinate reference systems. These coordinate systems often reference points to the center of the earth, defined as the position of the center of an ellipsoid that approximates the shape of the earth. Some coordinate systems are three-dimensional (3D), and some are only two-dimensional (2D). A 2D ground coordinate system usually defines only the horizontal position along the ellipsoid, not including the vertical position with respect to the ellipsoid or the geoid.

5.2.2 Types of coordinate reference systems

There are a wide variety of types of (ground) coordinate reference systems. The primary types of coordinate systems included in this standard are:

a) Geocentric ~~Cartesian coordinate system~~CRS – An~~y~~ ~~orthogonal~~ 3D Cartesian coordinate reference system with its origin at the (approximate) center of the Earth.

b) Geographic 3D ~~coordinate system~~CRS – Any 3D coordinate reference system based on Latitude and Longitude position around an ellipsoid that approximates the shape of the Earth. The third axis is the height above or below the ellipsoid surface.

c) Geographic 2D ~~coordinate system~~CRS – Any 2D coordinate reference system based on Latitude and Longitude position around an ellipsoid that approximates the shape of the Earth.

d) Projected ~~coordinate system~~CRS – Any 2D map coordinate reference system~~-~~, derived from a Geographic 2D CRS by applying a "map projection" parameterized coordinate transformation~~formed by "projecting" the Latitude and Longitude of a base geographic coordinate system onto a planar surface. The base coordinate reference system must be a geographic coordinate system, and the coordinate transformation is a parameterized transformation that implements the map projection, from geographic coordinates to projected coordinates~~.

e) Vertical ~~coordinate system~~CRS – Any 1D coordinate reference system used for vertical measurements. Such a vertical coordinate reference system can be elevation above the geoid, depth below the geoid, or height above the ellipsoid.

f) Compound ~~coordinate system~~CRS – Any coordinate reference system that combines ~~coordinates from~~ two or more simpler coordinate reference systems. A compound ~~coordinate system~~CRS can have any number of dimensions, from 1 to 4 or more. A compound ~~coordinate system~~CRS is often used to combine a 2D geographic or projected coordinate system with a ~~1D vertical~~ Vertical ~~coordinate system~~CRS, to produce a 3D coordinate system.

g) Local ~~Coordinate System~~CRS: Any coordinate reference system that is defined for and usually used in a limited region, significantly less than the complete surface of the earth. Examples are local engineering or architectural coordinates, grids, and drawings. ~~In this specification, the axes of a spatial Local Coordinate System are always Cartesian.~~ A ~~local~~ Local ~~coordinate system~~CRS can have ~~any number of~~one or more dimensions.~~, from 1 to 4 or more.~~ A Local ~~Coordinate System~~CRS usually has two characteristics:

   1) Does not account for the curvature of the earth's surface. (A Local ~~Coordinate System~~CRS is not a ~~geocentric~~ Geocentric ~~Cartesian coordinate system~~CRS, ~~although it may use Cartesian axes,~~ and is not a map ~~projected~~ Projected ~~coordinate system~~CRS.)

   2) The datum is within or on the edge of the region in which the Local ~~Coordinate System~~CRS is expected to be used. (The origin is neither the center of the earth

ellipsoid nor the point with zero Latitude and Longitude in any common geographic coordinate system.)

The datum of a Local ~~Coordinate System~~CRS might or might not have a specified (or known) position in geodetic coordinates. Similarly, the coordinate axes might or might not have specified directions in geodetic coordinates. Such geodetic coordinates include geographic coordinates, geocentric coordinates, and map projected coordinates.

1) If the origin position and axes directions are specified in any geodetic coordinate system, point positions in a local coordinate system can be transformed to and from other coordinate systems. One common use of such a georeferenced local coordinate system is a grid coordinate system that defines the positions of a rectangular grid of points located in a base coordinate reference system. Such a grid coordinate system uses the grid origin for the axis origin, and uses the grid spacing for the axis units. Such a grid of points is used in a grid coverage, and for other purposes. Such a grid coordinate system can be specified by an affine transformation from the base coordinate reference system. (Note that such a grid coordinate system can be derived from geographic or map projected coordinates, in which case it does account for the curvature of the earth's surface.)

2) In general, if the origin position and/or axes directions are not specified in any geodetic coordinate system, point positions in a local coordinate system cannot be transformed to other coordinate systems. However, if two local coordinate systems have the same number of dimensions, axes names, axes units, and local datum, then software is permitted to assume that the two local coordinate systems are identical. This allows multiple datasets from a common source (e.g. a CAD system) to be overlaid. In addition, some implementations of a Coordinate Transformation software may have a mechanism for correlating Local Datums. (e.g. from a database of transformations, which is created and maintained from measurements.)

5.2.3 Relationships between coordinate reference systems

With the exception of some local coordinate reference systems, there are known relationships between the different types of coordinate reference systems. Therefore, the data that ~~specifies~~ describes a coordinate reference system ~~usually~~ often includes data that specifies the coordinate transformation ~~of~~ to that ~~coordinate system~~CRS ~~to~~ from a different ~~coordinate reference system~~CRS, often to a more-basic ~~coordinate system~~CRS. For example, a geographic coordinate reference system ~~specification~~ description defines its relationship to a ~~geocentric~~ Geocentric ~~Cartesian coordinate system~~CRS. Similarly, a ~~local~~ Local ~~coordinate system~~CRS ~~specification~~ description often defines its relationship to another ~~coordinate reference system~~CRS. This coordinate transformation information is included so that data for a source and a target coordinate system often provides enough information to define at least one coordinate transformation from that source coordinate reference system to that target coordinate reference system.

5.3 Coordinate transformations

5.3.1 Coordinate transformation definitions

In general, the complete definition of a coordinate transformation requires definitions of three parts:

a) Source coordinate reference system, of input point coordinates.

b) Target coordinate reference system, of output point coordinates.

c) Specific coordinate transformation or transformation choice.

A specific coordinate transformation defines the transformation method and specifies all the needed transformation parameters. A transformation choice selects one from multiple alternative coordinate transformations from the source to the target coordinate reference systems. The transformation choice is needed whenever more than one transformation alternative between the two related coordinate systems could be used by the server software. For example, one known transformation may be fast and approximate (quick and dirty) and other transformations may be slower and more accurate.

This standard makes no distinctions between coordinate conversions (specified without any errors) and coordinate transformations (usually specified with some errors). In both cases, a coordinate conversion or transformation service will commit some computation errors. Furthermore, the software interface should be identical for both coordinate conversions and transformations. In this standard, the term coordinate transformation is thus used for both conversions and transformations, using the same interface.

5.3.2 Categories of coordinate transformations

There are three broad categories of coordinate transformations: implied, well-known, and custom.

Some coordinate transformations are implied (or indirectly specified) by the source and target coordinate reference systems. This occurs when the complete specifications of the source and target coordinate reference systems include sufficient data (as discussed in Clause ~~6~~5.2.3) to define a useful coordinate transformation from the source to the target coordinate reference system.

Some coordinate transformations are well-known (including the source and target coordinate reference systems). Well-known coordinate transformations are often, but not always, used for ground coordinate systems. Detailed definitions of such well-known coordinate transformations can be built into server software, or can be automatically accessed by server software from an “authority” source of data. Many such “authorities” provide detailed definitions of multiple coordinate reference systems and multiple transformations, with each identified using some identifier.

However, some coordinate transformations are custom (or special), with the detailed definitions known only to, or primarily by, the client. Such custom coordinate transformations are common for image coordinates, and are sometimes used for ground coordinates. To handle custom coordinate transformations, server software must allow

some clients to provide the detailed definition (or specifications) of a custom coordinate transformation.

5.3.3 Selection from multiple coordinate transformations

An implied or a well-known coordinate transformation will often be just one of multiple possible coordinate transformations from one source to one target coordinate reference system. That is, several different coordinate transformations may exist and have different domains of validity, accuracies, and computation speeds. To handle this situation, server software shall allow clients to provide the detailed definition of a specific desired coordinate transformation.

In this architecture, the ability for a client to provide information specifying the desired coordinate transformation is provided by including coordinate transformation definition data in transformation metadata sent to server software. When the client exercises this capability, this coordinate transformation definition data will be populated, and the server must use the client provided data. When the client does not exercise this capability, this transformation data will be empty. This can occur when there is only one implied or well-known transformation from the source to the target coordinate system. This can also occur when there are multiple alternative transformations, but the client chooses to let the server software make the choice.

Server software that implements this interface specification is allowed, but not required, to automatically select from among multiple available coordinate transformations, when a client does not define a choice. For example, server software could select default transformations coded for selected pairs of source and target coordinate reference systems. Also, server software could make more "intelligent" choices from among multiple available transformations. Whenever server software is unable to select from multiple transformations, it shall reject the client request.

For “intelligent” choices between multiple coordinate transformations, a wide variety of decision algorithms and information sources could be used by server software. This version of this interface specification does not specify these decision algorithms and information sources. Those decisions are left up to the vendors that implement these specifications, allowing competition on this subject. In the future, server software could allow clients to provide information to help the server select from multiple coordinate transformations available. For example, a client may need a coordinate transformation with high computation speed and high accuracy but only a limited domain of validity.

5.3.4 Multiple coordinate transformations

Server software can provide the ability to perform a set of different coordinate transformations. However, each additional coordinate transformation (including its’ source and target coordinate reference systems) that is simultaneously active in a server has a computer memory cost. This cost will usually be borne by the user of the server, perhaps by:

a) Additional memory is consumed by server software that runs on a user-supplied computer.

b) User is charged for each additional active coordinate transformation, either actually used or maximum allowed active.

c) Additional time is consumed by server software in responding to client called operations.

Therefore, means are provided to limit the number of coordinate transformations simultaneously active in a server. These means allow each client to limit the number of coordinate transformations which are simultaneously active in the server for that client.

5.4 Ground coordinate transformation service

5.4.1 Introduction

The ground coordinate transformation service performs coordinate transformations between many pairs of different coordinate reference systems. This ground coordinate transformation service supports all the types of ground coordinates listed in Subclause 6.2.

The standard interface to a ground coordinate transformation service has only one <<Interface>> object visible to a client, and this object always exists while the server is active. This object is of the "GroundCoordinateTransformationService" class shown in the UML diagram in Figure 2. This interface object has five specified public operations as listed in the box in the diagram and as described in the following subclauses. (As an interface object, this object has no public attributes.)

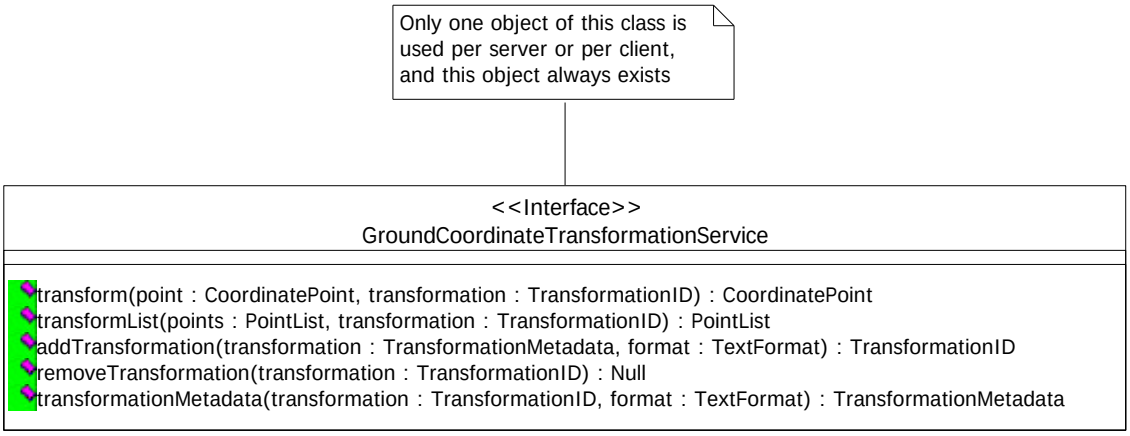

Figure 2 – Ground coordinate transformation service class

The operations provided by this interface object use one <<CodeList>> class and seven <<DataType>> classes, named:

a) TextFormat <<CodeList>>

b) CoordinatePoint <<DataType>>

c) PointList <<DataType>>

d) TransformationID <<DataType>>

e) TransformationMetadata <<DataType>>

f) CoordinateReferenceSystem <<DataType>>

g) TransformationDefinition <<DataType>>

These eight data classes are described in subclauses following the descriptions of the five public operations.

All seven of these <<DataType>> classes in this DCP-independent UML model are used primarily for specification precision. That is, each data type has a definition that applies to all the operation inputs and outputs which use that data type. However, each such data type has a single attribute, or has simple contents. These data types are defined because using "CharacterString", Sequence<Double>, or other unnamed data types for different operation arguments does not indicate which of these arguments must have exactly the same format and contents. In DCP-specific profiles, these data types can be replaced by the underlying contents of that class, if useful for efficiency, without changing the specified meaning of each data type.

5.4.2 The "transform" operation

transform (point : CoordinatePoint, transformation : TransformationID) : CoordinatePoint

The most fundamental operation provided by the Ground Coordinate Transformation Service object is the "transform" operation. This "transform" operation uses a Coordinate Point input, in a source coordinate reference system, and outputs the corresponding Coordinate Point, in a target coordinate reference system. The other input to this operation is a Transformation ID, which identifies the source and target coordinate reference systems.

This "transform" operation is optional implementation by server software, since the same function can be performed using the "transformList" operation discussed next.

5.4.3 The "transformList" operation

transformList (points : PointList, transformation : TransformationID) : PointList

The "transformList" operation uses a Point List input, in one source coordinate reference system, and outputs the corresponding Point List, in one target coordinate reference system. The other input to this operation is a Transformation ID, which identifies the source and target coordinate reference systems.

Each Point List contains a list of Coordinate Point data, or the equivalent. This "transformList" operation is provided for efficiently transforming the coordinates of a list of points, by the client calling just one operation.

5.4.4 The "addTransformation" operation

addTransformation (metadata : TransformationMetadata, format : TextFormat) :
TransformationID

The “addTransformation” operation allows a client to obtain the Transformation ID value for the transformation between client-specified source and target coordinate reference systems. This operation uses a Transformation Metadata input, and outputs the corresponding Transformation ID. The Transformation Metadata input includes data defining the source and target coordinate reference systems. Whenever needed, the Transformation Metadata input also includes data defining the desired coordinate transformation. This operation also uses a Text Format input that specifies the format used by the Transformation Metadata input.

The Transformation Metadata input will often contain only the identifications of the source and target coordinate reference systems, when those systems are well-known. This requires that the server software already know or be able to find the definitions of a set of well-known coordinate transformations (including the corresponding source and target coordinate reference systems). The specific set of coordinate transformations that are well-known is implementation specific, but a standard set of well-known coordinate transformations for compliant implementations are specified in Subclause B.4.

The server software is expected to make the specified transformation active when the “addTransformation” operation is executed for a client. If the server software must find and/or convert for quick use the specifications of well-known coordinate transformations (including the corresponding source and target coordinate reference systems), the server is expected to cache the (converted) specifications of multiple recently-used transformations.

Each client is required to use this addTransformation operation to obtain each Transformation ID, before using that Transformation ID in a “transform”, “transformList”, or “transformationMetadata” operation. For server interoperability, a server implementation is not allowed to specify a set of Transformation ID values that are well-known to clients of that service implementation.

5.4.5 The “removeTransformation” operation

removeTransformation (transformation : TransformationID) : Null

The “removeTransformation” operation allows a client to remove or make inactive a specific Transformation ID value, including the definition of a custom coordinate transformation. This operation is provided for client use when that coordinate transformation is no longer needed by that client. The coordinate transformation (including source and target coordinate reference systems) to be removed is identified by the Transformation ID input. This operation has no output.

This removeTransformation operation is optional implementation by server software, since some clients are not expected to need this capability. Whether this operation is provided or not, a server implementation:

a) Is allowed to limit the number of currently active coordinate transformations by automatically making non-active server-selected active transformations that were not recently used.

b) Is not allowed to limit the number of currently active coordinate transformations by rejecting additional coordinate transformations received from client(s), after some maximum number of active transformations is reached.

5.4.6 The "transformationMetadata" operation

transformationMetadata (transformation : TransformationID, format : TextFormat) :
TransformationMetadata

The "transformationMetadata" operation allows a client to retrieve a detailed definition of the coordinate transformation (including source and target coordinate reference systems) that is identified by an input Transformation ID. This operation also uses a Text Format input that specifies the format to be used by the Transformation Metadata output.

This "transformationMetadata" operation is optional implementation by server software, since some clients are not expected to need this capability.

5.4.7 The "Coordinate Point" data type

The Ground Coordinate Transformation Service uses the "Coordinate Point" data type to transfer the position coordinates of one point. This data type includes an ordered sequence of double precision floating point (or Double) values for the ordinates of the point.

For efficiency in a DCP-dependent profile, the Coordinate Point data type could be replaced by an ordered sequence of Double data types.

5.4.8 The "Point List" data type

The "Point List" data type is used to transfer a list of coordinate points. This data type includes an ordered sequence of Coordinate Point data values.

For efficiency in a DCP-dependent profile, the Point List data type could be replaced by a two-dimensional array of Double data types, or the equivalent.

5.4.9 The "Transformation ID" data type

The "Transformation ID" data type is used to transfer the identifier of a specific coordinate transformation. The Transformation ID data type contains one character string, whose contents are not interpretable by (are opaque to) clients, and are not shared by different server software.

For efficiency in a DCP-dependent profile, the Transformation ID data type could be replaced by the character string data type.

5.4.10 The "Text Format" code list

The "Text Format" code list is used to transfer the identity of the format of the related Transformation Metadata. The allowed codes correspond to the names of one or more XML-based formats, perhaps plus one or more WKT-based formats.

At least one of the XML based formats is required implementation by server software, with the other possible formats being optional implementation.

5.4.11 The "Transformation Metadata" data type

The "Transformation Metadata" data type is used to transfer transformation definition data, including the source coordinate reference system, target coordinate reference system, and transformation choice. That is, this data type contains one source Coordinate Reference System definition, one target Coordinate Reference System definition, and one optional Transformation Definition. For input to server software, the Transformation Definition can be omitted or empty unless a transformation selection or specification is needed between the related source and target coordinate reference systems.

For efficiency in a DCP-dependent profile, the Transformation Metadata data type could be replaced by the character string data type. For example, that character string could use XML to combine the source and target Coordinate Reference System values with the optional Transformation Definition value.

5.4.12 The "Coordinate Reference System" data type

The "Coordinate Reference System" data type is used to transfer definition data for a source or target coordinate reference system. The Coordinate Reference System data type contains one character string, whose contents encode the needed coordinate reference system definition data using the format specified by the related Text Format value. XML encoding of a Coordinate Reference System definition is introduced later in Subclause 8.2.

For efficiency in a DCP-dependent profile, the Coordinate Reference System data type could be replaced by the character string data type.

5.4.13 The "Transformation Definition" data type

The "Transformation Definition" data type is used to transfer definition data for a coordinate transformation (not including the related source and target coordinate reference systems). For input to server software, the Transformation Definition can be empty unless a transformation specification or selection is needed between the related source and target coordinate reference systems. The Transformation Definition data type contains one character string, whose contents encode the needed coordinate transformation definition data using the format specified by the related Text Format value. XML encoding of a Transformation Definition is introduced later in Subclause 8.2.

For efficiency in a DCP-dependent profile, the Transformation Definition data type could be replaced by the character string data type.

6 DCP specific interface profiles

6.1 Introduction

This clause introduces the TBD specified profiles of the general Ground Coordinate Transformation interface described in Clause 6. These profiles are for specific Distributed

Computing Platforms (DCPs) by which separate software components communicate. Profiles are included for the CORBA, DCOM, and TBD.

6.2 CORBA interface profile

This subclause introduces the CORBA specific profile of the general Coordinate Transformation interface introduced in Clause 5. The details of this CORBA profile are specified in Annex D (normative). The CORBA profile is a largely straightforward translation from the general interface UML model into the ISO/CORBA Interface Design Language. Some specifics are:

a) The ground coordinate transformation service is translated into one IDL module.

b) The Text Format code list is implemented as a CORBA string, for ease of expansion. The CORBA profile specifies standard values for this string for each currently specified text format.

c) The Transformation Metadata is transferred as a CORBA data structure containing three strings. (The CORBA profile does not use XML encoding of the Transformation Metadata as one XML document.)

d) Three CORBA exception types are specified, with multiple subtypes. All operations of all interfaces are allowed to raise any of these exceptions. All exception types are allowed since different coordinate systems and coordinate transformations, including probable future additions, could cause different exceptions to be raised.

e) Each use-specific exception returns a data structure containing four strings. The first string describes the exception subtype (and implies the exception type). The possible contents of this first string are specified in this CORBA profile, and shall be used by all compliant implementations of this specification. The other three strings provide more detailed information in human understandable text. The contents of these three strings are not standardized in this CORBA profile, allowing implementations to use different string contents. That is, the contents of these strings can be tailored to specifics of the server software, and the expected usage of that software. The last two strings can be empty.

6.3 DCOM interface profile

This subclause introduces the DCOM specific profile of the general Image Coordinate Transformation interface introduced in Clause 5. The details of this DCOM profile are specified in Annex E (normative). The DCOM profile is TBD

7 Supporting information

7.1 Introduction

This clause introduces information supporting the software interface specifications that are summarized in Clauses 5 and 6, and are more fully specified in Annexes A through E.

7.2 XML encoding of coordinate reference systems and transformations

The coordinate transformation interfaces specified in this document make heavy use of XML encoding for transfer of data between a client and a server. Indeed, much of the interface complexity is encapsulated in the XML data structure alternatives and options. The same XML encodings are used by the general ground coordinate transformation interface introduced in Clause 5 and by all the DCP specific profiles introduced in Clause 6. These XML encodings are largely the same as proposed in OGC document ~~00-040r8~~01-014.

The details of this XML encoding are specified in Annex F (normative). Annex G (informative) then provides overviews and examples of the specified XML encoding. Annex G also summaries the differences in this XML encoding, compared to the corresponding XML DTDs proposed in OGC document ~~00-040r8~~01-014.

Annexes F and G each describe the XML used in three parts, for representation of:

a) Coordinate Reference System definition

b) Coordinate Transformation Definition

c) Shared Elements, used by both above

7.3 Interface design decisions

Annex H (informative) provides insight into the design process used to develop this interface specification. This annex first summarizes the design objectives and design approaches used to guide design decisions. This annex then summarizes ~~some~~ many of the decisions made in designing these coordinate transformation interfaces. These design decisions are listed in groups depending on the interface part affected:

a) Ground coordinate transformation service

b) XML for coordinate reference system definition

c) XML for coordinate transformation definition

7.4 Possible future expansion

There are a large number of possible refinements and expansions that could be made to the coordinate transformation interfaces specified in this document. Annex I (informative) briefly lists a few of the capabilities that might be added in the future.

8 Terms and definitions

For the purposes of this specification, the following terms and definitions apply.

NOTE These definitions are copied from the proposed OGC Recommendation Paper on Recommended Definition Data for Coordinate Reference Systems and Coordinate Transformations, OGC document ~~00-040r8~~01-014. However, some definitions in that document are not relevant here, and are thus omitted here.

8.1
affine coordinate system
coordinate system with straight axes that are not necessarily mutually-perpendicular

8.2
Cartesian coordinate system
coordinate system which gives the position of points relative to *N* mutually-perpendicular straight axes

NOTE 1 These coordinates are often called rectangular coordinates.

NOTE 2 *N* is 2 or 3.

8.3
compound coordinate reference system
coordinate system describing the position of points through two or more independent coordinate reference systems

EXAMPLE One coordinate reference system can be a two-dimensional horizontal coordinate system, and the other coordinate reference system can be a vertical gravity-related height system.

8.4
concatenated transformation
sequential application of multiple coordinate transformations.

8.5
coordinate
one of a sequence of *N* numbers designating the position of a point in *N*-dimensional space

NOTE In a coordinate reference system, the numbers must be qualified by units.

8.6
coordinate conversion
change of coordinates, based on a one-to-one relationship, from one coordinate system to another based on the same datum

EXAMPLE Between geographic and Cartesian coordinate systems or between geographic coordinates and map projected coordinates, or change of units such as from radians to degrees or feet to meters.

NOTE A conversion uses parameters which have ~~constant~~ specified values, not empirically determined values.

8.7
coordinate reference system
coordinate system which is related to the real world by a datum

NOTE For geodetic and vertical datums, it will be related to the Earth.

8.8
coordinate system ~~description~~
~~description of~~ the set of coordinate axes used to record point coordinates in a coordinate reference system

NOTE One coordinate system may be used in many coordinate reference systems.

8.9

coordinate transformation

computational process of converting a position given in one coordinate reference system into the corresponding position in another coordinate reference system

NOTE 1 A coordinate transformation can require and use datum and ellipsoid parameters.

NOTE 2 The term 'transformation' is strictly used only when the parameter values associated with the transformation have been determined empirically from a measurement / calculation process. This is typically the case when a change of datum is involved.

~~NOTE 2 This term is strictly used only when the coordinate transformation is known only approximately. For example, this term is used when the transformation coefficients are determined by least squares adjustment. This term is loosely used when the coordinate transformation is known either approximately or exactly.~~

8.10

datum

parameter or set of parameters that may serve as a reference or basis for the calculation of other parameters

NOTE 1 A datum defines the position of the origin, the scale, and the orientation of the axes of a coordinate system.

NOTE 2 A datum may be a geodetic datum, a vertical datum or an engineering datum.

8.11

depth

distance of a point below a chosen reference surface, usually measured along the local vertical (gravity vector).

NOTE 1 Depth is sometimes measured along a line that does not follow the local gravity vector. An example is depth in an oil or gas well. These are generally measured along the wellbore path, which may vary significantly from the local vertical. Some sections of a wellbore path may even run horizontally or slope upwards. Nevertheless the distance along the wellbore path is referred to as 'depth'.

~~distance of a point below a chosen reference surface along a line not in that surface~~

~~NOTE 1 Depth is usually, but not always, measured along a line that is perpendicular to the reference surface. Well depths are generally measured along the wellbore path, which may vary significantly from the perpendicular.~~

NOTE 2 See elevation, ellipsoidal height, and gravity-related height.

8.12

dimension

number of ~~measurements or~~ ordinates ~~axes~~ needed to describe a position in a coordinate system

8.13

Distributed Computing Platform (DCP)

a standardized software environment enabling distributed computing, by supporting cooperation among software executing on multiple computers that communicate over a communications network

NOTE 1 A DCP enables software applications to interact with each other even though they execute on different computers. A DCP addresses issues of networking, communicating between different computer systems from different manufacturers, distributed data storage, security, and other client/server issues.

NOTE 2 This OGC term is an abstraction of the various distributed computing environments available, especially those in the marketplace today. A DCP is thus a generalization of the CORBA, DCE, JAVA, SQL, DCOM, and other distributed computing platforms. (The OGC would have used "Distributed Computing Environment" (DCE) but that term was already used by one particular DCP.)

EXAMPLES Microsoft's DCP is called OLE/COM or DCOM. The Object Management Group's DCP is called CORBA. Microsoft's DCPs include DCOM, OLE/COM, and SOAP. The SQL standard that includes a protocol for network-based access to SQL databases is also considered a DCP. Other examples include Distributed Computing Environment (DCE) from the Open Software Foundation (OSF) and Java from SunSoft.

8.14
elevation
distance of a point from a chosen reference surface along a line perpendicular to that surface

NOTE 1 See ellipsoidal height and gravity-related height.

NOTE 2 Height of a point outside the surface treated as positive; negative height is also named as depth.

8.15
ellipsoid
surface formed by the rotation of an ellipse about an axis

NOTE In this ~~Standard~~document, ~~ellipsoids are always oblate, meaning that~~ the axis of rotation is always the minor axis.

8.16
ellipsoidal height
geodetic height
distance of a point from the ellipsoid measured along the perpendicular from the ellipsoid to this point, positive if upwards or outside of the ellipsoid

NOTE Only used as part of a three-dimensional geodetic coordinate system and never on its own.

8.17
flattening
ratio of the difference between the semi-major (a) and semi-minor axis (b) of an ellipsoid to the semi-major axis; $f = (a-b)/a$

NOTE Sometimes inverse flattening $1/f = a/(a-b)$ is given instead of flattening; $1/f$ is also known as reciprocal flattening.

8.18
geocentric ~~Cartesian~~ coordinate reference system
3-dimensional Cartesian coordinate reference system with its origin at the (approximate) center of the Earth

8.19
geodetic coordinate system
ellipsoidal coordinate system
coordinate system in which position is specified by geodetic latitude, geodetic longitude and (in the three-dimensional case) ellipsoidal height

8.20
geodetic coordinates
coordinates defined in a Geocentric, Geographic (2D or 3D) or Projected coordinate reference system.

~~8.20~~8.21
geodetic datum
datum describing the relationship of a coordinate system to the Earth

NOTE In most cases, the geodetic datum includes an ellipsoid definition.

~~8.21~~8.22
geographic coordinate reference system
coordinate reference system using an ellipsoidal coordinate system and based on an ellipsoid that approximates the shape of the Earth

~~coordinate reference system based on Latitude and Longitude position around an ellipsoid that approximates the shape of the Earth~~

NOTE A geographic coordinate reference system can be 2D or 3D. In a 3D geographic coordinate reference system, the third dimension is height above or below the ellipsoid surface.

~~8.22~~8.23
geoid
level surface which best fits mean sea level either locally or globally

NOTE "Level surface" means an equipotential surface of the Earth's gravity field which is everywhere perpendicular to the direction of gravity.

~~8.23~~8.24
gravity-related height
height dependent on the Earth's gravity field

NOTE In particular, orthometric height or normal height, which are both approximations of the distance of a point above the ~~sea level~~geoid.

~~8.24~~8.25
Greenwich meridian
meridian passing through Greenwich, United Kingdom

NOTE Most geodetic datums use Greenwich meridian as the prime meridian.

~~8.25~~8.26
ground coordinates
earth referenced coordinates
terrestrial coordinates
coordinates of points measured in a non-image coordinate reference system

NOTE The term ground coordinates is used herein to distinguish such coordinates from image coordinates. Even when an image is collected by a near vertical camera, image coordinates are different from ground coordinates!

~~8.26~~8.27
height
altitude
distance of a point from a chosen reference surface along a line perpendicular to that surface

NOTE 1 See ellipsoidal height and gravity-related height.

NOTE 2 Height of a point outside the surface treated as positive; negative height is also named as depth.

~~8.27~~8.28
image
record of the likeness of any features, objects, and activities

NOTE An image can be acquired through the sensing of visual or any other segment of the electromagnetic spectrum by sensors, such as thermal infrared, and high resolution radar.

~~8.28~~8.29
image coordinates
definition of position within an image, expressed in image row and column coordinates

~~8.29~~8.30
interface
shared boundary between two functional entities

NOTE An interface standard specifies the services in terms of the functional characteristics and behavior observed at the interface. The standard is a contract in the sense that it documents a mutual obligation between the service user and provider and assures stable definition of that obligation.

~~8.30~~8.31
latitude
geodetic latitude
ellipsoidal latitude
angle from the equatorial plane to the perpendicular to the ellipsoid through a given point, northwards treated as positive

~~8.31~~8.32
local coordinate system
a coordinate reference system that is defined for and usually used in a limited region, significantly less than the complete surface of the earth

EXAMPLES Local engineering and architectural coordinates, grids, and drawings.

NOTE The datum of a Local Coordinate Reference System might or might not have a specified (or known) position in geodetic coordinates. Similarly, the coordinate axes might or might not have specified directions in geodetic coordinates. Such geodetic coordinates include geographic coordinates, geocentric coordinates, and map projected coordinates.

~~8.32~~8.33
local datum
engineering datum
datum with a local reference, used as a basis for a local coordinate system

NOTE Engineering datum excludes both geodetic and vertical datums.

~~8.33~~8.34
longitude
geodetic longitude
ellipsoidal longitude
angle from the prime meridian plane to the meridian plane of the given point, eastward treated as positive

~~8.34~~8.35
map projection
conversion from a geodetic coordinate system to a plane

~~8.35~~8.36
mean sea level
average level of the surface of the sea average over all stages of tide

NOTE Mean sea level in a local context normally means mean sea level for the region as measured by tide gauge measurements at one or more points over a given period of time. Mean sea level in a global context differs from a global geoid by not more than 2 metres.

~~8.36~~8.37
meridian
intersection of an ellipsoid by a plane containing the semi-minor axis of the ellipsoid

NOTE This term is often used for the pole-to-pole arc rather than the complete closed figure.

~~8.37~~8.38
position
spatial reference of a point or an object

~~8.38~~8.39
prime meridian
zero meridian
meridian from which the longitudes of other meridians are quantified

~~8.39~~8.40
projected coordinate reference system
two-dimensional ~~Cartesian~~ coordinate reference system resulting from a map projection, commonly using a Cartesian set of coordinate axes (coordinate system)

NOTE A projected coordinate system is derived from a 2D geographic coordinate system by applying a parameterized coordinate transformation known as a "map projection" ~~parameterized coordinate transformation~~.

~~8.40~~8.41
semi-major axis
equatorial radius of an ellipsoid

~~8.41~~8.42
semi-minor axis
distance from centre of an ellipsoid to either of its poles

~~8.42~~8.43
transformation
change of coordinates from one coordinate reference system to another coordinate reference system based on a different datum through a one-to-one relationship

NOTE A transformation uses parameter values which may have to be derived empirically by a set of points common to both coordinate reference systems. See coordinate conversion and coordinate transformation.

~~8.43~~8.44
unit
defined quantity in which dimensioned parameters are expressed

NOTE In this document, the subtypes of units are length units and angular units.

~~8.44~~8.45
vertical coordinate reference system
1-dimensional coordinate reference system used for elevation, height, or depth measurements

~~8.45~~8.46
vertical datum
datum describing the relation of gravity-related heights to the Earth

NOTE In most cases the vertical datum will be related to sea level. Ellipsoidal heights are treated as related to a three-dimensional ellipsoidal coordinate system referenced to a geodetic datum. Vertical datums include sounding datums (used for hydrographic purposes), in which case the heights may be negative heights or depths.

~~8.46~~8.47
well-known
shared
data that has been completely specified and published by some recognized authority, and is accessible through use of an identifier

NOTE 1 Well-known data does not always have to be transferred between client and server software.

NOTE 2 Well-known usually implies that this data is very trustworthy. Well-known does NOT imply that this data is less trustworthy or of lesser importance.

Annex A
(normative)

Compliance testing

1A.1 Compliance testing overview

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

Compliance testing of 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.

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 coordinate reference system, supported target coordinate reference system, and supported type of coordinate transformation without specific source and target coordinate reference systems.

For each tested coordinate transformation, a sequence of service operations shall be called, all using that coordinate transformation. These sequences of service operations are specified below in Subclause A.3.

For most coordinate transformations, the sequence of service operations called to test that coordinate transformation shall be intermixed with the service operations called to test at least one other coordinate transformation. However, a few coordinate transformations shall be tested without intermixing operations with other coordinate transformations.

1A.2 Documentation review

Compliance testing shall begin with a documentation review, to determine the implementation capabilities that need to be tested. This documentation review shall also check if the set of all capabilities implemented is sufficiently specified. Specifically, this review shall check if the documentation clearly specifies the capabilities implemented, including:

a) Which of the specified compliance levels it fully supports.

b) Which DCP-specific profile(s) it implements.

c) Which coordinate transformations it implements.

d) Which coordinate reference systems it implements.

e) Which interface operations it implements.

f) Which data types it implements.

g) Which Text Format code values it implements.

All of this information shall be used to define or select the compliance tests to be performed. For example, the DCP-specific profile(s) shall be used to select the form to be used in calling interface operations during the testing. The list of implemented coordinate transformations defines the set of transformations which 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, data type, and Text Format value shall be tested.

~~1~~A.3 Operations calling sequence

For each tested coordinate transformation, a sequence of service operations shall be called, all using that coordinate transformation. The first service operation called shall be the addTransformation operation~~, of the Ground Coordinate Transformation Service~~. The Transformation ID value returned by the addTransformation operation shall then be used as an input to all other operations. If the removeTransformation operation of the Ground coordinate transformation service is implemented, it shall be the last service operation called.

Between the addTransformation operation and the removeTransformation operation, each other implemented transformation service operation shall be called at least once. These other interface operations are:

a) transformList ~~(Ground coordinate transformation service)~~

b) transform ~~(Ground coordinate transformation service)~~

c) transformationMetadata ~~(Ground coordinate transformation service)~~

NOTE Depending on the compliance level, not all the operations listed above need to be implemented. The operations not implemented can be omitted in the compliance testing.

The three interface operations listed above can be legally called in any order, so a variety of operation orders shall be used for testing different coordinate transformations. Also, each of the interface operations listed above can be called multiple times, so each implemented operation shall be called more than once in testing several different coordinate transformations.

Each of the interface operations listed above has one or more outputs that should have the proper correspondence to the operation inputs, depending on the specific coordinate transformation. For conformance testing, the output from each call to each of these

operations shall be checked to ensure that it has the correct correspondence to the inputs provided.

NOTE The bulk of the work involved in preparing compliance tests will be in obtaining the "correct" outputs that correspond to the various inputs, for each coordinate transformation. For each "correct" output 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 Coordinate Points for each coordinate transformation tested.

Annex B
(normative)

Compliance levels and options

~~1~~B.1 Compliance overview

This Implementation Specification defines TBD different DCP-specific profiles, for CORBA, DCOM, and TBD. A compliant implementation of this specification shall implement at least one of these DCP-specific profiles, and shall clearly specify which profile(s) it implements.

This specification specifies a basic set of coordinate transformations, and a basic set of coordinate reference systems. However, a compliant implementation can implement additional coordinate transformations and coordinate reference systems. Furthermore, a compliant implementation might not implement all the coordinate transformations and coordinate reference systems specified herein, as discussed below under compliance levels.

Any compliant implementation of this Implementation Specification shall clearly specify which sets of coordinate transformations and of coordinate reference systems it implements. Whenever applicable, the definition of each implemented coordinate transformation shall identify the source and target coordinate reference systems.

This specification currently defines two compliance levels with the service interfaces specified herein. Since many clients will not require the full capabilities provided by these image coordinate transformation interfaces, some of the specified interface elements are considered optional implementation by server software. These optional elements are specified in Subclause B.2.

Any compliant implementation of this Implementation Specification shall clearly specify which interface elements it implements, including both required and optional interface operations and data types. The implemented interface elements are allowed to include operations and data types not specified in this document. Any compliant implementation shall also clearly specify which code values of the Text Format code list are implemented, and any limitations on when each code value can be used.

To support interoperability among compliant implementations, Subclause B.3 specifies two levels of compliance with this Implementation Specification. Each compliance level specifies which of the optional interface elements, if any, must be implemented.

~~1~~B.2 Optional interface elements

This Implementation Specification defines many interface elements that are optional implementation by server software, since many clients will not require the full interface capabilities. The optional interface elements are:

a) Ground coordinate transformation service:

1) Optional operations:

i) transform

ii) transformationMetadata

iii) removeTransformation

2) Optional data types: none optional

3) Optional coordinate reference system types: none optional

4) Optional coordinate transformation types:

i) Inverse transformation (TBR)

ii) Pass through transformation

b) XML for coordinate transformation definition:

1) Optional XML Elements:

i) InverseTransformation

NOTE The InverseTransformation XML element is required if the Inverse Transformation coordinate transformation type is implemented.

ii) PassThroughTransformation

iii) firstModiedAxis

iv) numberModifiedAxes

NOTE The three XML elements listed above are required if the Pass Through Transformation coordinate transformation type is implemented.

2) Optional XML Attributes: none optional

c) XML for coordinate reference system definition:

1) Optional XML Elements: none optional

2) Optional XML Attributes: none optional

d) XML for shared elements:

1) Optional XML Elements:

i) ValidityRegion

ii) validityArea

iii) minimumCoordinates

iv) maximumCoordinates

2) Optional XML Attributes: none optional

~~1~~B.3 Compliance levels

To promote interoperability among compliant implementations, this ~~annex~~ subclause specifies two levels of compliance with this Implementation Specification. The lower level of server implementation compliance does not require implementation of any of the optional interface elements. The higher level of server implementation compliance requires implementation of all the optional interface elements.

The levels of compliance are specified in the following subclauses, and are summarized in Table B.1. In this table, the defined levels of compliance are enumerated across the top, and various groups of interface elements are enumerated down the left side. Each group of interface elements not starting with the word "Options" excludes the set of options referred to in the following row that starts with the word "Options". A table cell containing the tern "shall" means that this level of compliance shall implement this group of interface elements. A table cell containing the term "need not" means that this level of compliance need not implement this group of interface elements.

Table B.1 – Compliance levels specification

Group of Interface Elements	Compliance Level Name	
	Minimum ground coordinate transformation server	Maximum ground coordinate transformation server
Ground coordinate transformation service	shall	shall
Options listed in item a) of clause B.2~~1~~	need not	shall
XML for coordinate transformation definition	shall	shall
Options listed in item b) of clause B.2~~1:~~	need not	shall
XML for coordinate reference system definition	shall	shall
Options listed in item c) of clause B.2~~1~~	need not	shall
XML for shared elements	shall	shall
Options listed in item d) of clause B.2~~1~~	need not	shall

~~1~~B.3.1 Minimum ground coordinate transformation server

The "Minimum ground coordinate transformation server" compliance level shall implement the ground coordinate transformation service interface, but need not implement any of the optional interface elements listed for that service under item a) in Subclause B.2.

~~1~~B.3.2 Maximum ground coordinate transformation server

The "Maximum ground coordinate transformation server" compliance level shall fully implement the ground coordinate transformation service interface, including implementing all the optional interface elements listed for this service under item a) in Subclause B.2.

~~1~~B.4 Coordinate transformations

This subclause specifies a set of parameterized coordinate transformations for possible implementation by the Ground Coordinate Transformation Service. The possible parameterized coordinate transformation types shall include:

a) Affine

b) Ellipsoid To Geocentric

c) Geocentric To Ellipsoid

d) Abridged Molodenski

e) Bursa Wolf

f) Longitude Rotation

~~NOTE 1 Descriptions of the above listed transformation types are included in Subclause F.2.~~

NOTE ~~2~~ The following Parameterized Transformations are map projections. These map projections and their parameters are all specified in the EPSG database.

g) Transverse Mercator

h) Transverse Mercator South Orientated

i) Molodenski

j) Lambert Conformal Conic 1SP

k) Lambert Conformal Conic 2SP

l) Lambert Conformal Conic 2SP Belgium

m) Mercator 1SP

n) Mercator 2SP

o) Cassini Soldner

p) Oblique Stereographic

q) Polar Stereographic

r) New Zealand Map Grid

s) Hotine Oblique Mercator

t) Laborde Oblique Mercator

u) Swiss Oblique Cylindrical

v) Oblique Mercator

w) Tunisia Mining Grid

Annex C
(normative)

DCP independent interface specification

~~1~~C.1 Introduction

This annex specifies in detail the general coordinate transformation interface, independent of any specific Distributed Computing Platform (DCP). The architecture and overview of this general interface is provided by Clause 5. This detailed specification is provided in Unified Modeling Language (UML) notation, with the extensions specified in Subclause 4.2.

The ground coordinate transformation service performs coordinate transformations between many pairs of different ground coordinate reference systems. This service defines five operations, and uses ground coordinates sets of contents, alternatives, and options for Coordinate Reference System definitions and Transformation Definitions. The following subclauses specify in detail the DCP-independent interface to a ground coordinate transformation service.

This ground coordinate transformation service can support all the types of coordinate reference systems listed in Subclause 5.2.2. For this ground coordinate transformation service interface, the coordinate systems will often be well-known to both the server and client software. For any coordinate system that is well-known, the Coordinate Reference System definition need not include the full specification, but can be limited to only the Identifier. This Identifier is defined to be sufficient to allow the server or client to obtain more information from another source. This other source can be internal to the client or server software, or can be a third party source such as a web site maintained by a coordinate system definition authority.

This ground coordinate transformation service can also support all the types of parameterized coordinate transformation listed in Subclause B.4. For this ground coordinate transformation service interface, there will often be only one reasonable or implemented coordinate transformation between the source and target coordinate reference system. For example, this will occur if the source or target coordinate reference system identifies or implies the proper coordinate transformation. Even when there is more than one available coordinate transformation between the source and target coordinate reference systems, the client may not choose to specify which transformation is desired. In both of these cases, the Transformation Definition transferred from client to server shall be empty. Also, a Transformation Definition transferred from server to client can be empty.

The constraints on the sequence of invoking various operations are specified in Subclause C.3. The provided descriptions of the operations do not mention the conditions under which execution of that operation will fail. These failure conditions and behavior are discussed in Subclause C.4.

1C.2 UML model

Figure C.1 contains the UML class diagram for the ground coordinate transformation service, including all five specified operations.

NOTE This UML class diagram does not include the UML classes that could be used to describe the contents of the XML or WKT character strings which are used to encode CoordinateReferenceSystem and TransformationDefinition. The contents and options within these two character strings capture a significant part of the complexity of this ground coordinate transformation service interface.

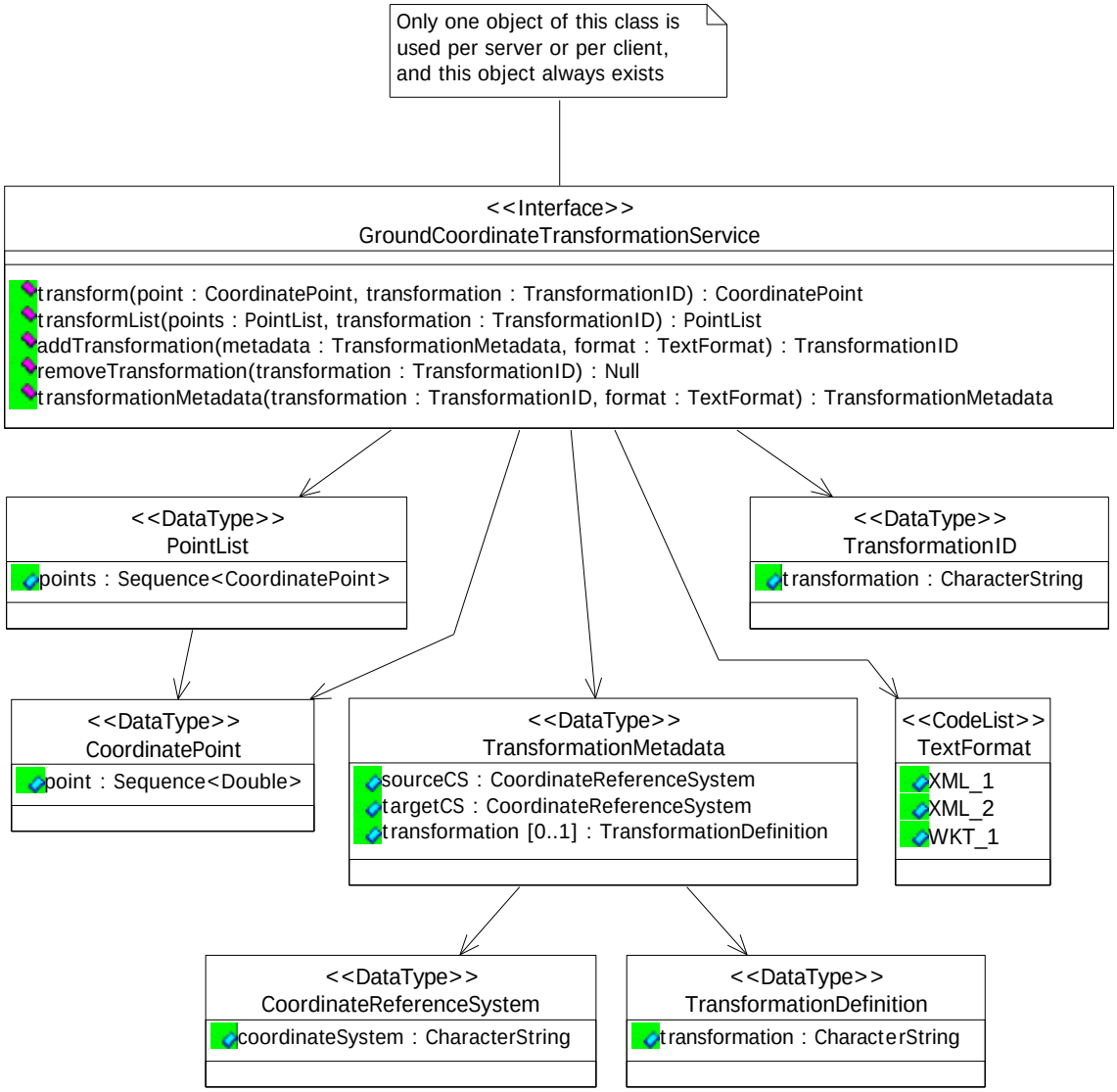

Figure C.1 – Ground coordinate transformation service class diagram

Rational Rose documentation produced from the above UML model is provided in the following subclauses, one for each class shown in the diagram.

1C.2.1 GroundCoordinateTransformationService interface class

Provides ground coordinate transformation services, to one or more clients. Only one object of this class is used per server or per client, and this object always exists.

Public Operations:

1transform (point : CoordinatePoint, transformation : TransformationID) : CoordinatePoint

Transforms multi-dimensional coordinates of one point, from the source coordinate system to the target coordinate system. The Transformation ID input identifies the source and target coordinate reference systems.

1transformList (points : PointList, transformation : TransformationID) : PointList

Transforms multi-dimensional coordinates of a list of zero or more points, from the source coordinate system to the target coordinate system. The Transformation ID input identifies the source and target coordinate reference systems.

1addTransformation (metadata : TransformationMetadata, format : TextFormat) : TransformationID

Gets the identifier of the coordinate transformation for the specified Transformation Metadata. This metadata specifies the source and target coordinate systems, plus the selected transformation alternative and/or the specified transformation parameters. This operation also uses a Text Format input that specifies the format used by the Transformation Metadata input. If the source and target coordinate system definitions are reversed in the input Transformation Metadata, the identifier of the inverse transformation will be obtained, assuming the coordinate transformation is one to one.

1removeTransformation (transformation : TransformationID) : Null

Deactivates the specified coordinate transformation for a client, allowing the freeing of server resources.

1transformationMetadata (transformation : TransformationID, format : TextFormat) : TransformationMetadata

Gets metadata defining the specified coordinate transformation, including metadata defining the source coordinate reference system, target coordinate reference system, and selected coordinate transformation alternative and/or specified transformation parameters. This operation also uses a Text Format input that specifies the format to be used by the Transformation Metadata output.

1C.2.2 CoordinatePoint data type class

Position of one point defined by list of values of one or more ordinates. The number of ordinate values included in the list is the number of dimensions of the coordinate reference system the point is defined in, either source or target coordinates.

Public Attributes:

point : Sequence<Double>

List of values of the ordinates of one point

C.2.3 PointList data type class

List of the coordinates of zero or more points, normally used for one or more points. The contents of this list are equivalent to a sequence of zero or more Coordinate Points. The number of ordinate values included for each point must be the number of dimensions of the one coordinate reference system that all the points are defined in, either source or target coordinates. The same point coordinates can be repeated in this list.

Public Attributes:

points : Sequence<CoordinatePoint>

List of coordinates of zero or more points, usually one or more points.

C.2.4 TextFormat code list class

Code that identifies the format for the related Transformation Metadata.

Public Attributes:

XML_1 :

Code for first XML format. (This format has not yet been selected.)

XML_2 :

Code for second XML format. (This format has not yet been selected.)

WKT_1 :

Code for first Well Known Text (WKT) format. (This format has not yet been selected.)

C.2.5 TransformationID data type class

Character string identifier of one specific coordinate transformation, from one source coordinate system to one target coordinate system. All Transformation ID values must be respected for the lifetime of a server object. A proper Transformation ID value is obtained using the "getTransformation" operation.

Public Attributes:

transformation : CharacterString

Identifier of one specific coordinate transformation, from one source coordinate system to one target coordinate system. The format of this character string is arbitrary and chosen by the server. The string contents are not interpretable by a client, and are not shared by different server software.

~~1~~C.2.6 TransformationMetadata data type class

Metadata for one specific coordinate transformation, including metadata for the source coordinate reference system, the target coordinate reference system, and, when needed, the transformation choice and/or transformation parameters. This metadata is all recorded in the format specified by the value of the related Text Format.

Public Attributes:

~~1~~sourceCS : CoordinateReferenceSystem

Definition of the source coordinate reference system, the system used by the coordinates input to a coordinate transformation.

~~1~~targetCS : CoordinateReferenceSystem

Definition of the target coordinate reference system, the system used by the coordinates output from a coordinate transformation.

~~1~~transformation [0..1] : TransformationDefinition

Optional definition of one coordinate transformation, including specification of transformation parameters and/or selection from transformation alternatives.

~~1~~C.2.7 CoordinateReferenceSystem data type class

Definition of one coordinate reference system, including the coordinate system identification and/or its complete specification.

Public Attributes:

~~1~~coordinateSystem : CharacterString

Definition of a coordinate reference system, encoded in text using format specified by the value of the related Text Format.

~~1~~C.2.8 TransformationDefinition data type class

Definition of one coordinate transformation, including the coordinate transformation identification and/or its specification, but not including the definitions of the source and target coordinate reference systems. For input to a server, this Transformation Definition value will be empty when there is only one transformation between the two related coordinate systems, or when a specific transformation is not selected by the client.

Public Attributes:

~~1~~transformation : CharacterString

Definition of a coordinate transformation, encoded in text using format specified by the value of the related Text Format.

1C.3 Object state and operation sequence constraints

The ground coordinate transformation service object has no client-visible dynamic state except the set of Transformation ID values that are currently recognized by that object, each ID with the corresponding Transformation Metadata. For each specific Transformation ID, the allowed sequence of invoking the various implemented operations is limited. Specifically, for each specific Transformation ID:

a) The addTransformation operation must first be invoked and completed, to obtain a valid value for the Transformation ID.

b) When implemented, the removeTransformation operation must be invoked last, and will make that Transformation ID invalid.

c) All other implemented operations can each be invoked at any time, in any order, after the corresponding addTransformation operation is completed and before the corresponding removeTransformation operation is invoked.

1C.4 Failure conditions and exceptions

The operation descriptions provided above do not specify the conditions under which each specified operation can fail, and thus not produce correct outputs. These failure conditions and behavior are discussed in this subclause. In general, the occurrence of any failure condition will produce the normal failure behavior of the DCP. For example, for a DCP that supports exceptions (such as CORBA), the failure behavior will be to throw an exception. A variety of exceptions are used, that mean:

a) Invalid input argument:

   1) One or more input point coordinates (in a Coordinate Point or a Point List) are not within the valid range of the (corresponding) specified coordinate transformation.

   2) One or more input point coordinates have the wrong number of dimensions for the source coordinate reference system, specified in the ~~(~~corresponding~~)~~ Transformation ID value.

b) Capability not supported by this server implementation:

   1) Operation called is not implemented by this server.

   2) Input Transformation ID value is not (currently) supported by this server.

   3) Coordinate transformation identified, in Transformation Metadata input, is not supported by this server.

   4) Source or target coordinate reference system identified, in Transformation Metadata input, is not supported by this server.

   5) Input Text Format value is not supported by this server, for the related Transformation Metadata input or output.

6) Some required external resource is not currently available to this server.

c) Computation problems:

1) Some computation error occurred during processing.

Annex D
(normative)

CORBA interface profile

D.1 Introduction

This annex specifies in detail the CORBA profile of the general coordinate transformation interface. This CORBA profile is based on the DCP independent interface specification in Annex C. This Annex D includes a copy of some of the text in Annex C, but not all. In particular, Subclause C.3 is not copied in Annex D, but is normative for this CORBA profile.

This profile includes some details and variations that depend on the CORBA Distributed Computing Platform (DCP). The architecture and overview of this CORBA profile is discussed by Subclause 6.2. This detailed specification is provided in the ISO/OMG Interface Definition Language (IDL) notation, as specified in ISO/IEC 14750.

~~1~~D.2 CORBA IDL

```
// Module: Ground Coordinate Transformation
// Function: The Ground Coordinate Transformation Service
//    performs coordinate transformations between many pairs of
//    different ground coordinate systems.
// Uses modules: None

// Shared data type definitions:

// Data Type: Coordinate Point: Position of one point defined by
//    list of values of one or more ordinates. The number of
//    ordinate values included in the list is the number of
//    dimensions of the coordinate reference system the point
//    is defined in, either source or target coordinates.
   typedef sequence <double>   CoordinatePoint;

// Data Type: Point List: List of the coordinates of zero or
//    more points, normally used for one or more points. The
//    contents of this list are equivalent to a sequence of zero
//    or more Coordinate Points. The number of ordinate values
//    included for each point must be the number of dimensions
//    of the one coordinate reference system that all the
//    points are defined in, either source or target
//    coordinates. The same point coordinates can be repeated
//    in this list.
   typedef sequence <CoordinatePoint>   PointList;

// Data Type: Transformation ID: Character string identifying one
//    specific coordinate transformation, from one source
//    coordinate system to one target coordinate system. All
```

```
//    TransformationID values must be respected for the
//    lifetime of a server object. A proper TransformationID
//    value is obtained using the "getTransformation"
//    operation. The format of this character string is
//    arbitrary and chosen by the server. The string contents
//    are not interpretable by a client, and are not shared by
//    different server software.
   typedef string   TransformationID;

// Data Type: Coordinate Reference System: Definition of one
//    coordinate reference system, including the coordinate
//    system identification and/or its complete specification.
//    This definition is encoded in text using the format
//    specified by the value of the related Text Format.
   typedef string   CoordinateReferenceSystem;

// Data Type: Transformation Definition: Definition of one
//    coordinate transformation, including the coordinate
//    transformation identification and/or its specification, but
//    not including the definitions of the source and target
//    coordinate reference systems. For input to a server, this
//    Transformation Definition value will be empty when there is
//    only one transformation between the two related coordinate
//    systems, or when a specific transformation is not selected
//    by the client. This definition is encoded in text using the
//    format specified by the value of the related Text Format.
   typedef string   TransformationDefinition;

// Data Type: Transformation Metadata: Metadata for one specific
//    coordinate transformation, including metadata for the
//    source coordinate reference system, the target reference
//    coordinate system, and, when needed, the transformation
//    choice and/or transformation parameters. This metadata is
//    all recorded in the format specified by the value of the
//    related Text Format.
   struct TransformationMetadata {
      CoordinateReferenceSystem   sourceCS;
//       Definition of the source coordinate reference system,
//       the system used by the coordinates input to a
//       coordinate transformation.
      CoordinateReferenceSystem   targetCS;
//       Definition of the target coordinate reference system,
//       the system used by the coordinates output from a
//       coordinate transformation.
      TransformationDefinition     transformation;
//       Optional definition of one coordinate transformation,
//       including specification of transformation parameters
//       and/or selection from transformation alternatives.
      };
```

```
// Exception definitions:
// All specified operations can raise the same set of three use-
//    specific exceptions specified below. In addition, all
//    operations can raise any CORBA defined standard exception.
//    The CORBA standard exceptions take precedence over these
//    use-specific exceptions.

// Type: Exception Body: Contains information describing a
//    raised exception.
   struct ExceptionBody {
      string   exceptionType;
//       Contains standardized, user-understandable text uniquely
//          identifying this type of error.
      string   userErrorMessage;
//       Contains non-standardized, user-understandable text
//          further describing the specific error condition
//          detected. This information is likely to pinpoint the
//          error, and might include the name of the operation.
      string   userRecoveryMessage;
//       Contains non-standardized, user-understandable text
//          suggesting error recovery methods. This string can be
//          empty, when no information is available. This string
//          can include a URL for obtaining more information.
      string   developerErrorMessage;
//       Contains non-standardized, user-understandable text
//          providing more detail of possible use to software
//          developers and testers, for client and server
//          software. This information is likely to pinpoint the
//          point in the server implementation software where the
//          error was detected. This string can be empty, when no
//          information is available. This string can include a
//          URL for obtaining more information.
      };

// Exception: Invalid Input: Exception raised when any input
//    provided to the operation was not correct, and the called
//    operation thus could not be completed.
   exception InvalidInput {ExceptionBody  exceptionText;};

// Standard values for Invalid Input exceptionType:
      const string   OutOfRange         =
“One or more input point coordinates are not within valid domain of
specified coordinate transformation.”;
     const string   WrongDimensions    =
“One or more input point coordinates have different number of
dimensions than specified source coordinate system.”;
     const string   OtherInputError    =
“Some other input error was detected.”;
// For the Invalid Input exception, the userErrorMessage contains
//    text providing information such as which inputs are not
//    correct or are inconsistent.
```

```
// Exception: Not Supported: Exception raised when any requested
//    capability is not supported by this server, and the called
//    operation thus could not be completed.
   exception NotSupported {ExceptionBody  exceptionText;};

// Standard values for Not Supported exceptionType:
      const string   OperationNotImplemented        =
“Operation called is not implemented by this server.”;
      const string   TransformationIdNotSupported   =
“One or more input TransformationID value is not (currently)
supported by this server.”;
      const string   TransformationNotSupported     =
“Specified coordinate transformation is not supported by this
server.”;
      const string   CoordinateSystemNotSupported   =
“Specified source and/or target coordinate system is not supported
by this server.”;
      const string   TextFormatNotSupported         =
“Input Text Format value is not supported by this server.”;
      const string   ResourceNotAvailable           =
“Some required external resource is not available to this server.”;
      const string   OtherNotSuppoprted             =
“Some other non-supported capability was needed.”;
// For the Not Supported exception, the userErrorMessage contains
//    text providing information such as which operation,
//    transformation, coordinate system, format, or resource is
//    not (currently)supported.

// Exception: Computation Problem: Exception raised when any
//    computation or other problem was detected, and the called
//    operation thus could not be completed.
   exception ComputationProblem {ExceptionBody  exceptionText;};

// Standard values for Computation Problem exceptionType:
      const string   OtherComputationError       =
“Some other computation error was detected.”;
// For the Computation Problem exception, the userErrorMessage
//    contains information such as how the computation failed
//    (for example, iteration did not converge)

// End of exception definitions.

// End of shared data type definitions


// Begin module: Ground Coordinate Transformation
module GroundCoordinateTransformation {
```

```
// Local data type definitions:

// Code List: Text Format: Code that identifies the format for
//    the related Transformation Metadata.
   typedef string   TextFormat;
//    Character string encoding of format code.
//    Code for first XML format.
         const string XML_1 = "XML_1";
//    Code for second XML format.
         const string XML_2 = "XML_2";
//    Code for first Well Known Text (WKT) format.
         const string WKT_1 = "WKT_1";

// End of local data type definitions.


// Interface: Ground Coordinate Transformation Service
// Function: Provides ground coordinate transformation services,
//    to one or more clients. Only one object of this class is
//    used per server or per client, and this object always
//    exists.

// Concrete, implemented interface class.
// Other interface objects used: None
// Inherits interfaces: None

interface GroundCoordinateTransformationService {

// Public Attributes: None
// Public Operations:

// Operation: Transform
// Function: Transforms multi-dimensional coordinates of one
//    point, from the source coordinate system to the target
//    coordinate system. The TransformationID input identifies
//    the source and target coordinate reference systems.
   void transform (
      in CoordinatePoint    inputPoint,
      in TransformationID   transformation,
      out CoordinatePoint   outputPoint)
      raises (InvalidInput, NotSupported, ComputationProblem);

// Operation: Transform List
// Function: Transforms multi-dimensional coordinates of a list
//    of zero or more points, from the source coordinate system
//    to the target coordinate system. The TransformationID input
//    identifies the source and target coordinate reference
//    systems.
   void transformList (
      in PointList          inputPoints,
      in TransformationID   transformation,
      out PointList         outputPoints)
```

```
      raises (InvalidInput, NotSupported, ComputationProblem);

// Operation: Add Transformation
// Function: Gets the identifier of the coordinate transformation
//    for the specified Transformation Metadata. This metadata
//    specifies the source and target coordinate reference
//    systems, plus the selected transformation alternative
//    and/or the specified transformation parameters. This
//    operation also uses a Text Format input that specifies the
//    format used by the Transformation Metadata input. If the
//    source and target coordinate reference systems are reversed
//    in the input Transformation Metadata, the identification of
//    the inverse transformation will be obtained, assuming the
//    coordinate transformation is one to one.
   void addTransformation (
      in TransformationMetadata   metadata,
      in TextFormat               format,
      out TransformationID        transformation)
      raises (InvalidInput, NotSupported, ComputationProblem);

// Operation: Remove Transformation
// Function: Deactivates the specified coordinate transformation
//    for a client, allowing the freeing of server resources.
   void removeTransformation (
      in TransformationID         transformation)
      raises (InvalidInput, NotSupported, ComputationProblem);

// Operation: Transformation Metadata
// Function: Gets metadata defining the specified coordinate
//    transformation, including metadata defining the source
//    coordinate reference system, target coordinate reference
//    system, and selected coordinate transformation alternative
//    and/or specified transformation parameters. This operation
//    also uses a Text Format input that specifies the format to
//    be used by the Transformation Metadata output.
   void transformationMetadata (
      in TransformationID          transformation,
      in TextFormat                format,
      out TransformationMetadata   metadata)
      raises (InvalidInput, NotSupported, ComputationProblem);

}; // End of interface: Ground Coordinate Transformation Service

}; // End of module: Ground Coordinate Transformation
```

Annex E
(normative)

DCOM interface profile

TBD

Annex F
(normative)

XML contents and formats

~~1~~F.1 Introduction

This annex specifies the XML contents and format used by the specified coordinate transformation interfaces, including the many contents alternatives and options. These XML formats provide for the transfer of Coordinate Reference System definitions and Coordinate Transformation Definitions. The same XML formats are used by all DCP-specific profiles, of the general DCP-independent interfaces. Some XML capabilities are optional implementation, as specified in Subclause B.2.

These XML contents, alternatives, and options are a profile or modified subset of the OGC standard definition data XML now proposed in OGC document 01-014. Some elements and attributes of that proposed OGC standard XML are omitted here, because they are not useful in these coordinate transformation interfaces.

These XML formats are now stated in the form of XML DTDs. In future revisions, these DTDs are likely to be converted to XML Schema. The current XML DTDs are written as external DTDs that are not included with actual XML data. Each DTD includes extensive comments to explain the actual XML data, using a format that we find useful. These DTDs are currently written such that they must all be included in one DTD file. These DTDs are written for XML version 1.0. For uncertain points, this draft uses the abbreviation “TBR” for “To Be Reviewed” and “TBD” for “To Be Determined”.

This document specifies standard XML in three parts, for representation of:

a) Coordinate Transformation Definition, including Coordinate Transformation Metadata

b) Coordinate Reference System definition, used by above

c) Shared Elements, shared by both above

These three parts are presented in the following subclauses. For each part, a commented XML DTD is given. Since the extensive comments included in the commented XML DTDs make it difficult to quickly scan these DTDs, these DTDs are repeated in Annex G (informative) with all comments removed.

NOTE ~~1~~ This information was copied from OGC document 01-014, and should be updated when improvements are made to that document. The information copied does not include the XML elements that are not considered needed in this Implementation Specification, as summarized in Subclause G.1

~~NOTE 2 This XML can be modified if needed for this Implementation Specification. For example, more of the elements and attributes included in 01-014 could be eliminated if not considered needed. Additional attributes and elements could be added, to contain additional information needed. Also, the structure could be modified if considered desirable, such as to simplify the XML or the software which produces and uses~~

~~it. For example, the Entity construct might be used instead of some XML Elements, such as for the Identifier.~~

~~1~~F.2 XML DTDs for coordinate transformation definition

This subclause presents the commented XML DTDs used for transfer of a Coordinate Transformation Definition or Coordinate Transformation Metadata. The Coordinate Transformation Definition XML element, with all needed components, shall be used for ~~a~~ the Coordinate Transformation Definition data type in the Ground Coordinate Transformation Service UML model. Alternately, the Coordinate Transformation Metadata XML element, with all needed components, shall be used for a complete Transformation Metadata data type in the UML model.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Version 0.0 of XML DTD for Coordinate Transformations. This
DTD uses XML elements that are specified in other DTDs. -->

<!-- Expected usage: Either the Coordinate Transformation
Metadata or Coordinate Transformation Definition can be directly
used under different circumstances:

    The Coordinate Transformation Metadata element should be used
    when definitions of both the source and target coordinate
    reference systems are included.  In that case, the Coordinate
    Transformation Definition component element is optional.

    The Coordinate Transformation Definition element should be
    directly used when the source and target coordinate reference
    systems are NOT included. This element could thus be used in
    the current "low-level" Coordinate Transformation Services
    interface specification.
-->

<!-- Coordinate Transformation Metadata: Metadata for one
specific coordinate transformation, including definitions of the
source coordinate reference system, target reference coordinate
system, and coordinate transformation parameters and/or
transformation choice. All of this metadata can be recorded in
this XML format. The source and target coordinate reference
systems can each be for any number of axes or ordinates included
in the coordinates of a point, from one to four or more axes.

The Coordinate Transformation Metadata data contains:

    Source Coordinate Reference System: The source coordinate
    reference system for this transformation, including the
    coordinate reference system identification and/or its
    complete specification. This Source Coordinate Reference
    System is used by the input point coordinates to a coordinate
    transformation.
```

```
    Target Coordinate Reference System: The target coordinate
    reference system for this transformation, including the
    coordinate reference system identification and/or its
    complete specification. This Target Coordinate Reference
    System is used by the output point coordinates from a
    coordinate transformation.

    Coordinate Transformation Definition: One coordinate
    transformation, including specification of transformation
    parameters and/or selection from transformation alternatives.
    This definition includes the transformation identification
    and/or its specification, but not the definitions of the
    source and target coordinate reference systems. This
    Transformation Definition is optional, included only when
    needed or useful. For example, for input to a coordinate
    transformation server, this Coordinate Transformation
    Definition can be omitted when the server software does not
    require this information.
-->
<!ELEMENT CoordinateTransformationMetadata (
        CoordinateReferenceSystem,
        CoordinateReferenceSystem,
        CoordinateTransformationDefinition?) >

<!—- Coordinate Transformation Definition: Definition of one
coordinate transformation, including the coordinate
transformation identification and/or its specification, but not
including the definitions of the source and target coordinate
systems. This transformation can be for any number of axes or
ordinates included in the coordinates of a point, from one to
four or more axes.

The Coordinate Transformation Definition data contains:

    Name Set: Set of one or more human understandable names for
    this coordinate transformation. This Name Set is optional,
    but usually should be included as discussed for the Name Set
    element.

    Identifier: Unique identifier used by software for this
    coordinate transformation. Either this Identifier or the
    following Coordinate Transformation Specification must be
    included. Both can be included if useful, as discussed for
    the Identifier element.

    Coordinate Transformation Specification: Detailed
    specification of this specific coordinate transformation,
    between the related source and target coordinate reference
    systems. This Coordinate Transformation Specification data
    includes:
```

```
        Source Dimensions: Positive integer number of dimensions
        in a source coordinate reference system to which this
        transformation can be applied.

        Target Dimensions: Positive integer number of dimensions
        in a target coordinate reference system that this
        transformation can produce.

        Validity Region: Description of the region of validity of
        this coordinate transformation, in the source coordinate
        references system. This data is optional, included
        whenever the region of validity is known and can be
        usefully represented.

        Transformation Type Specification: Specification of one
        of a set of alternative types of coordinate
        transformation.

The current coordinate transformation types are:

    Concatenated Transformation: Specification of a coordinate
    transformation that sequentially applies two or more other
    transformations.

    Inverse Transformation: Specification of a coordinate
    transformation that is the inverse of another transformation.

    Pass Through Transformation: Specification of a coordinate
    transformation that transforms only a subset of the
    ordinates, and passes the remaining ordinates unchanged. The
    modified ordinates are transformed by a referenced Coordinate
    Transformation.

    Parameterized Transformation: Specification of a coordinate
    transformation that is defined by a specified transformation
    type and set of parameters.
-->
<!ELEMENT CoordinateTransformationDefinition (
        NameSet?,
        (Identifier
        | (sourceDimensions,
          targetDimensions,
          ValidityRegion?,
          ( ConcatenatedTransformation
          | InverseTransformation
          | PassThroughTransformation
          | ParameterizedTransformation),
          Identifier?) ) ) >
<!ELEMENT sourceDimensions   (#PCDATA) >
<!ELEMENT targetDimensions   (#PCDATA) >
```

```
<!-- Concatenated Transformation: Specification of a coordinate
transformation that sequentially applies two or more other
coordinate transformations. The target coordinate reference
system from the first coordinate transformation must be the
source coordinate reference system to the second coordinate
transformation, and so forth. This restriction holds whether or
not those target and source coordinate reference systems are
explicitly defined. (Among other things, the number of dimensions
of the output coordinate system from the first transformation
must match the number of dimension of the input coordinate to the
second transformation, and so forth.)

The Concatenated Transformation data contains:

    List of Coordinate Transformation Definitions: Ordered list
    of the two or more individual coordinate transformations that
    are combined in this Concatenated Transformation. The order
    of listing is the order of transformation application, to go
    from the associated source to the associated target
    coordinate reference systems.
-->
<!ELEMENT ConcatenatedTransformation (
        CoordinateTransformationDefinition+) >

<!-- Inverse Transformation: Specification of a coordinate
transformation that is the inverse of another coordinate
transformation. That is, this inverse transformation transforms
coordinates in the opposite direction between the same two
coordinate reference systems. (Note: The Inverse Transform will
not exist if the referenced coordinate transformation is not one
to one. All map projections should succeed.)

The Inverse Transformation data contains:

    Coordinate Transformation Definition: Definition of the
    coordinate transformation that is inverted (or reversed) by
    this Inverse Transformation.
-->
<!ELEMENT InverseTransformation (
        CoordinateTransformationDefinition) >

<!-- Pass Through Transformation: Specification of a coordinate
transformation that transforms only a subset of the ordinates,
and passes the remaining ordinates unchanged. The ordinates that
are modified are transformed by a referenced Coordinate
Transformation. One major use of a Pass Through Transformation is
transforming one component of a compound coordinate system.

    An example: If you have (Latitude, Longitude, Height)
    coordinates, this transformation allows converting the Height
    above the ellipsoid to the Elevation above mean sea level,
    without affecting the Latitude and Longitude values. (This
```

~~transformation might be modeled as having a Geographic 3D coordinate reference system as source and a Compound coordinate reference system as target.) Alternately, this type of transformation allows changing the Prime Meridian of the Longitude values.~~

~~Another~~ An example: If you have (Latitude, Longitude, Height) coordinates, this transformation allows converting the Height values from meters to feet, without affecting the Latitude and Longitude values. Alternately, this transformation allows converting the Latitude and Longitude values from degrees to radians, without affecting the Height value. Of course, either of these physical Unit conversions could alternately be done using a degenerate affine Parameterized Transformation. Note that in such a unit conversion, two coordinate reference systems are involved: the source CRS (with the original unit) and the target CRS (with the target unit). Both of these CRSs shall be referenced in the Coordinate Transformation Metadata, when that XML element is used.

The Pass Through Transformation data contains:

Coordinate Transformation Definition: Definition of the base coordinate transformation that is used by this Pass Through Transformation to transform the coordinate axes that are modified.

First Modified Ordinate: Positive integer index to the first ordinate to be transformed by the identified coordinate transformation.

Number Modified Ordinates: Positive integer number of dimensions to be transformed by the identified coordinate transformation.

```
-->
<!ELEMENT PassThroughTransformation (
        CoordinateTransformationDefinition,
        firstModifiedOrdinate,
        numberModifiedOrdinates) >
<!ELEMENT firstModifiedOrdinate     (#PCDATA) >
<!ELEMENT numberModifiedOrdinates   (#PCDATA) >

<!-- End of XML DTD for Coordinate Transformation Definition -->


<!-- Version 0.0 of XML DTD for Parameterized Transformation
definition. This DTD uses XML elements that are specified in
other DTDs. -->

<!-- Parameterized Transformation: Specification of a coordinate
transformation that is defined by a specified transformation
```

```
method plus the set of parameter values needed by that
transformation method. Each parameter is recorded with its name,
value, and physical unit when applicable.
    Notes: The number of dimensions of the target coordinate
    system is always equal to or less than the number of
    dimensions of the source coordinate system. The specified
    coordinate transformation should be one-to-one within the
    domain of the source coordinate system.

The Parameterized Transformation data contains:

    Transformation Method: Identification of the method used by
    this Parameterized Transformation.

    List of Parameters: Unordered list of the zero or more
    parameters that define the specific coordinate transformation
    of the associated transformation method.
-->
<!ELEMENT ParameterizedTransformation (
        TransformationMethod,
        Parameter*) >

<!-- Parameter: Data structure containing the name and value of
one coordinate transformation or map projection parameter. The
Parameter data contains:

    Code Name: Character string name or identifier of this
    transformation parameter. This name implies the required type
    of the parameter value. This Code Name should not specify or
    imply the units used; the units used shall be specified in
    the associated Unit element. The valid Code Names are defined
    by the associated Transformation Method, including the method
    Identifier. This name should be a standard, widely-used
    parameter name whenever applicable.

    Value: Value of this transformation parameter, often a double
    precision floating point numeric value. However, a parameter
    value can alternately be an integer, string, boolean, list of
    integers, or list of double precision floating point values.

    Unit: Physical unit used for this parameter value.  This unit
    can be either a Linear or Angular unit. This data is
    optional, but should be included whenever applicable.
        The value Unit is variable for some parameter Code Names
        for some Transformation Methods, and the actual Unit used
        by the value must be specified. The value Unit is fixed
        for other parameter Code Names for some Transformation
        Methods, but the actual unit should be included for human
        understanding and/or software checking. There is no
        applicable physical unit for some parameter Code Names
        for some Transformation Methods, and the Unit data is
        then omitted.
```

```
-->
<!ELEMENT Parameter (
        codeName,
        value,
        (LinearUnit | AngularUnit)? ) >
<!ELEMENT codeName   (#PCDATA) >
<!ELEMENT value      (#PCDATA) >

<!-- Transformation Method: Identification of the method used by
a Parameterized Transformation.

The Transformation Method data contains:

    Name Set: Set of one or more human understandable names for
    this Transformation Method. This Name Set is optional, but
    usually should be included as discussed for the Name Set
    element.

    Identifier: Unique identifier used by software for this
    Transformation Method.
-->
<!ELEMENT TransformationMethod (
        NameSet?,
        Identifier) >

<!-- Transformation Methods: The possible names of Transformation
Methods, with some of their meanings, include:

    Affine: Transformation using matrix multiplication, of the
    source coordinates vector to produce the target coordinates
    vector. An affine transformation transforms all straight
    lines into other straight lines. An affine transformation can
    perform translation, rotation, scaling, and shearing.

    Ellipsoid To Geocentric: Transformation from 3D geographical
    coordinates to 3D geocentric Cartesian coordinates. The first
    input ordinate is the Longitude, and the second input
    ordinate is the Latitude. The third input ordinate is the
    Height above the ellipsoid. In the output geocentric
    coordinates, the X axis is through the intersection of the
    Greenwich meridian and equator, and the Y axis is through the
    intersection of the equator with Longitude 90 degrees E.  The
    Z axis is through the Polar axis positive northwards.

    Geocentric To Ellipsoid: Transformation from 3D geocentric
    Cartesian coordinates to 3D geographical coordinates. In the
    input geocentric coordinates, the X axis is through the
    intersection of the Greenwich meridian and equator, and the Y
    axis is through the intersection of the equator with
    Longitude 90 degrees E.  The Z axis is through the Polar axis
    positive northwards. The first output ordinate is the
```

Longitude, and the second output ordinate is the Latitude. The third output ordinate is the Height above the ellipsoid.

- Abridged Molodenski: As an alternative to computation of Latitude, Longitude, and Height above ellipsoid in discrete steps through geocentric coordinates, the changes in these coordinates may be computed directly by formulas derived by Molodenski. Abridged versions of these formulas are quite satisfactory for three parameter transformations. The Abridged Molodenski transformation can operate on 2D or 3D coordinates. In either case, the first ordinate is the Latitude in degrees, and the second ordinate is the Longitude in degrees. In the 3D form, the third ordinate is the Height above the ellipsoid.

- Bursa Wolf: An approximate transformation from one Geodetic Datum to another Geodetic Datum. A Bursa Wolf transformation is applied to the geocentric coordinates of one Geodetic Datum to produce the corresponding geocentric coordinates of another Geodetic Datum. A Bursa Wolf transformation is sometimes used for transformation from another Geodetic Datum into WGS84. A Bursa Wolf transformation is defined by a list of up to 7 transformation parameters. Sometimes, only the first three or six parameters are specified, with the remaining parameters being zero. This transformation is only an approximation; for a given Geodetic Datum, different Bursa Wolf transformations can be used to minimize the errors over different regions.

- Longitude Rotation: This transformation adds a constant to the Longitude in (Longitude, Latitude) or (Longitude, Latitude, Height) coordinates. This transformation is useful for handling changes in the Prime Meridian. The Longitude is assumed to be the first ordinate, and the Latitude is assumed to be the second ordinate. Any subsequent ordinates are left unchanged. All Longitudes should be in degrees, in the range (-180, 180). The Latitude values should be in degrees, in the range (-90, 90). If the Latitude value is not in the range (-90, 90), then the Longitude value should be 0. These rules apply to input and output Longitude and Latitude values.

There are many different ways of classifying map projection transformation types. Here is one set:

- Transverse Mercator:
- Transverse Mercator South Oriented:
- Molodenski:
- Lambert Conformal Conic 1SP:
- Lambert Conformal Conic 2SP:
- Lambert Conformal Conic 2SP Belgium:
- Mercator 1SP:
- Mercator 2SP:
- Cassini Soldner:

- Oblique Stereographic:
- Polar Stereographic:
- New Zealand Map Grid:
- Hotine Oblique Mercator:
- Laborde Oblique Mercator:
- Swiss Oblique Cylindrical:
- Oblique Mercator:
- Tunisia Mining Grid:

Note that some of the above parameterized transformation types will use parameters that define ellipsoids, prime meridians, datums, units, and other data used in Coordinate Reference System definitions. Also note that definitions of most of the above listed parameterized transformations are provided in the EPSG database, with all the associated parameter names, definitions, and values. These and other parameterized-transformation definitions are currently available through the web page: http://www.remotesensing.org/geotiff/proj_list/. The documentation of these methods and the formulas that go with the parameter sets are on the web page: http://www.posc.org/Epicentre.2_2/DataModel/ExamplesofUsage/eu_cs.html. Documentation is also available at the EPSG web site at: http://www.petroconsultants.com/products/geodetic.html.

The EPSG specifies a set of transformation methods, including:

EPSG code	Transformation Method Name (EPSG)
9601	Longitude rotation
9602	Geographic/geocentric conversions
9603	Geocentric translations
9604	Molodenski
9605	Abridged Molodenski
9606	Position Vector 7-param. transformation
9607	Coordinate Frame rotation
9613	NADCON
9614	NTv1
9615	NTv2
9616	Vertical Offset
9617	Madrid to ED50
9618	Geographical and Height Offsets
9619	Geographical Offsets
9620	Norway Offshore Interpolation
9621	Similarity transformation
9622	Affine orthogonal geometric transformation
9623	Affine general geometric transformation
9624	Affine general parametric transformation
9625	General polynomial (2nd-order)
9626	General polynomial (3rd-order)
9627	General polynomial (4th-order)
9628	Reversible polynomial (2nd-order)
9629	Reversible polynomial (3rd-order)
9630	Reversible polynomial (4th-order)
9631	Complex polynomial (3rd-order)

```
9632        Complex polynomial (4th-order)
9801        Lambert Conic Conformal (1SP)
9802        Lambert Conic Conformal (2SP)
9803        Lambert Conic Conformal (2SP Belgium)
9804        Mercator (1SP)
9805        Mercator (2SP)
9806        Cassini-Soldner
9807        Transverse Mercator
9808        Transverse Mercator (South Orientated)
9809        Oblique Stereographic
9810        Polar Stereographic
9811        New Zealand Map Grid
9812        Hotine Oblique Mercator
9813        Laborde Oblique Mercator
9814        Swiss Oblique Cylindrical
9815        Oblique Mercator
9816        Tunisia Mining Grid
9817        Lambert Conic Near-Conformal
9818        American Polyconic
9819        Krovak Oblique Conic Conformal
-->

<!-- End of XML DTD for Parameterized Transformation -->
```

1F.3 XML DTDs for coordinate reference system definition

This subclause presents the commented XML DTDs used for transfer of a Coordinate Reference System definition. The Coordinate Reference System XML element, with all needed components, shall be used for a Coordinate Reference System data type in the Ground Coordinate Transformation Service UML model.

```
<!-- Version 0.0 of XML DTD for Coordinate Reference System
definition. This DTD uses XML elements that are specified in
other DTDs. -->

<!-- Coordinate Reference System: Definition of one Coordinate
Reference System (CRS), including the system identification and
possibly its complete specification. This coordinate reference
systemCRS can defines a coordinate space having any number of
dimensions, from one to four or more dimensions. The coordinate
space axes can include from one to three spatial axes.

The Coordinate Reference System data contains:

    Name Set: Set of one or more human understandable names for
    this Coordinate Reference System. This Name Set is optional,
    but usually should be included as discussed for the Name Set
    element.

    Identifier: Unique identifier used by software for this
    Coordinate Reference System. Either this Identifier or the
```

following Coordinate Reference System Specification must be included. Both can be included if useful, as discussed for the Identifier element.

Coordinate Reference System Specification: Detailed specification of this coordinate reference system. This specification data contains:

Coordinate System Definition: Definition of the set of coordinate axes used to record point coordinates based on this Coordinate Reference System.

Validity Region: Description of the region of validity of this coordinate system. This data is optional, included whenever the region of validity is known and can be usefully represented.

Coordinate Reference System Type Specification: Specification of one of a set of alternative types of coordinate reference systems.

The current ~~alternative~~ coordinate reference system types are:

Compound ~~Coordinate System~~CRS: Specification of a coordinate reference system that combines two or more simpler coordinate reference systems.

Geocentric ~~Cartesian Coordinate System~~CRS: Specification of a 3D Cartesian coordinate reference system with its origin at the (approximate) center of the Earth.

Geographic 3D ~~Coordinate System~~CRS: Specification of a 3D coordinate reference system based on Latitude and Longitude position around an ellipsoid that approximates the shape of the Earth. The third dimension is height above or below the ellipsoid surface.

Geographic 2D ~~Coordinate System~~CRS: Specification of a 2D coordinate reference system based on Latitude and Longitude position around an ellipsoid that approximates the shape of the Earth.

Projected ~~Coordinate System~~CRS: Specification of a 2D ~~Cartesian~~ map coordinate reference system, derived from a Geographic 2D ~~Coordinate System~~CRS by applying a “map projection” parameterized coordinate transformation.

Local ~~Coordinate System~~CRS: Specification of a coordinate reference system that is defined for and usually used in a limited region. The origin of a Local ~~Coordinate System~~CRS might or might not have a specified (or known) position in geodetic coordinates.

Vertical ~~Coordinate System~~CRS: Specification of a 1D coordinate reference system used for elevation, depth, or height measurements.

~~Note: Software implementers may notice that the coordinate reference system types listed above can be grouped by similarity of the specification data used. For example:~~

~~Several types of "Geodetic Coordinate System" are based on a Geodetic Datum, including:~~
~~Geocentric Cartesian Coordinate System~~
~~Geographic 3D Coordinate System~~
~~Geographic 2D Coordinate System~~

~~Several types of coordinate systems are always or optionally defined by a specified coordinate transformation from (or to) a base coordinate reference system, including:~~
~~Projected Coordinate System~~
~~Local Coordinate System~~

```
-->
<!ELEMENT CoordinateReferenceSystem (
        NameSet?,
        (Identifier
        | (CoordinateSystemDefinition,
          ValidityRegion?,
          (  Compound~~CoordinateSystem~~ CRS
          |  ~~GeocentricCartesian~~Geocentric~~CoordinateSystem~~ CRS
          |  Geographic3d~~CoordinateSystem~~ CRS
          |  Geographic2d~~CoordinateSystem~~ CRS
          |  Projected~~CoordinateSystem~~ CRS
          |  Local~~CoordinateSystem~~ CRS
          |  VerticalCRS~~CoordinateSystem~~),
          Identifier?) ) ) >

<!-- Coordinate System Definition: Definition of the set of
coordinate axes used to record point coordinates in a Coordinate
Reference System. This definition includes the name, direction,
unit, and sequence of each axis. The number of axes should equal
the number of dimensions in the Coordinate Reference System
space, and not contain duplicate or redundant axes.

Note that this Coordinate System Definition can specify a
different axis order and/or different axis units than normally
used by the associated type of Coordinate Reference System. Also,
this Coordinate System Definition can omit one or more of the
axes normally used by the associated type of Coordinate Reference
System.

The Coordinate System Definition data contains:
```

```
    Name Set: Set of one or more human understandable names for
    this set of coordinate axes. This Name Set is optional, but
    usually should be included as discussed for the Name Set
    element.

    Identifier: Unique identifier used by software for this set
    of coordinate axes. Either this Identifier or the following
    Coordinate Axes Specification must be included. Both can be
    included if useful, as discussed for the Identifier element.

    Coordinate Axes Specification: Specification of this set of
    Coordinate Axes. This specification data contains:

        Dimensions: Positive integer number of coordinate axes
        (or dimensions) in this coordinate reference system.

        List of Coordinate Axes: Ordered list of one or more
        definitions of the axes of this coordinate system. These
        Coordinate Axis descriptions are ordered to match the
        order of the axes values in point coordinates.
-->
<!ELEMENT CoordinateSystemDefinition (
        NameSet?,
        (Identifier
        | (dimensions,
          CoordinateAxis+,
          Identifier?) ) ) >
<!ELEMENT dimensions   (#PCDATA) >

<!-- Coordinate Axis: Definition of one axis or ordinate of a
coordinate system. The Coordinate Axis data contains:

    Axis Name: Human understandable text name of this axis or
    ordinate. This name can also be used by software, to match
    the same axis names. This name can be omitted if no useful
    axis name is defined. This axis name should be a standard,
    widely-used name whenever applicable. This axis name shall
    not specify or imply the units used; the units used shall be
    specified in the associated Unit element.

    Axis Abbreviation: Human understandable text abbreviated name
    of this axis or ordinate. If unique abbreviations are used,
    this abbreviation can also be used by software, to match the
    same axis names. This abbreviation can be omitted if no
    useful axis name abbreviation is defined.

    Axis Direction: Text specifying the direction of more-
    positive values along this axis or ordinate. This text should
    be one of a standard set of direction names whenever
    applicable, but other values can be used when needed. The
    currently defined axis direction names are:
```

~~North: Toward the North pole, precisely or approximately.~~

~~South: Toward the South pole, precisely or approximately.~~

~~East: Toward the East, precisely or approximately.~~

~~West: Toward the West, precisely or approximately.~~

North: Toward the North Pole, along the surface of the ellipsoid or the map plane, precisely or approximately.

East: Toward the East, along the surface of the ellipsoid or the map plane, precisely or approximately.

South: Toward the South Pole, along the surface of the ellipsoid or the map plane, precisely or approximately.

West: Toward the West, along the surface of the ellipsoid or the map plane, precisely or approximately.

Prime Meridian: Toward the Prime Meridian, to be used for the X axis of a ~~geocentric~~ Geocentric ~~Cartesian Coordinate System~~CRS.

Up and Down: Directions to be used for a vertical coordinate. While the north, south, east, and west directions are intended to be close to the geodetic directions, the up and down directions can be more flexible. An axis that begins obliquely downward can be said to be positive “down” (or positive “up”, depending on how the axis is defined). This is useful, for example, with wells. Depths of a well are often measured along the wellbore path. Since the wellbore path can become highly oblique – even horizontal, and at times, decreasing in true vertical depth – it is important that the concept of “down” not be restricted to increasing true vertical depth. In this case, the concept of “down” means that the measurement increases with increasing distance from the well reference point.

A more precise definition of the up and down directions that fits many cases is as follows. Consider the unit vector from the vertical reference point to the center of the earth. Consider also the unit tangent vector of the axis at the reference point, pointing in the direction of increasing values. If the dot product of these two vectors is positive, the direction is “down.” If negative, the direction is “up.” If zero, the axis is horizontal, and should not be considered to be a vertical axis.

Other: Any axis direction not described by one of the other standard Axis Direction names, and not described by using a non-standard Axis Direction name.

Unit: Physical unit used for ordinate values of this axis. This unit can be either a Linear or Angular unit. Note that the Unit types must be consistent with the type of the associated coordinate reference system. For example, a Geographical 2D ~~Coordinate System~~CRS uses only Angular Units, and a Geocentric ~~Cartesian Coordinate System~~CRS uses only Linear Units.
-->
```
<!ELEMENT CoordinateAxis (
        axisName?,
        axisAbbreviation?,
        axisDirection,
        (LinearUnit | AngularUnit) ) >
<!ELEMENT axisName           (#PCDATA) >
<!ELEMENT axisAbbreviation   (#PCDATA) >
<!ELEMENT axisDirection      (#PCDATA) >
```

<!-- Compound ~~Coordinate System~~CRS: Specification of a coordinate reference system that combines two or more simpler coordinate reference systems. None of the simpler coordinate reference systems can itself be compound. ~~(TBR)~~ In general, a compound coordinate system could contain any number of axes.
~~One or more of the axes in the underlying coordinate systems may be omitted in the Compound Coordinate System. In that case, the association between the coordinates in the Compound Coordinate System and the coordinates in the underlying coordinate systems is specified by using the same Coordinate Axis names for corresponding coordinate axes.~~

For spatial coordinates, a number of constraints exist for the construction of a C~~c~~ompound ~~coordinate system~~CRS~~s~~. For example, the coordinate reference systems that are combined should not contain any duplicate or redundant axes. Valid combinations include:

Geographic 2D + Vertical
Geographic 2D + Local 1D (near vertical)
Projected + Vertical
Projected + Local 1D (near vertical)
Local (horizontal 2D or 1D linear) + Vertical

The Compound ~~Coordinate System~~CRS data contains:

List of Coordinate Reference Systems: Ordered list of the two or more individual coordinate reference systems that are combined in this Compound ~~Coordinate System~~CRS.
-->
```
<!ELEMENT Compound~~CoordinateSystem~~CRS (
        CoordinateReferenceSystem+) >
```

<!-- Geocentric ~~Cartesian Coordinate System~~CRS: Specification of a 3D ~~Cartesian~~ coordinate reference system with its origin at an approximation to the center of the Earth. This can only be a Cartesian coordinate system, for which the axes are commonly named Geocentric X, Geocentric Y and Geocentric Z. The X and Y axes lie in the equatorial plane, with the positive X axis intersecting the prime meridian. The positive Y-axis direction is obtained by rotating the positive X-axis by 90 degrees counter-clockwise when viewed from the northern hemisphere. The Z axis is perpendicular to the equatorial plane, with its positive half intersecting the ellipsoid's North Pole and thus completing a right-handed 3D Cartesian coordinate system. Each associated Coordinate Axis must have a Linear Unit.

The Geocentric Cartesian ~~Coordinate System~~CRS data contains:

Geodetic Datum: The geodetic datum that specifies ~~where~~ the ~~center~~ origin and orientation of ~~the Earth is considered to be. In a~~this Geocentric ~~Coordinate System~~CRS~~, all coordinate points are measured from the center of the Earth (not along or from the ellipsoidal earth surface)~~.
-->
<!ELEMENT ~~GeocentricCartesian~~GeocentricCRS ~~CoordinateSystem~~ (
GeodeticDatum) >

<!-- Geographic 3D ~~Coordinate System~~CRS: Specification of a 3D coordinate reference system that allows positions to be specified relative to the surface of a reference ellipsoid~~based on an ellipsoid that approximates the shape of the Earth~~. Positions are ~~defined~~ specified ~~relative to the surface of this ellipsoid~~ by ~~means of two angles, L~~latitude and Longitude angles, and by the Height above or below the ellipsoid surface. The units of Latitude and Longitude must be Angular Units, and the Height units must be Linear Units. ~~(Note that a 3D geographic coordinate system using heights referenced to the ellipsoid can alternatively be handled as a Compound Coordinate System (TBR).)~~

Note: In the current associated OpenGIS Implementation Specification: Coordinate Transformation Services (OGC document 01-009) a Geographic 3D CRS is modeled as a Compound CRS, composed of a Geographic 2D CRS and a Vertical CRS, of which the latter is based on an Ellipsoidal Height Datum. This solution is now actively discouraged; the current understanding is that ellipsoidal heights cannot exist on their own, but only as an inseparable part of 3D coordinate tuples. This will be rectified in the next revision of the Implementation Specification.

~~Note: Some geographic coordinate systems use Latitude and Longitude, and some use Longitude and Latitude. You can find out which this system is by checking the list of Coordinate Axis descriptions. You should also check the Angular Unit, since not all geographic coordinate systems use degrees.~~

The Geographic 3D ~~Coordinate System~~CRS data contains:

Geodetic Datum: The geodetic datum that specifies the origin, orientation, and scale of this Geographic 3D ~~Coordinate System~~CRS.

-->
<!ELEMENT Geographic3d~~CoordinateSystem~~ CRS (
GeodeticDatum) >

<!-- Geographic 2D ~~Coordinate System~~CRS: Specification of a 2D coordinate reference system that allows positions to be specified on the surface of a reference ellipsoid~~based on an ellipsoid that approximates the shape of the Earth~~. Positions are ~~defined~~ specified ~~on the surface of this ellipsoid~~ by ~~means of two angular measures:~~ Latitude and Longitude angles. The Unit of each associated Coordinate Axis must be an Angular Unit.~~ (Note that a 3D geographic coordinate system using elevations or depths referenced to the geoid must be handled as a Compound Coordinate System, and a 3D geographic coordinate system using heights referenced to the ellipsoid can handled as a Compound Coordinate System (TBR).)~~

~~Note: Some geographic coordinate systems use Latitude and Longitude, and some use Longitude and Latitude. You can find out which this system is by checking the list of Coordinate Axis descriptions. You should also check the Angular Unit, since not all geographic coordinate systems use degrees.~~

The Geographic 2D ~~Coordinate System~~CRS data contains:

Geodetic Datum: The geodetic (or horizontal) datum that specifies the origin, orientation, and scale of this Geographic 2D ~~Coordinate System~~CRS.

-->
<!ELEMENT Geographic2d~~CoordinateSystem~~ CRS (
GeodeticDatum) >

<!-- Projected ~~Coordinate System~~CRS: Specification of a 2D ~~Cartesian~~ map coordinate reference system, derived from a Geographic 2D ~~Coordinate System~~CRS by applying a “map projection” coordinate transformation (usually termed a “conversion”). This coordinate transformation is applied to coordinates in the Geographic 2D CRS to produce the map coordinates in the Projected CRS. The Geodetic Datum of the Coordinate Reference system is not affected by this coordinate transformation. The Unit of each associated Coordinate Axis must be a Linear Unit.

The Projected ~~Coordinate System~~CRS data contains:

Coordinate Reference System: The base Geographic 2D ~~Coordinate System~~CRS ~~for~~ upon which this Projected ~~Coordinate System~~CRS is based.

Coordinate Transformation Definition: The Parameterized Transformation that ~~"projects" the Latitude and Longitude of the base Geographic 2D Coordinate System onto a planar surface.~~
defines this Projected CRS. This coordinate transformation is a map projection.
-->
<!ELEMENT Projected~~CoordinateSystem~~ CRS (
CoordinateReferenceSystem,
CoordinateTransformationDefinition) >

<!-- Local ~~Coordinate System~~CRS: Specification of a local coordinate reference system that is defined for and usually used in a limited region, significantly less than the complete surface of the earth. Examples are local engineering or architectural coordinates, grids, and drawings. A spatial Local ~~Coordinate System~~CRS usually has two characteristics:

Does not account for the curvature of the earth's surface. (A Local ~~Coordinate System~~CRS is not ~~a~~ geocentric ~~Cartesian coordinate system, although it may use Cartesian axes,~~ and ~~is~~ not a map P~~p~~rojected ~~coordinate system~~CRS.)

The datum is within or on the edge of the region in which the Local ~~Coordinate System~~CRS is expected to be used. (The origin is neither the center of the earth ellipsoid nor the point with zero Latitude and Longitude in any common geographic ~~coordinate system~~CRS.)

In this specification, the axes of a spatial Local Coordinate System are always Cartesian. A local coordinate system have any number of dimensions, from 1 to 3.

The datum of a Local ~~Coordinate System~~CRS might ~~or might not~~ have a specified (or known) position in a geodetic ~~coordinates~~CRS. Similarly, the coordinate axes might ~~or might not~~ have specified directions in a geodetic ~~coordinates~~CRS. A~~Such~~ geodetic ~~coordinates~~ CRS includes G~~g~~eographic ~~coordinates~~CRS, G~~g~~eocentric ~~coordinates~~CRS, and map P~~p~~rojected ~~coordinates~~CRS.

If the origin position and axes directions are specified in any geodetic coordinate system, point positions in a L~~l~~ocal ~~coordinate system~~CRS can be transformed to and from other coordinate reference systems. One common use of such a georeferenced L~~l~~ocal ~~coordinate system~~CRS is a grid coordinate system that defines the positions of a rectangular grid of points located in a base coordinate reference system. Such a grid coordinate system uses the grid origin for the axis origin, and uses the grid spacing for the axis units. Such a grid of points is used in a grid coverage and for other purposes. Such a grid coordinate system can be

specified by an affine transformation from the base coordinate reference system. ~~(Note that such a grid coordinate system can be derived from geographic or map projected coordinates, in which case it does account for the curvature of the earth's surface.)~~

In general, if the origin position and/or axes directions are not specified in any geodetic coordinate system, point positions in a local coordinate system cannot be transformed to other coordinate systems. However, if two L~~l~~ocal ~~coordinate systems~~CRSs have the same number of dimensions, axes names, axes units, and ~~local~~ Local ~~datum~~Datum, then software is permitted to assume that these two L~~l~~ocal ~~coordinate systems~~CRSs are identical. This allows multiple datasets from a common source (e.g. a CAD system) to be overlaid. In addition, some implementations of ~~a~~ Coordinate Transformation software may have a mechanism for correlating Local Datums. (e.g. from a database of transformations, which is created and maintained from measurements.)

The Local ~~Coordinate System~~CRS data contains either or both:

Local Datum: The datum that specifies the reference used to measure positions in this local or engineering coordinate system.

Coordinate system derivation: Data that relates this Local ~~Coordinate System~~CRS to another coordinate reference system. This data is included only when this Local ~~Coordinate System~~CRS is ~~geo~~referenced to another coordinate reference system. This data can be repeated for a georeferenced L~~l~~ocal ~~coordinate system~~CRS, to represent alternative ~~derivations~~references. This derivation data contains both:

Coordinate Reference System: The base Coordinate Reference System for this georeferenced Local ~~Coordinate System~~CRS. This base coordinate reference system can be of any type, including another Local ~~Coordinate System~~CRS.

Coordinate Transformation Definition: The Parameterized Transformation that specifies the transformation from the base coordinate reference system. ~~This transformation is usually an affine transformation.~~

-->

```
<!ELEMENT Local~~CoordinateSystem~~ CRS (
        LocalDatum
        | ( (CoordinateReferenceSystem,
             CoordinateTransformationDefinition)+,
          LocalDatum?) ) >
```

```
<!-- Vertical Coordinate SystemCRS: Specification of a 1D
coordinate reference system used for elevation, depth, or height
measurements. The Unit of the one associated Coordinate Axis must
be a Linear Unit.

The Vertical Coordinate SystemCRS data contains:

    Vertical Datum: The datum that specifies the reference used
    to measure vertical heights or depths. This is generally
    often a named datum that is specified by a geodetic
    authority. However, "local vertical datums" also exist, but
    are likely to be specified as part of a Local Datum., and can
    be either a geoidal or ellipsoidal datum.
-->
<!ELEMENT VerticalCoordinateSystem CRS (
        VerticalDatum) >

<!-- End of XML DTD for Coordinate Reference Systems -->


<!-- Version 0.0 of XML DTD for Datum definitions. This DTD uses
XML elements that are specified in other DTDs. -->

<!-- Vertical Datum: The reference used to measure elevations,
depths, or heights in a vertical coordinate system.

The Vertical Datum data contains:

    Name Set: Set of one or more human understandable names for
    this Vertical Datum. This Name Set is optional, but usually
    should be included as discussed for the Name Set element.

    Identifier: Unique identifier used by software for this
    Vertical Datum. Either this Identifier or the following
    vertical Datum Type must be included. Both can be included if
    useful, as discussed for the Identifier element.

    Datum Type: Text specifying the subtype of this vertical
    datum. The currently defined vertical datum subtypes and
    their meanings are:

        Geoidal: The zero point of the vertical axis is defined
        by a an approximately constant potential surface, usually
        – the geoid. A geoid surface is usually a surface
        determined specified by a national authority, and is
        related to sea level. When such is the case, it, and is
        then is a well known, named datum. For a geoidal datum,
        the vertical axis is measured in a distance unit from the
        geoid surface, and is usually named "elevation" or
        "depth".
```

Ellipsoidal: The ellipsoid surface defines the zero point of the datum. This datum type only makes sense if ~~this~~ the associated V~~v~~ertical ~~coordinate system~~CRS is combined with a 2D horizontal ~~coordinate system~~CRS. The ellipsoid is the same ellipsoid as for the horizontal ~~coordinate system~~CRS. For an ellipsoidal datum, the vertical axis is measured in a distance unit from the ellipsoid surface, and is usually named 'height". ~~(TBR)~~

Depth: The zero point of the vertical axis is defined by a surface that has meaning for the purpose which the associated vertical measurements are used for. For hydrographic charts this is often a predicted nominal sea surface (i.e. without waves or other wind and current effects) that occurs at low tide. Examples are Lowest Astronomical Tide and Lowest Low Water Spring. A different example is a sloping and undulating River Datum defined as the nominal river water surface occurring at a quantified river discharge.

Barometric: The zero point of the vertical axis is defined by a surface of equal atmospheric pressure. Height is then measured by barometer. This technique is routinely used in aircraft. The zero level is chosen as the atmospheric pressure on the ground at the time of take-off and the measured height value is commonly named "altitude", interpreted as vertical distance above the reference ground level.

-->

```
<!ELEMENT VerticalDatum (
        NameSet?,
        (Identifier
        | (datumType,
           Identifier?) ) ) >
<!ELEMENT datumType   (#PCDATA) >
```

<!-- Local Datum: The reference used to measure positions in a local or engineering coordinate system. See the description of a Local ~~Coordinate System~~CRS for more information.

The Local Datum data contains:

Name Set: Set of one or more human understandable names for this Local Datum. This Name Set is optional, but usually should be included as discussed for the Name Set element.

Identifier: Unique identifier used by software for this Local Datum. Either this Identifier or the following local Datum Type must be included. Both can be included if useful, as discussed for the Identifier element.

```
    Datum Type: Text describing this local datum. This text
    should describe or name the position of the coordinate system
    origin and the directions of the coordinate axes.
-->
<!ELEMENT LocalDatum (
        NameSet?,
        (Identifier
        | (datumType,
           Identifier?) ) ) >

<!-- Geodetic Datum: The reference used to measure positions in a
geographic or geocentric coordinate reference system, where the
shape of the earth is approximated by an ellipsoid.

The Geodetic Datum data contains:

    Name Set: Set of one or more human understandable names for
    this Geodetic Datum. This Name Set is optional, but usually
    should be included as discussed for the Name Set element.

    Identifier: Unique identifier used by software for this
    Geodetic Datum. Either this Identifier or the following
    Geodetic Datum Specification must be included. Both can be
    included if useful, as discussed for the Identifier element.

    Geodetic Datum Specification: Detailed specification of this
    geodetic datum. The Geodetic Datum Specification data
    contains:

        Ellipsoid: The approximation of the Earth's surface as a
        squashed sphere, used by this Geodetic Datum.

        Prime Meridian: Meridian toward which the X axis points.
        This data is optional, included only when the prime
        meridian is not the Greenwich meridian.
-->
<!ELEMENT GeodeticDatum (
        NameSet?,
        (Identifier
        | (Ellipsoid,
           PrimeMeridian?,
           Identifier?) ) ) >

<!-- End of XML DTD for Datum definitions -->


<!-- Version 0.0 of XML DTD for Ellipsoid and Prime Meridian
definitions. This DTD uses XML elements that are specified in
other DTDs. -->
```

<!-- Ellipsoid: Definition of an ellipsoid that approximates the shape of the Earth.

The Ellipsoid data contains:

Name Set: Set of one or more human understandable names for this Ellipsoid. This Name Set is optional, but usually should be included as discussed for the Name Set element.

Identifier: Unique identifier used by software for this Ellipsoid. Either this Identifier or the following Ellipsoid Specification must be included. Both can be included if useful, as discussed for the Identifier element.

Ellipsoid Specification: Detailed specification of this ellipsoid. The Ellipsoid Specification data contains:

Linear Unit: Definition of the linear unit used for the Semi-Major Axis and Semi-Minor Axis values of this ellipsoid.

Semi-Major Axis: The numerical value of the equatorial radius of the ellipsoid, expressed in the related Linear Unit.

Semi-Minor Axis: The numerical value of the distance from the center of the ellipsoid to either of its poles, expressed in the related Linear Unit.

Inverse Flattening: The numerical value of the inverse flattening constant for this ellipsoid. The inverse flattening of a spheroid is related to the semi-major (a) and semi-minor (b) axes by the formula 1/f = a/(a-b). For perfect spheres, this formula breaks down, and the special inverse flattening value of zero is used.

Flattening Definitive: The Boolean value that indicates if the Inverse Flattening value is definitive for this ellipsoid. Some ellipsoids use the inverse flattening as the defining value, and calculate the polar radius when needed. Other ellipsoids use the polar radius to calculate the Inverse Flattening value when needed. This distinction can be important to avoid floating point rounding errors.

Ellipsoid Shape: The Boolean value that indicates if the reference surface is an ellipsoid. True means the reference surface is am ellipsoid, and False means it is a sphere. This data is optional, included when desired. When not included, the reference surface is a sphere when the values of the Semi-Major Axis and Semi-Minor Axis are equal

```
-->
<!ELEMENT Ellipsoid (
        NameSet?,
        (Identifier
        | (LinearUnit,
          semiMajorAxis,
          semiMinorAxis,
          inverseFlattening,
          Identifier?) ) ) >
    <!ATTLIST Ellipsoid
        flatteningDefinitive   (true | false)   #REQUIRED
        ellipsoidShape         (true | false)   #IMPLIED >
<!ELEMENT semiMajorAxis       (#PCDATA) >
<!ELEMENT semiMinorAxis       (#PCDATA) >
<!ELEMENT inverseFlattening   (#PCDATA) >

<!-- Prime Meridian: The meridian used as the reference for
Longitude measurements.

The Prime Meridian data contains:

    Name Set: Set of one or more human understandable names for
    this Prime Meridian. This Name Set is optional, but usually
    should be included as discussed for the Name Set element.

    Identifier: Unique identifier used by software for this Prime
    Meridian. Either this Identifier or the following Prime
    Meridian Specification must be included. Both can be included
    if useful, as discussed for the Identifier element.

    Prime Meridian Specification: Detailed specification of this
    prime meridian. The Prime Meridian Specification data
    contains:

        Greenwich Longitude: The Longitude of this Prime Meridian
        relative to the Greenwich Meridian, expressed in the
        related Angular Unit.

        Angular Unit: Definition of angular unit for the related
        Greenwich Longitude value.
-->
<!ELEMENT PrimeMeridian (
        NameSet?,
        (Identifier
        | (greenwichLongitude,
          AngularUnit,
          Identifier?) ) ) >
<!ELEMENT greenwichLongitude   (#PCDATA) >

<!-- End of XML DTD for Ellipsoid and Prime Meridian -->
```

1F.4XML DTDs for shared elements

This subclause presents the commented XML DTDs containing elements used by the XML DTDs for both Coordinate Reference System definition and Coordinate Transformation Definition.

```
<!-- Version 0.0 of XML DTD for Identification information. -->

<!-- Identifier: Unique identifier used by software for a
specific coordinate reference system, coordinate transformation,
or other set of data. This identifier does not contain
specification information, but contains the minimum information
needed to allow finding specification data stored elsewhere.

The Identifier data contains:

    Code: Character string code, name, or other unique identifier
    of the related set of data. The meaning of this Code is
    defined by the related Code Space, and the range and format
    of these codes are defined by the Code Space authority. This
    code is usually not human understandable, but it could be.

    Code Space: Character string identifier of a code space
    within which one or more Codes are defined. This code space
    is often defined by some authority organization, where one
    organization may define multiple code spaces. The range and
    format of each Code Space identifier is defined by that code
    space authority. When the Code Space identifier is long, an
    all-capitals abbreviation of that identifier may be used
    (such as "EPSG" for European Petroleum Survey Group).

    Edition: Character string identifier of one edition or
    version of the related Code Space or "Code". This edition
    identifier applies to either the Code Space or the "Code" as
    defined by the Code Space authority. This data is optional,
    included whenever multiple editions exist or are expected to
    exist. When appropriate, the edition is identified by the
    effective date, coded using ISO 19108 date format.

Note 1: One authority (code space and edition) can define more
than one set of unique codes for identifying one item from a set
of alternatives. For example, the EPSG currently defines three
different sets of unique codes for identifying a Length Unit and
for an Angular Unit. When an authority defines more than one set
of unique codes, any of these sets of codes could be used.

Note 2: Whenever an Identifier element can be included in another
element, certain “specification” data can alternately be included
in the same other element. Either the Identifier element or the
```

```
specification data element(s) must be included, allowing a
“definition” XML element to be used in two ways:

    Only the Identifier can be included for a geospatial entity
    that is “well-known” to the XML receiver. Well-known means
    that the corresponding "specification" data can be found
    elsewhere when needed, using only this Identifier. This
    “elsewhere” is never in the same XML file that contains this
    Identifier-only definition element, is usually not in any
    public XML file, and is never in a XML file whose name or URI
    is known to the XML sender. (TBR)

    Only the "specification" data can be included for a custom
    geospatial entity that is not uniquely or permanently
    identified.

Both the "identifier" data and "specification" data can be
included in one XML element if useful, but this should generally
not be done. Specifically, the "specification" data should not be
included with an Identifier for any geospatial entity that is
“well-known” to the XML receiver, since these two forms of
information are redundant and may conflict. If both are included,
the included "specification" data should be used by the XML
receiver, instead of the included Identifier.
-->
<!ELEMENT Identifier (
        code,
        codeSpace,
        edition? ) >
<!ELEMENT code           (#PCDATA) >
<!ELEMENT codeSpace      (#PCDATA) >
<!ELEMENT edition        (#PCDATA) >

<!-- Name Set: Set of human understandable names for a specific
coordinate reference system, coordinate transformation, or
related set of data. These names usually do not uniquely identify
the set of data (or metadata), but provide human understandable
names for that data. The Name Set data contains at least one of:

    Name: The primary human understandable name for this set of
    data. This data is optional, included only when a human
    readable name is defined and useful.

    Abbreviation: An abbreviated human understandable name for
    this set of data. This data is optional, included only when
    an abbreviation is defined and useful. More than one
    abbreviation can be included if useful.

    Alias: An alternate human understandable name for this set of
    data. This data is optional, included only when an alias is
    defined and useful. More than one alias can be included if
    useful.
```

```
    Remarks: Human understandable remarks or comments about this
    coordinate reference system, coordinate transformation, or
    related set of data. More than one remark can be included if
    useful.

Note: Whenever a Name Set element can be included in another
element, the Identifier element and/or “specification” data can
also be included in the same other element. Whatever other data
is included in an element, the Name Set element should be
included if useful, and it will often be useful to a human user.
However, the Name Set element may not be useful when the included
Identifier element or "specification" data is sufficiently human
understandable.
-->
<!ELEMENT NameSet (
        name,
        abbreviation*,
        alias*,
        remarks*) >
<!ELEMENT name           (#PCDATA) >
<!ELEMENT abbreviation   (#PCDATA) >
<!ELEMENT alias          (#PCDATA) >
<!ELEMENT remarks        (#PCDATA) >

<!-- End of XML DTD for Identification information -->


<!-- Version 0.0 of XML DTD for Unit definitions. -->

<!-- Linear Unit: Definition of a linear unit of measure.

The Linear Unit data contains:

    Name Set: Set of one or more human understandable names for
    this Linear Unit. This Name Set is optional, but usually
    should be included as discussed for the Name Set element.

    Identifier: Unique identifier used by software for this
    Linear Unit. Either this Identifier or the following Linear
    Unit Specification must be included. Both can be included if
    useful, as discussed for the Identifier element.

    Linear Unit Specification: Specification of this linear unit
    relative to the international standard meter. The Linear Unit
    Specification data contains either or both:

        Meters Per Unit: Number of meters per linear unit.

        Units Per Meter: Number of linear units per meter.
```

```
        Note: If both of these scale factors are included, the
        proper scale factor is the ratio of these individual
        scale factors.
-->
<!ELEMENT LinearUnit (
        NameSet?,
        (Identifier
        | ( ( (metersPerUnit, unitsPerMeter?)
            |  unitsPerMeter),
           Identifier?) ) ) >
<!ELEMENT metersPerUnit   (#PCDATA) >
<!ELEMENT unitsPerMeter   (#PCDATA) >

<!-- Angular Unit: Definition of an angular unit of measure.

The Angular Unit data contains:

    Name Set: Set of one or more human understandable names for
    this Angular Unit. This Name Set is optional, but usually
    should be included as discussed for the Name Set element.

    Identifier: Unique identifier used by software for this
    Angular Unit. Either this Identifier or the following Angular
    Unit Specification must be included. Both can be included if
    useful, as discussed for the Identifier element.

    Angular Unit Specification: Specification of this angular
    unit relative to radians. The Angular Unit Specification data
    contains either or both:

        Radians Per Unit: Number of radians per angular unit.

        Units Per Radian: Number of angular units per radian.

        Note: If both of these scale factors are included, the
        proper scale factor is the ratio of these individual
        scale factors.
-->
<!ELEMENT AngularUnit (
        NameSet?,
        (Identifier
        | ( ( (radiansPerUnit, unitsPerRadian?)
            |  unitsPerRadian),
           Identifier?) ) ) >
<!ELEMENT radiansPerUnit   (#PCDATA) >
<!ELEMENT unitsPerRadian   (#PCDATA) >

<!-- End of XML DTD for Unit definitions -->
```

```
<!-- Version 0.0 of XML DTD for Validity Region definition. -->

<!-- Validity Region: Description of the region of validity of a
coordinate reference system, coordinate transformation, or
related set of data. For a coordinate transformation, this region
of validity applies to the source coordinate reference system.
The Validity Region data contains either or both:

    Validity Area: Human-understandable text defining the region
    of validity.

    Validity Envelope: Specification of the region of validity as
    a rectangle in two dimensions, or a rectangular volume in
    three dimensions. The Validity Envelope contains:

        Minimum Coordinates: Ordered list of numbers, one for
        each coordinate system axis, that specify the minimum
        coordinates of this region of validity. Each number is
        normally the algebraically minimum axis value over the
        region of validity. However, a minimum Longitude is the
        West-most value over the region of validity.

        Maximum Coordinates: Ordered list of numbers, one for
        each coordinate system axis, that specify the maximum
        coordinates of this region of validity. Each number is
        normally the algebraically maximum axis value over the
        region of validity. However, a maximum Longitude is the
        East-most value over the region of validity.

        Notes: The numbers in the Minimum and Maximum Coordinates
        lists are separated by spaces, and can be either real
        (floating point) or integer numbers. A point with these
        Minimum Coordinates or Maximum Coordinates may be outside
        the valid domain of the related coordinate reference
        system or coordinate transformation.

        Validity Type: Character string code indicating the
        precision or meaning of this envelope of validity. For a
        coordinate system, the possible Validity Type values and
        meanings are:

            Exact: Any point within this envelope is valid, and
            no point outside this envelope is valid.

            Typical: Most points within this envelope are valid,
            and most points outside this envelope are not valid.

            Minimum: Any point within this envelope is valid, and
            some points outside this envelope are also valid.
```

```
            Maximum: Most points inside this envelope are valid,
            and no point outside this envelope is valid.

            Unlimited: The region of validity is limited only by
            the valid ranges of all the individual axes values.
            In this case, the minimum and maximum envelope values
            contain the axes minimum and maximum values. Any
            point with all axes values within that axis value
            range are valid.

        For a coordinate transformation, the possible values and
        meanings of the Validity Type are:

            Exact: Any source coordinate system point within this
            envelope can be accurately transformed, and no point
            outside this envelope can be accurately transformed.

            Typical: Most source coordinate system points within
            this envelope can be accurately transformed, and most
            points outside this envelope can not be accurately
            transformed.

            Minimum: Any source coordinate system point within
            this envelope can be accurately transformed, and some
            points outside this envelope will also be accurately
            transformed.

            Maximum: Most source coordinate system points within
            this envelope can be accurately transformed, but no
            points outside this envelope will be accurately
            transformed.

            Unlimited: The envelope of validity is limited only
            by the valid ranges of all the individual axes values
            for the source coordinates. Any source coordinate
            system point with all axes values within that axis
            value range can be accurately transformed.
-->
<!ELEMENT ValidityRegion (
        validityArea
        | (minimumCoordinates,
           maximumCoordinates,
           validityArea?) ) >
    <!ATTLIST ValidityRegion
        validityType   (exact | typical | minimum | maximum
            | unlimited)          #IMPLIED >
<!ELEMENT   validityArea         (#PCDATA) >
<!ELEMENT   minimumCoordinates   (#PCDATA) >
<!ELEMENT   maximumCoordinates   (#PCDATA) >

<!-- End of XML DTD for Validity Region definition -->
```

Annex G
(informative)

XML overviews and examples

1G.1 Introduction

This annex provides overviews and examples of the XML specified in Annex F. Since the XML structure is not easy to see in commented XML DTDs, this section includes the above XML DTDs with all comments eliminated.

This annex discusses the specified XML in four parts, used for representation of:

a) Shared Elements, used by all below

b) Coordinate Reference System definition

c) Coordinate Transformation Definition

d) Parameter values, of some parameters used in Parameterized Transformations

The primary differences in the XML DTDs in Annex F, compared to the corresponding XML DTDs in the currently proposed OGC standard XML (OGC document 01-014), is omission of the XML elements not needed in this Implementation Specification. Specific omissions here include the:

a) Image ~~Coordinate System~~CRS (XML element) type of Coordinate Reference System, and the Image Datum and Pixel Spacing Unit (XML elements)

b) Temporal ~~Coordinate System~~CRS (XML element) type of Coordinate Reference System, and the Time Unit (XML element)

c) Position Error Estimates and Covariance Element (XML elements), included in the Transformation Definition (XML element)

d) Transformation Method Specification set of elements in Transformation Method (XML element), and the Parameter Definition (XML element)

~~e) Name Set and Identifier (XML elements) included in Coordinate System Definition (XML element)~~

~~f)~~e) Axis Number (XML element) included in Coordinate Axis (XML element)

1G.2 XML for shared elements

This subclause lists without the comments the XML DTDs specified in Subclause F.4 for shared XML elements.

```
<!ELEMENT Identifier (
        code,
        codeSpace,
        edition? ) >
<!ELEMENT code           (#PCDATA) >
<!ELEMENT codeSpace      (#PCDATA) >
<!ELEMENT edition        (#PCDATA) >

<!ELEMENT NameSet (
        name,
        abbreviation*,
        alias*,
        remarks*) >
<!ELEMENT name           (#PCDATA) >
<!ELEMENT abbreviation   (#PCDATA) >
<!ELEMENT alias          (#PCDATA) >
<!ELEMENT remarks        (#PCDATA) >


<!ELEMENT LinearUnit (
        NameSet?,
        (Identifier
        | ( ( (metersPerUnit, unitsPerMeter?)
            |  unitsPerMeter),
           Identifier?) ) ) >
<!ELEMENT metersPerUnit   (#PCDATA) >
<!ELEMENT unitsPerMeter   (#PCDATA) >

<!ELEMENT AngularUnit (
        NameSet?,
        (Identifier
        | ( ( (radiansPerUnit, unitsPerRadian?)
            |  unitsPerRadian),
           Identifier?) ) ) >
<!ELEMENT radiansPerUnit   (#PCDATA) >
<!ELEMENT unitsPerRadian   (#PCDATA) >


<!ELEMENT ValidityRegion (
        validityArea
        | (minimumCoordinates,
           maximumCoordinates,
           validityArea?) ) >
    <!ATTLIST ValidityRegion
        validityType   (exact | typical | minimum | maximum
            | unlimited)          #IMPLIED >
<!ELEMENT   validityArea         (#PCDATA) >
<!ELEMENT   minimumCoordinates   (#PCDATA) >
<!ELEMENT   maximumCoordinates   (#PCDATA) >
```

1G.3 XML for coordinate reference system definition

This subclause lists without the comments the XML DTDs specified in Subclause F.3 for Coordinate Reference System definition

```
<!ELEMENT CoordinateReferenceSystem (
        NameSet?,
        (Identifier
        | (CoordinateSystemDefinition,
          ValidityRegion?,
          (  CompoundCoordinateSystem_CRS
          |  GeocentricCartesianGeocentricCoordinateSystem_CRS
          |  Geographic3dCoordinateSystem_CRS
          |  Geographic2dCoordinateSystem_CRS
          |  ProjectedCoordinateSystem_CRS
          |  LocalCoordinateSystem_CRS
          |  VerticalCRSCoordinateSystem),
          Identifier?) ) ) >

<!ELEMENT CoordinateSystemDefinition (
        NameSet?,
        (Identifier
        | (dimensions,
          CoordinateAxis+,
          Identifier?) ) ) >
<!ELEMENT dimensions   (#PCDATA) >

<!ELEMENT CoordinateAxis (
        axisName?,
        axisAbbreviation?,
        axisDirection,
        (LinearUnit | AngularUnit) ) >
<!ELEMENT axisName           (#PCDATA) >
<!ELEMENT axisAbbreviation   (#PCDATA) >
<!ELEMENT axisDirection      (#PCDATA) >

<!ELEMENT CompoundCoordinateSystem_CRS (
        CoordinateReferenceSystem+) >

<!ELEMENT GeocentricCartesianGeocentricCoordinateSystem_CRS (
        GeodeticDatum) >

<!ELEMENT Geographic3dCoordinateSystem_CRS (
        GeodeticDatum) >

<!ELEMENT Geographic2dCoordinateSystem_CRS (
        GeodeticDatum) >

<!ELEMENT ProjectedCoordinateSystem_CRS (
        CoordinateReferenceSystem,
        CoordinateTransformationDefinition) >

<!ELEMENT LocalCoordinateSystem_CRS (
```

```
        LocalDatum
        | ( (CoordinateReferenceSystem,
             CoordinateTransformationDefinition)+,
          LocalDatum?) ) >

<!ELEMENT Vertical~~CoordinateSystem~~ CRS (
        VerticalDatum) >


<!ELEMENT VerticalDatum (
        NameSet?,
        (Identifier
        | (datumType,
           Identifier?) ) ) >
<!ELEMENT datumType   (#PCDATA) >

<!ELEMENT LocalDatum (
        NameSet?,
        (Identifier
        | (datumType,
           Identifier?) ) ) >

<!ELEMENT GeodeticDatum (
        NameSet?,
        (Identifier
        | (Ellipsoid,
           PrimeMeridian?,
           Identifier?) ) ) >


<!ELEMENT Ellipsoid (
        NameSet?,
        (Identifier
        | (LinearUnit,
          semiMajorAxis,
          semiMinorAxis,
          inverseFlattening,
          Identifier?) ) ) >
    <!ATTLIST Ellipsoid
        flatteningDefinitive   (true | false)   #REQUIRED
        ellipsoidShape         (true | false)   #IMPLIED >
<!ELEMENT semiMajorAxis       (#PCDATA) >
<!ELEMENT semiMinorAxis       (#PCDATA) >
<!ELEMENT inverseFlattening   (#PCDATA) >

<!ELEMENT PrimeMeridian (
        NameSet?,
        (Identifier
        | (greenwichLongitude,
          AngularUnit,
          Identifier?) ) ) >
<!ELEMENT greenwichLongitude   (#PCDATA) >
```

1G.4 XML for coordinate transformation definition

This subclause lists without the comments the XML DTDs specified in Subclause F.2 for Coordinate Transformation Definition.

```
<!ELEMENT CoordinateTransformationMetadata (
        CoordinateReferenceSystem,
        CoordinateReferenceSystem,
        CoordinateTransformationDefinition?) >

<!ELEMENT CoordinateTransformationDefinition (
        NameSet?,
        (Identifier
        | (sourceDimensions,
          targetDimensions,
          ValidityRegion?,
          ( ConcatenatedTransformation
          | InverseTransformation
          | PassThroughTransformation
          | ParameterizedTransformation),
          Identifier?) ) ) >
<!ELEMENT sourceDimensions   (#PCDATA) >
<!ELEMENT targetDimensions   (#PCDATA) >

<!ELEMENT ConcatenatedTransformation (
        CoordinateTransformationDefinition+) >

<!ELEMENT InverseTransformation (
        CoordinateTransformationDefinition) >

<!ELEMENT PassThroughTransformation (
        CoordinateTransformationDefinition,
        firstModifiedOrdinate,
        numberModifiedOrdinates) >
<!ELEMENT firstModifiedOrdinate     (#PCDATA) >
<!ELEMENT numberModifiedOrdinates   (#PCDATA) >


<!ELEMENT ParameterizedTransformation (
        TransformationMethod,
        Parameter*) >

<!ELEMENT Parameter (
        codeName,
        value,
        (LinearUnit | AngularUnit)? ) >
<!ELEMENT codeName   (#PCDATA) >
<!ELEMENT value      (#PCDATA) >

<!ELEMENT TransformationMethod (
```

```
        NameSet?,
        Identifier) >
```

~~1~~G.5 Example XML for transformation parameter values

This subclause provides example XML contents for the values of the parameters used in some Parameterized Transformations. This XML is specified using the DTD elements for "Parameterized Transformation" and for "Parameter" specified in Subclause F.4. (This XML can be used as part of the XML for a Coordinate Transformation Definition or Coordinate Transformation Metadata.)

The definitions of some Parameterized Transformations, with all the Parameter values used, are well-known and are clearly specified in other documents. For example, the EPSG database clearly specifies many Parameterized Transformations with the associated Parameter values. This subclause includes only a few representative examples for Parameterized Transformations clearly specified elsewhere.

The definitions of many Parameterized Transformations are not clearly specified in other OGC documents, and this subclause thus includes more extensive examples. In many cases, the specific parameter values will be specific to a particular application.

NOTE Some examples of XML use for Coordinate Reference System definitions, with the associated Shared Elements, are provided in Section 6 of OGC document 01-014.

~~1~~G.5.1 Transverse Mercator map projection transformation

This subclause provides example XML contents for the values of the parameters used for a Transverse Mercator map Projected ~~Coordinate System~~CRS . This map projection is from a Geographic 2D ~~Coordinate System~~CRS. The Parameterized Transformation for that map projection can be specified by the following XML:

```
<ParameterizedTransformation>
   <TransformationMethod>
      <NameSet>
         <name>Transverse_Mercator</name>
      </NameSet>
      <Identifier>
         <code>TBD</code>
         <codeSpace>EPSG</codeSpace>
      </Identifier>
   </TransformationMethod>
   <Parameter>
      <codeName>latitude_of_origin</codeName>
      <value>49</value>
      <AngularUnit>
         <NameSet>
            <name>DMSH</name>
         </NameSet>
         <radiansPerUnit>1.74532925199433E-02</radiansPerUnit>
         <Identifier>
            <code>9108</code>
```

```
            <codeSpace>EPSG</codeSpace>
         </Identifier>
      </AngularUnit>
   </Parameter>
   <Parameter>
      <codeName>central_meridian</codeName>
      <value>2</value>
      <AngularUnit>
         <NameSet>
            <name>DMSH</name>
         </NameSet>
         <radiansPerUnit>1.74532925199433E-02</radiansPerUnit>
         <Identifier>
            <code>9108</code>
            <codeSpace>EPSG</codeSpace>
         </Identifier>
      </AngularUnit>
   </Parameter>
   <Parameter>
      <codeName>scale_factor</codeName>
      <value>0.999601272</value>
   </Parameter>
   <Parameter>
      <codeName>false_easting</codeName>
      <value>400000</value>
      <LinearUnit>
         <NameSet>
            <name>metre</name>
            <abbreviation>m</abbreviation>
         </NameSet>
         <metersPerUnit>1</metersPerUnit>
         <Identifier>
            <code>9001</code>
            <codeSpace>EPSG</codeSpace>
         </Identifier>
      </LinearUnit>
   </Parameter>
   <Parameter>
      <codeName>false_northing</codeName>
      <value>-100000</value>
      <LinearUnit>
         <NameSet>
            <name>metre</name>
            <abbreviation>m</abbreviation>
         </NameSet>
         <metersPerUnit>1</metersPerUnit>
         <Identifier>
            <code>9001</code>
            <codeSpace>EPSG</codeSpace>
         </Identifier>
      </LinearUnit>
   </Parameter>
```

```
</ParameterizedTransformation>
```

Annex H
(informative)

Interface design decisions

H.1 Introduction

This annex summarizes some of the decisions made in designing these coordinate transformation interfaces. These design decisions are listed in groups depending on the service part affected:

a) Ground coordinate transformation service

b) XML for coordinate reference system definition

c) XML for coordinate transformation definition

Before discussing these design decisions, the design objectives and design approach used to guide these design decisions are summarized.

H.2 Design objectives

A very important objective in the design of this high-level coordinate transformation service interface was to minimize the difficulty of understanding and using this interface. Of primary concern was minimizing the difficulty of client use of these transformation services, since many more clients will be programmed than server implementations. (For example, perhaps 300 clients will be programmed, but only 3 server implementations.) This objective assumes this OGC standard interface is commercially successful.

Another very important design objective was to not significantly reduce the functionality from that of the current draft low-level CT interface (specified in OGC document ~~00-007r4~~01-009). Clients that need only reduced functionality should be able to be satisfied by a server that implements only a subset of the specified interface elements.

Lesser objectives include computation efficiency and ease of server implementation. Small reductions in efficiency and other server properties were considered acceptable in order to better achieve the primary objectives.

H.3 Design approach

In designing this easy-to-use interface, the basic approach followed was to minimize the total numbers of client-visible interface elements. These interface elements include all:

a) Interface objects (not counting interface classes)

b) Arguments of all operations, including operation outputs and inputs

c) Attributes of all <<DataType>> classes (where a <<DataType>> class has no operations)

d) Distinct values (or attributes) of all <<CodeList>> and <<Enumeration>> classes

e) All <<DataType>>, <<CodeList>>, and <<Enumeration>> classes (not counting objects of these classes)

An interface object is any UML object having one or more operations. An interface object (using the <<Interface>> UML stereotype) does not have any attributes, but uses operations to set and get the values of any data that might otherwise be considered attributes.

The types of interface elements are listed above most critical first. That is, one <<Interface>> object seems most difficult to understand, and one <<DataType>>, <<CodeList>>, or <<Enumeration>> class seems least difficult to understand (not including the contained attributes, which are counted separately). In addition, repeated use of the same argument or same attribute is preferred over using somewhat different arguments or attributes. The same argument or attribute means the same data type, same name, and same definition.

H.4 Ground coordinate transformation service

The decisions made in the process of designing the ground coordinate transformation service interface included:

a) Do not implement the concept of a “session” for a client. The alternative not selected was to implement a “session” for a client. Note that the “session” concept might be implemented by an external software layer for access control, or in a DCP-dependent profile of this general interface.

b) Include the capability for a client to obtain the “Transformation ID” value for the transformation between client-specified source and target coordinate systems. In one alternative, the client would be restricted to coordinate transformations with well-known values of the “Transformation ID”.

c) Include an input to the “addTransformation” operation that allows a client to provide the definition of a custom or special coordinate transformation, or to specify a coordinate transformation choice (when applicable), with optional implementation by server. Some of the alternatives not selected include:

    1) Not allow a client to provide the definition of a custom or special coordinate transformation.

    2) Not allow a client to specify a coordinate transformation choice (when applicable).

    3) Provide operation(s) separate from the “addTransformation” operation to allow a client to provide the definition of a custom or special coordinate transformation, and/or to specify a coordinate transformation choice (when applicable).

4) Require all compliant servers to implement this capability (for some set of coordinate transformations).

d) Use one "Transformation ID" input to the "transform" and "transformList" operations, where this input identifies the combination of source coordinate system, target coordinate system, and coordinate transformation. In one alternative not selected, separate inputs would be used to identify the source coordinate system, target coordinate system, and coordinate transformation.

e) Use a "Transformation ID" input to the "transform" and "transformList" operations whose contents are not interpretable by clients. That is, the Transformation ID is encoded as a Character String without any further public specification of the string contents. An alternative would be to provide clients with visibility into the contents of a Transformation ID string.

f) Allow the "Transformation Metadata" input to the "addTransformation" operation to contain only the identifications of the source and target coordinate reference systems, when those coordinate systems are well-known. One alternative not selected would be to require a client to always send to the server detailed definitions of all coordinate transformations.

g) Allow the contents of a CoordinateReferenceSystem definition to optionally contain one or more additional Coordinate Transformation Definitions, each to that CRS from another CRS. (That is, the contents of a CoordinateReferenceSystem definition include but are deliberately NOT limited to the information required to specify a CRS.) The purpose of allowing inclusion of such additional Coordinate Transformation Definition elements is to minimize the need to separately transfer a Coordinate Transformation Definition in Transformation Metadata.

~~g)~~h) Require each client to use the "addTransformation" operation to obtain each "Transformation ID", before using that Transformation ID in a "transform" or "transformList" operation. That is, a server implementation is not allowed to specify a set of Transformation ID values that can be considered well-known by clients of that service. One alternative would be to allow a server implementation to specify a set of Transformation ID values that can be considered well-known by clients of that service.

~~h)~~i) Allow and expect the server software to make this transformation active when the "addTransformation" operation is performed for a client. Some of the alternatives not selected include:

1) Require the server to make this transformation active when this operation is performed.

2) Not allow the server to make this transformation active when this operation is performed.

3) Provide separate operation(s) that allow a client to make active a well-known or previously specified coordinate transformation.

~~i)~~j) Require a client to use the "addTransformation" operation to convert "Transformation Metadata" to a "Transformation ID", before using the "transform" and "transformList" operations. One alternative would be to include separate operation(s) that perform coordinate transformation with the client directly supplying a value of Transformation Metadata. Another alternative would be to define the "transform" and "transformList" operations to accept input of either Transformation ID or Transformation Metadata.

~~j)~~k) Include the "transformationMetadata" operation to allow a client to retrieve the detailed definition of any coordinate transformation (including source and target coordinate systems), with optional implementation by server. Some of the alternatives not selected include:

1) Not allow clients to retrieve detailed definitions or specifications of coordinate transformations and coordinate systems.

2) Require implementation of this operation by all server implementations.

3) Specify multiple operations for accessing various parts of the transformation metadata, such as the individual source and target coordinate systems.

~~k)~~l) Include the "Text Format" input to the "transformationMetadata" operation to allow clients to specify the format desired for the "Transformation Metadata" output. One alternative would be to allow clients to retrieve transformation metadata in only one format (probably one XML format). Other alternatives would allow clients to retrieve transformation metadata in a binary or other non-text format.

~~l)~~m) Include the "removeTransformation" operation to allow a client to remove or make inactive the detailed definition of a (custom or special or well-known) coordinate transformation. One alternative would be to require server software to keep active all client specified or selected definitions of coordinate transformations until a client session is terminated. Another alternative would be to require server software to automatically remove (or make inactive) coordinate transformation definitions that were not recently used.

~~m)~~n) Allow a server implementation to limit the number of currently active coordinate transformations by automatically making non-active selected active transformation definitions that were not recently used. One alternative would be to not allow a server implementation to make non-active selected active transformation definitions that were not recently used.

~~n)~~o) Not allow a server implementation to limit the number of currently active coordinate transformations by rejecting additional coordinate transformations received from client(s) after some maximum number of active transformations is reached. One alternative would be to allow a server implementation to reject additional coordinate transformations received from client(s) after some maximum number of active transformations is reached.

p) Not include any operations, data types, and other interface elements to assemble and disassemble the definitions of coordinate reference systems and coordinate

transformations (“CoordinateReferenceSystem” and “TransformationDefinition”).) One alternative would be to provide such capabilities (and this is a possible future extension of this interface).

1 H.5 XML for coordinate reference system definition

The decisions made in adapting the currently proposed OGC standard XML for Coordinate Reference System definition included:

a) Not include temporal coordinate reference systems.

b) Not include polar, cylindrical, and spherical coordinate systems.

c) For some types of coordinate reference system, allow the contents of a Coordinate Reference System element to optionally contain one or more additional Coordinate Transformation Definition elements, each to that CRS from another CRS. (See item g) in Subclause H.4.)

1H.6 XML for coordinate transformation definition

The decisions made in adapting the currently proposed OGC standard XML for Transformation Definition included:

a) Not include Parameter Definitions in data transferred.

Annex I
(informative)

Possible future expansion

This annex briefly lists some of the capabilities that might be added in the future to this high-level coordinate transformations interface. Some of the server capabilities that might be added in the future include:

a) Determine position error estimates for transformed coordinates, combining error estimate data for coordinate transformations and input coordinates.

b) Obtain partial derivatives of target coordinates with respect to source coordinates.

c) Retrieve service metadata, by a client from a service implementation.

d) Handle additional types of coordinate reference systems, such as:

   1) Temporal coordinate systems.

   2) Image coordinate systems

   ~~2)~~3) Polar, cylindrical, and spherical coordinate systems.

   ~~3)Image coordinate systems~~

e) Use an externally-maintained database of ~~CRS~~ coordinate reference system and ~~CT~~ coordinate transformation definitions, and specify the format for input of information from that database.

f) Support interactive and/or automated selection of coordinate transformations from multiple available alternatives. For example, expanded server interface capabilities might:

   1) Allow client to retrieve list of available coordinate reference systems.

   2) Allow client to retrieve list of available coordinate transformations for identified source and target coordinate reference systems

   3) Allow client to retrieve only selected parts of identified coordinate reference system definition or coordinate transformation definition.

   4) Require server software to provide specified capabilities for automated selection of specific coordinate transformation from multiple available alternatives.

   5) Allow client to provide hints or guidance for automated server selection of specific coordinate transformation from multiple available alternatives.

Notice that some of these listed additional capabilities might be provided though separately specified interfaces, which could be implemented by either the same or separate server software packages.

Bibliography

[1] OGC Abstract Specification Topic 0: Overview, OGC document 99-100r1

[2] Guidelines for Successful OGC Interface Specifications, OGC document 00-014r1

[3] ISO 31 (all parts), *Quantities and units.*