Open Geospatial Consortium

Submission Date: 2021-02-23

Approval Date: 2021-03-26

Publication Date: 2021-07-02

External identifier of this OGC® document: http://www.opengis.net/doc/AS/topic-2/5.0.1

Internal reference number of this OGC® document: 18-005r5

Version: 5.0.1

Category: OGC Abstract Specification

Editor: Roger Lott

OGC Abstract Specification

Topic 2: Referencing by coordinates Corrigendum

Copyright notice

Warning

This document is an OGC Member approved international standard. This document is available on a royalty free, non-discriminatory basis.

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

Document type: OGC® Standard
Document subtype: Abstract Specification
Document stage: Approved
Document language: English

i.   Keywords

The following are keywords to be used by search engines and document catalogues.

Ogcdoc, OGC document, Topic 2, Spatial Referencing, Referencing by coordinates

ii.   Preface

This document is consistent with the third edition (2019) of ISO 19111, Geographic Information - Referencing by coordinates. ISO 19111:2019 was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics*, in close collaboration with the Open Geospatial Consortium (OGC). It replaces the second edition, ISO 19111:2007 and also ISO 19111-2:2009, OGC documents 08-015r2 and 10-020. This OGC document, 18-005r5, incorporates three editorial corrections made in ISO 19111:2019 amendment 1 of 2021.

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

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

iii.   Submitting organizations

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

Name	Affiliation
Roger Lott (editor)	IOGP
Keith Ryden	ESRI
Martin Desruisseaux	Geomatys
Mark Hedley	UK Met Office
Charles Heazel	WiSC Enterprises

All questions regarding this submission should be directed to the editor or the submitters:

iv.   Revision history

Date	Release	Author	Paragraph modified	Description
2018-03-01	1.0.0	Roger Lott		Initial draft
2018-04-04	1.0.1	Roger Lott	(iii) Submitting organisations	Additional submitters added
2018-08-23	1.0.2	Roger Lott	Minir revisions throughout to address comments made in ISO DIS ballot and OGC RFC	As submitted to ISO for publication as IS.
2018-09-13	1.03	Roger Lott	Figures 5, 9, 13 and 214 replaced, tables 2, 9, 50 and 64 updated.	Correction of UML errors.
2019-01-17	1.04	Roger Lott		Minor editorial corrections.
2021-02-22	5.0.1	Roger Lott	3.18, 7.4 figure 5 and table 2.	Corrigenda.

Contents

Figures

Tables

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) (see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 211 *Geographic information/Geomatics*, in close collaboration with the Open Geospatial Consortium (OGC).

This third edition cancels and replaces the second edition (ISO 19111:2007), which has been technically revised. This document also incorporates the provisions of ISO 19111-2:2009, which is cancelled.

The changes in this edition compared to the previous edition are:

— inclusion of applicable modern geodetic terminology;

— extension to describe dynamic geodetic reference frames;

— extension to describe geoid-based vertical coordinate reference systems;

— extension to allow triaxial ellipsoid for planetary applications;

— extension to describe three-dimensional projected coordinate reference systems;

— addition of 'datum ensembles' to allow grouping of related realizations of a reference frame where for lower accuracy applications the differences are insignificant;

— clarification in the modelling of derived coordinate reference systems;

— remodelling of the metadata elements scope and extent;

— addition of requirements to describe coordinate metadata and the relationship between spatial coordinates.

— additional modelling of temporal coordinate reference system components sufficient for spatio-temporal coordinate referencing;

— consolidation of the provisions of ISO 19111-2:2009 (Spatial referencing by coordinates - Extension for parametric values) into this document;

— change in name from 'Spatial referencing by coordinates' to 'Referencing by coordinates', due to the inclusion of the non-spatial coordinate reference system subtypes of parametric (from 19111-2) and temporal.

— the correction of minor errors.

Further details are given in Annex G.

In accordance with the ISO/IEC Directives, Part 2, 2018, *Rules for the structure and drafting of International Standards*, in International Standards the decimal sign is a comma on the line. However the General Conference on Weights and Measures (*Conférence Générale des Poids et Mesures*) at its meeting in 2003 passed unanimously the following resolution:
        "The decimal marker shall be either a point on the line or a comma on the line."
In practice, the choice between these alternatives depends on customary use in the language concerned. In the technical areas of geodesy and geographic information it is customary for the decimal point always to be used, for all languages. That practice is used throughout this document.

Introduction

Geographic information is inherently four-dimensional and includes time. The spatial component relates the features represented in geographic data to positions in the real world. Spatial references fall into two categories:

— those using coordinates;

— those based on geographic identifiers.

Spatial referencing by geographic identifiers is defined in ISO 19112[5]. This document describes the data elements, relationships and associated metadata required for spatial referencing by coordinates, expanded from a strictly spatial context to include time. The temporal element is restricted to temporal coordinate systems having a continuous axis. The temporal element excludes calendars and ordinal reference systems due to their complexities in definition and in transformation. The context is shown in Figure 1.
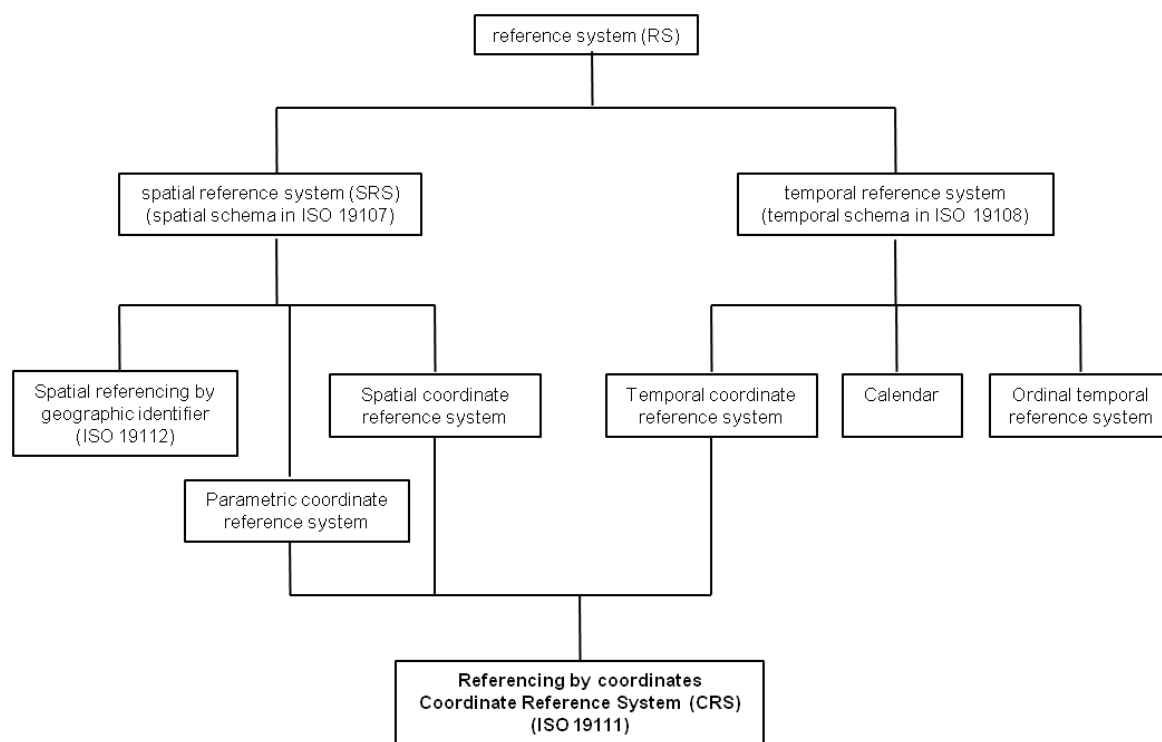
Figure 1 — Context of referencing by coordinates

Certain scientific communities use three-dimensional systems where horizontal position is combined with a non-spatial parameter. In these communities, the parameter is considered to be a third, vertical, axis. The parameter, although varying monotonically with height or depth, does not necessarily vary in a simple manner. Thus conversion from the parameter to height or depth is non-trivial. The parameters concerned are normally absolute measurements and the datum is taken with reference to a direct physical measurement of the parameter. These non-spatial parameters and parametric coordinate reference system modelling constructs were previously described in ISO 19111-2:2009 but have been incorporated into this revision because the modelling constructs are identical to the other coordinate reference system types included in this document.

This document describes the elements that are necessary to fully define various types of coordinate reference systems applicable to geographic information. The subset of elements required is partially dependent upon the type of coordinates. This document also includes optional fields to allow for the

inclusion of metadata about the coordinate reference systems. The elements are intended to be both machine and human readable.

In addition to describing a coordinate reference system, this document provides for the description of a coordinate operation between two different coordinate reference systems or a coordinate operation to account for crustal motion over time. With such information, spatial data referenced to different coordinate reference systems can be referenced to one specified coordinate reference system at one specified time. This facilitates spatial data integration. Alternatively, an audit trail of coordinate manipulations can be maintained.

Geographic information — Referencing by coordinates

1 Scope

This document defines the conceptual schema for the description of referencing by coordinates. It describes the minimum data required to define coordinate reference systems. This document supports the definition of:

— spatial coordinate reference systems where coordinate values do not change with time. The system may:

  — be geodetic and apply on a national or regional basis, or

  — apply locally such as for a building or construction site, or

  — apply locally to an image or image sensor;

  — be referenced to a moving platform such as a car, a ship, an aircraft or a spacecraft. Such a coordinate reference system may be related to a second coordinate reference system which is referenced to the Earth through a transformation that includes a time element;

— spatial coordinate reference systems in which coordinate values of points on or near the surface of the earth change with time due to tectonic plate motion or other crustal deformation. Such dynamic systems include time evolution, however they remain spatial in nature;

— parametric coordinate reference systems which use a non-spatial parameter that varies monotonically with height or depth;

— temporal coordinate reference systems which use dateTime, temporal count or temporal measure quantities that vary monotonically with time;

— mixed spatial, parametric or temporal coordinate reference systems.

The *definition* of a coordinate reference system does not change with time, although in some cases some of the defining parameters may include a rate of change of the parameter. The coordinate values within a dynamic and in a temporal coordinate reference system may change with time.

This document also describes the conceptual schema for defining the information required to describe operations that change coordinate values.

In addition to the minimum data required for the definition of the coordinate reference system or coordinate operation, the conceptual schema allows additional descriptive information - coordinate reference system metadata - to be provided.

This document is applicable to producers and users of geographic information. Although it is applicable to digital geographic data, the principles described in this document can be extended to many other forms of spatial data such as maps, charts and text documents.

2   Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 8601, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO 19103, *Geographic information — Conceptual schema language*

ISO 19115-1:2014, *Geographic information — Metadata — Part 1: Fundamentals*

3   Terms, definitions, symbols and abbreviated terms

3.1   Terms and definitions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at https://www.iso.org/obp

3.1.1
affine coordinate system
coordinate system in Euclidean space with straight axes that are not necessarily mutually perpendicular

3.1.2
Cartesian coordinate system
coordinate system in Euclidean space which gives the position of points relative to n mutually perpendicular straight axes all having the same unit of measure

Note 1 to entry: n is 2 or 3 for the purposes of this document.

Note 2 to entry: A Cartesian coordinate system is a specialisation of an affine coordinate system.

3.1.3
compound coordinate reference system
coordinate reference system using at least two independent coordinate reference systems

Note 1 to entry: Coordinate reference systems are independent of each other if coordinate values in one cannot be converted or transformed into coordinate values in the other.

3.1.4
concatenated operation
coordinate operation consisting of the sequential application of multiple coordinate operations

3.1.5
coordinate
one of a sequence of numbers designating the position of a point

Note 1 to entry: In a spatial coordinate reference system, the coordinate numbers are qualified by units.

3.1.6
coordinate conversion
coordinate operation that changes coordinates in a source coordinate reference system to coordinates in a target coordinate reference system in which both coordinate reference systems are based on the same datum

Note 1 to entry: A coordinate conversion uses parameters which have specified values.

EXAMPLE 1   A mapping of ellipsoidal coordinates to Cartesian coordinates using a map projection.

EXAMPLE 2   Change of units such as from radians to degrees or from feet to metres.

3.1.7
coordinate epoch
epoch to which coordinates in a dynamic coordinate reference system are referenced

3.1.8
coordinate operation
process using a mathematical model, based on a one-to-one relationship, that changes coordinates in a source coordinate reference system to coordinates in a target coordinate reference system, or that changes coordinates at a source coordinate epoch to coordinates at a target coordinate epoch within the same coordinate reference system

Note 1 to entry: Generalization of coordinate conversion, coordinate transformation and point motion operation.

3.1.9
coordinate reference system
coordinate system that is related to an object by a datum

Note 1 to entry: Geodetic and vertical datums are referred to as reference frames.

Note 2 to entry: For geodetic and vertical reference frames, the object will be the Earth. In planetary applications, geodetic and vertical reference frames may be applied to other celestial bodies.

3.1.10
coordinate set
collection of coordinate tuples referenced to the same coordinate reference system and if that coordinate reference system is dynamic also to the same coordinate epoch

3.1.11
coordinate system
set of mathematical rules for specifying how coordinates are to be assigned to points

3.1.12
coordinate transformation
coordinate operation that changes coordinates in a source coordinate reference system to coordinates in a target coordinate reference system in which the source and target coordinate reference systems are based on different datums

Note 1 to entry: A coordinate transformation uses parameters which are derived empirically. Any error in those coordinates will be embedded in the coordinate transformation and when the coordinate transformation is applied the embedded errors are transmitted to output coordinates.

Note 2 to entry: A coordinate transformation is colloquially sometimes referred to as a 'datum transformation'. This is erroneous. A coordinate transformation changes coordinate values. It does not change the definition of the datum. In this document coordinates are referenced to a coordinate reference system. A coordinate transformation operates between two coordinate reference systems, not between two datums.

3.1.13
coordinate tuple
tuple composed of coordinates

Note 1 to entry: The number of coordinates in the coordinate tuple equals the dimension of the coordinate system; the order of coordinates in the coordinate tuple is identical to the order of the axes of the coordinate system.

3.1.14
cylindrical coordinate system
three-dimensional coordinate system in Euclidean space in which position is specified by two linear coordinates and one angular coordinate

3.1.15
datum
reference frame
parameter or set of parameters that realize the position of the origin, the scale, and the orientation of a coordinate system

3.1.16
datum ensemble
group of multiple realizations of the same terrestrial or vertical reference system that, for approximate spatial referencing purposes, are not significantly different

Note 1 to entry: Datasets referenced to the different realizations within a datum ensemble may be merged without coordinate transformation.

Note 2 to entry: 'Approximate' is for users to define but typically is in the order of under 1 decimetre but may be up to 2 metres.

EXAMPLE   "WGS 84" as an undifferentiated group of realizations including WGS 84 (TRANSIT), WGS 84 (G730), WGS 84 (G873), WGS 84 (G1150), WGS 84 (G1674) and WGS 84 (G1762). At the surface of the Earth these have changed on average by 0.7m between the TRANSIT and G730 realizations, a further 0.2m between G730 and G873, 0.06m between G873 and G1150, 0.2m between G1150 and G1674 and 0.02m between G1674 and G1762).

3.1.17
depth
distance of a point from a chosen vertical reference surface downward along a line that is perpendicular to that surface

Note 1 to entry: The line direction may be straight, or be dependent on the Earth's gravity field or other physical phenomena.

Note 2 to entry: A depth above the vertical reference surface will have a negative value.

3.1.18
derived coordinate reference system
coordinate reference system that is defined through the application of a specified coordinate conversion to the coordinates within a previously established coordinate reference system

Note 1 to entry: The previously established coordinate reference system is referred to as the base coordinate reference system.

Note 2 to entry: A derived coordinate reference system inherits its datum or reference frame from its base coordinate reference system.

Note 3 to entry: The coordinate conversion between the base and derived coordinate reference system is implemented using the parameters and formula(s) specified in the definition of the coordinate conversion.

3.1.19
dynamic coordinate reference system
coordinate reference system that has a dynamic reference frame

Note 1 to entry: Coordinates of points on or near the crust of the Earth that are referenced to a dynamic coordinate reference system may change with time, usually due to crustal deformations such as tectonic motion and glacial isostatic adjustment.

Note 2 to entry: Metadata for a dataset referenced to a dynamic coordinate reference system should include coordinate epoch information.

3.1.20
dynamic reference frame
dynamic datum
reference frame in which the defining parameters include time evolution

Note 1 to entry: The defining parameters that have time evolution are usually a coordinate set.

3.1.21
easting
E
distance in a coordinate system, eastwards (positive) or westwards (negative) from a north-south reference line

3.1.22
ellipsoid
reference ellipsoid
<geodesy> geometric reference surface embedded in 3D Euclidean space formed by an ellipse that is rotated about a main axis

Note 1 to entry: For the Earth the ellipsoid is bi-axial with rotation about the polar axis. This results in an oblate ellipsoid with the midpoint of the foci located at the nominal centre of the Earth.

3.1.23
ellipsoidal coordinate system
geodetic coordinate system
coordinate system in which position is specified by geodetic latitude, geodetic longitude and (in the three-dimensional case) ellipsoidal height

3.1.24
ellipsoidal height
geodetic height
h
distance of a point from the reference ellipsoid along the perpendicular from the reference ellipsoid to this point, positive if upwards or outside of the reference ellipsoid

Note 1 to entry: Only used as part of a three-dimensional ellipsoidal coordinate system or as part of a three-dimensional Cartesian coordinate system in a three-dimensional projected coordinate reference system, but never on its own.

3.1.25
engineering coordinate reference system
coordinate reference system based on an engineering datum

EXAMPLE 1   System for identifying relative positions within a few kilometres of the reference point, such as a building or construction site.

EXAMPLE 2   Coordinate reference system local to a moving object such as a ship or an orbiting spacecraft.

EXAMPLE 3   Internal coordinate reference system for an image. This has continuous axes. It may be the foundation for a grid.

3.1.26
engineering datum
local datum
datum describing the relationship of a coordinate system to a local reference

Note 1 to entry: Engineering datum excludes both geodetic and vertical reference frames.

3.1.27
epoch
<geodesy> point in time

Note 1 to entry: In this document an epoch is expressed in the Gregorian calendar as a decimal year.

EXAMPLE 2017-03-25 in the Gregorian calendar is epoch 2017.23.

3.1.28
flattening
f
ratio of the difference between the semi-major axis (a) and semi-minor axis (b) of an ellipsoid to the semi-major axis: $f = (a - b)/a$

Note 1 to entry: Sometimes inverse flattening $1/f = a/(a - b)$ is given instead; $1/f$ is also known as reciprocal flattening.

3.1.29
frame reference epoch
epoch of coordinates that define a dynamic reference frame

3.1.30
geocentric latitude
angle from the equatorial plane to the direction from the centre of an ellipsoid through a given point, northwards treated as positive

3.1.31
geodetic coordinate reference system
three-dimensional coordinate reference system based on a geodetic reference frame and having either a three-dimensional Cartesian or a spherical coordinate system

Note 1 to entry: In this document a coordinate reference system based on a geodetic reference frame and having an ellipsoidal coordinate system is geographic.

3.1.32
geodetic latitude
ellipsoidal latitude
φ
angle from the equatorial plane to the perpendicular to the ellipsoid through a given point, northwards treated as positive

3.1.33
geodetic longitude
ellipsoidal longitude
λ
angle from the prime meridian plane to the meridian plane of a given point, eastward treated as positive

3.1.34
geodetic reference frame
reference frame or datum describing the relationship of a two- or three-dimensional coordinate system to the Earth

Note 1 to entry: In the data model described in this document, the UML class GeodeticReferenceFrame includes both modern terrestrial reference frames and classical geodetic datums.

3.1.35
geographic coordinate reference system
coordinate reference system that has a geodetic reference frame and an ellipsoidal coordinate system

3.1.36
geoid
equipotential surface of the Earth's gravity field which is perpendicular to the direction of gravity and which best fits mean sea level either locally, regionally or globally

3.1.37
gravity-related height
H
height that is dependent on the Earth's gravity field

Note 1 to entry: This refers to, amongst others, orthometric height and Normal height, which are both approximations of the distance of a point above the mean sea level, but also may include Normal-orthometric heights, dynamic heights or geopotential numbers.

Note 2 to entry: The distance from the reference surface may follow a curved line, not necessarily straight, as it is influenced by the direction of gravity.

3.1.38
height
distance of a point from a chosen reference surface positive upward along a line perpendicular to that surface

Note 1 to entry: A height below the reference surface will have a negative value.

Note 2 to entry: Generalisation of ellipsoidal height (h) and gravity-related height (H).

3.1.39
linear coordinate system
one-dimensional coordinate system in which a linear feature forms the axis

EXAMPLE 1   Distances along a pipeline.

EXAMPLE 2   Depths down a deviated oil well bore.

3.1.40
map projection
coordinate conversion from an ellipsoidal coordinate system to a plane

3.1.41
mean sea level
MSL
<geodesy> average level of the surface of the sea over all stages of tide and seasonal variations

Note 1 to entry: Mean sea level in a local context normally means mean sea level for the region calculated from observations at one or more points over a given period of time. To meet IHO standards that period should be one full lunar cycle of 19 years. Mean sea level in a global context differs from a global geoid by not more than 2 m.

3.1.42
meridian
intersection of an ellipsoid by a plane containing the shortest axis of the ellipsoid

Note 1 to entry: This term is generally used to describe the pole-to-pole arc rather than the complete closed figure.

3.1.43
northing
N
distance in a coordinate system, northwards (positive) or southwards (negative) from an east-west reference line

3.1.44
parameter reference epoch
epoch at which the parameter values of a time-dependent coordinate transformation are valid

Note 1 to entry: The transformation parameter values first need to be propagated to the epoch of the coordinates before the coordinate transformation can be applied.

3.1.45
parametric coordinate reference system
coordinate reference system based on a parametric datum

3.1.46
parametric coordinate system
one-dimensional coordinate system where the axis units are parameter values which are not inherently spatial

3.1.47
parametric datum
datum describing the relationship of a parametric coordinate system to an object

Note 1 to entry: The object is normally the Earth.

3.1.48
point motion operation
coordinate operation that changes coordinates within one coordinate reference system due to the motion of the point

Note 1 to entry: The change of coordinates is from those at an initial epoch to those at another epoch.

Note 2 to entry: In this document the point motion is due to tectonic motion or crustal deformation.

3.1.49
polar coordinate system
two-dimensional coordinate system in Euclidean space in which position is specified by one distance coordinate and one angular coordinate

Note 1 to entry: For the three-dimensional case, see spherical coordinate system.

3.1.50
prime meridian
meridian from which the longitudes of other meridians are quantified

3.1.51
projected coordinate reference system
coordinate reference system derived from a geographic coordinate reference system by applying a map projection

Note 1 to entry: May be two- or three-dimensional, the dimension being equal to that of the geographic coordinate reference system from which it is derived.

Note 2 to entry: In the three-dimensional case the horizontal coordinates (geodetic latitude and geodetic longitude coordinates) are projected to northing and easting and the ellipsoidal height is unchanged.

3.1.52
reference frame
datum
parameter or set of parameters that realize the position of the origin, the scale, and the orientation of a coordinate system

3.1.53
semi-major axis
a
semi-diameter of the longest axis of an ellipsoid

3.1.54
semi-minor axis
b
semi-diameter of the shortest axis of an ellipsoid

3.1.55
sequence
finite, ordered collection of related items (objects or values) that may be repeated

3.1.56
spatial reference
description of position in the real world

Note 1 to entry: This may take the form of a label, code or coordinate tuple.

3.1.57
spatio-parametric coordinate reference system
compound coordinate reference system in which one constituent coordinate reference system is a spatial coordinate reference system and one is a parametric coordinate reference system

Note 1 to entry: Normally the spatial component is "horizontal" and the parametric component is "vertical".

3.1.58
spatio-parametric-temporal coordinate reference system
compound coordinate reference system comprised of spatial, parametric and temporal coordinate reference systems

3.1.59
spatio-temporal coordinate reference system
compound coordinate reference system in which one constituent coordinate reference system is a spatial coordinate reference system and one is a temporal coordinate reference system

3.1.60
spherical coordinate system
three-dimensional coordinate system in Euclidean space in which position is specified by one distance coordinate and two angular coordinates

Note 1 to entry: Not to be confused with an ellipsoidal coordinate system based on an ellipsoid 'degenerated' into a sphere.

3.1.61
static coordinate reference system
coordinate reference system that has a static reference frame

Note 1 to entry: Coordinates of points on or near the crust of the Earth that are referenced to a static coordinate reference system do not change with time.

Note 2 to entry: Metadata for a dataset referenced to a static coordinate reference system does not require coordinate epoch information.

3.1.62
static reference frame
static datum
reference frame in which the defining parameters exclude time evolution

3.1.63
temporal coordinate reference system
coordinate reference system based on a temporal datum

3.1.64
temporal coordinate system
<geodesy> one-dimensional coordinate system where the axis is time

3.1.65
temporal datum
datum describing the relationship of a temporal coordinate system to an object

Note 1 to entry: The object is normally time on the Earth.

3.1.66
terrestrial reference system
TRS
set of conventions defining the origin, scale, orientation and time evolution of a spatial reference system co-rotating with the Earth in its diurnal motion in space

Note 1 to entry: The abstract concept of a TRS is realised through a terrestrial reference frame that usually consists of a set of physical points with precisely determined coordinates and optionally their rates of change. In this document terrestrial reference frame is included within the geodetic reference frame element of the data model.

3.1.67
transformation reference epoch
epoch at which the parameter values of a time-specific coordinate transformation are valid

Note 1 to entry: Coordinates first need to be propagated to this epoch before the coordinate transformation is applied. This is in contrast to a *parameter reference epoch* where the transformation parameter values first need to be propagated to the epoch of the coordinates before the coordinate transformation is applied.

3.1.68

tuple

ordered list of values

[SOURCE: ISO 19136:2007, 4.1.63]

3.1.69

unit

defined quantity in which dimensioned parameters are expressed

Note 1 to entry: In this document, the subtypes of units are length units, angular units, scale units, parametric quantities and time quantities.

3.1.70

vertical coordinate reference system

one-dimensional coordinate reference system based on a vertical reference frame

3.1.71

vertical coordinate system

one-dimensional coordinate system used for gravity-related height or depth measurements

3.1.72

vertical reference frame

vertical datum

reference frame describing the relation of gravity-related heights or depths to the Earth

Note 1 to entry: In most cases, the vertical reference frame will be related to mean sea level. Vertical datums include sounding datums (used for hydrographic purposes), in which case the heights may be negative heights or depths.

Note 2 to entry: Ellipsoidal heights are related to a three-dimensional ellipsoidal coordinate system referenced to a geodetic reference frame.

3.1.73

vertical reference system

VRS

set of conventions defining the origin, scale, orientation and time evolution that describes the relationship of gravity-related heights or depths to the Earth

Note 1 to entry: The abstract concept of a VRS is realised through a vertical reference frame.

3.2   Symbols

a semi-major axis of ellipsoid

b semi-minor axis of bi-axial ellipsoid

E easting

f flattening

H gravity-related height

h ellipsoidal height

N northing

λ geodetic longitude

φ geodetic latitude

E, N, [h]   Cartesian coordinates in a projected coordinate reference system

X, Y, Z Cartesian coordinates in a geodetic coordinate reference system

i, j, [k]   Cartesian coordinates in an engineering coordinate reference system

r, θ polar coordinates in a 2D engineering coordinate reference system

r, Ω, θ spherical coordinates in a 3D engineering coordinate reference system
Note: In this document Ω is the polar (zenith) angle and θ is the azimuthal angle.

φ, λ, [h]   ellipsoidal coordinates in a geographic coordinate reference system

3.3   Abbreviated terms

CC   coordinate conversion

CCRS   compound coordinate reference system

CRS   coordinate reference system

CT   coordinate transformation

MSL   mean sea level

pixel   a contraction of "picture element", the smallest element of a digital image to which attributes are assigned

PMO   point motion operation

SI   le Système International d'unités (International System of Units)

UML   Unified Modeling Language

URI   Uniform Resource Identifier

1D   one-dimensional

2D   two-dimensional

3D   three-dimensional

4   Conformance requirements

This document defines

— two classes of conformance for relating coordinates to coordinate metadata; and

— twenty six classes of conformance for the definition of a coordinate reference system (CRS) or of a coordinate operation.

These are differentiated by type, as shown in Table 1. Implementations should indicate which conformance classes they comply with. Any implementations claiming conformance shall satisfy the requirements in Annex A.

Table 1 — Conformance classes

Conformance class	Description	Conformance requirements given in
Conformance for relating coordinates to coordinate metadata		**A.2**
1	CRS with static reference frame	
2	CRS with dynamic reference frame	
Conformance of a CRS definition		**A.3**
	Geodetic CRS	
3	with static reference frame	
4	with dynamic reference frame	
5	derived geodetic CRS	
	Geographic CRS	
6	with static reference frame	
7	with dynamic reference frame	
8	derived geographic CRS	
9	Projected CRS	
10	derived projected CRS	
	Vertical CRS	
11	with static reference frame	
12	with dynamic reference frame	
13	derived vertical CRS	
14	Parametric CRS	
15	derived parametric CRS	
16	Engineering CRS	
17	derived engineering CRS	
	Temporal CRS	
18	dateTime	
19	temporal count	
20	temporal measure	
21	derived temporal CRS	
22	CRS with datum ensemble	

Conformance class	Description	Conformance requirements given in
23	Compound CRS	A.3
Conformance of a coordinate operation definition		**A.4**
24	Coordinate conversion	
25	Coordinate transformation	
26	Point motion operation	
27	Concatenated operation	
28	Pass-through operation	

The requirements classes for the definition of a coordinate reference system or a coordinate operation are described in this document through tables grouped by UML package. The requirements are then brought together in the conformance classes in Annex A. This retains the package-based layout for describing requirements used in previous versions of this document.

5   Conventions

5.1   Unified Modeling Language notation

In this document, the conceptual schema for describing coordinate reference systems and coordinate operations are presented in the Unified Modeling Language (UML). ISO 19103 *Conceptual schema language* presents the specific profile of UML used in this document.

In the UML diagrams in this document, a grey background surround to boxes indicates classes from other standards.

5.2   Attribute and association status

In this document the conceptual schema is described in Clauses 6 to 12 through tables. In these tables:

- attributes and associations are given an obligation status:

Obligation	Definition	Meaning
M	mandatory	This attribute shall be supplied.
C	conditional	This attribute shall be supplied if the condition (given in the attribute description) is true. It may be supplied if the condition is false.
O	optional	This attribute may be supplied.

  The Maximum Occurrence column in the tables indicates the maximum number of occurrences of attribute values that are permissible, with N indicating no upper limit.

- non-navigable associations are not included in the UML diagrams or tables.

In the event of any discrepancies between the UML diagrams and text, the UML shall prevail.

6 Referencing by coordinates - Data model overview

The specification for referencing by coordinates is described in this document in the form of a UML model with supplementary text. The UML model contains six UML packages, as shown in Figure 2. Each box represents a package, and contains the package name. Each arrowed line shows the dependency of one package upon another package (at the head of the arrow).
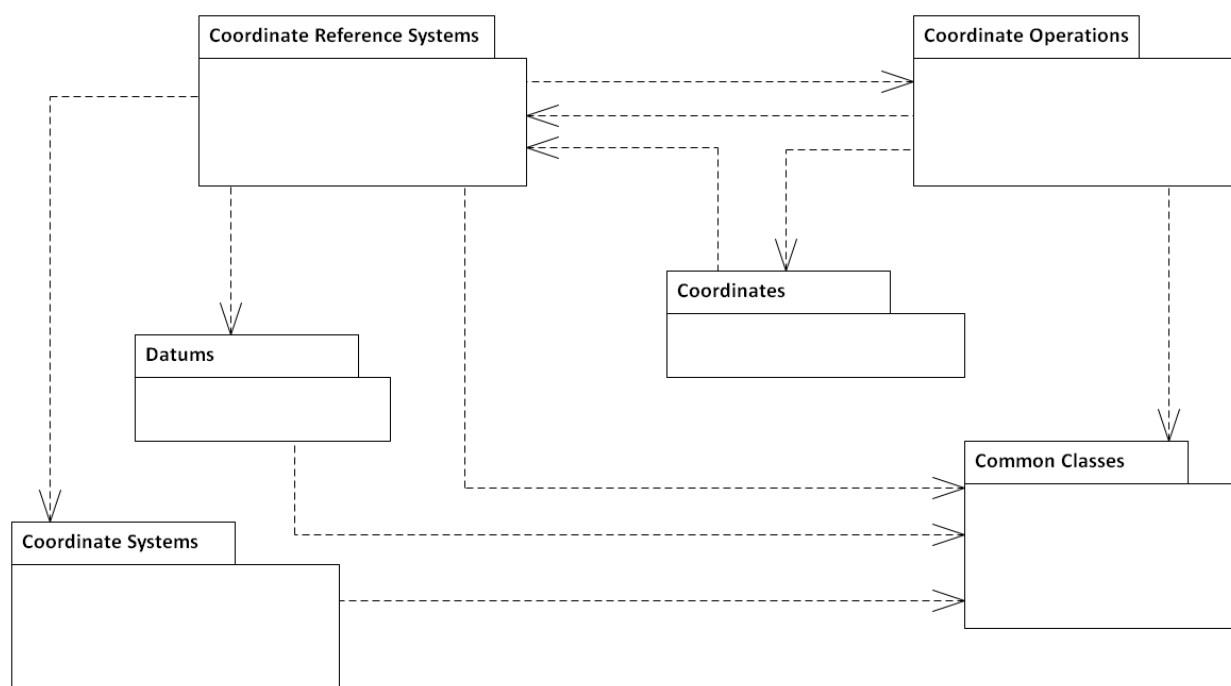
Figure 2 — UML model packages and dependencies

Coordinates require metadata that fully specifies the coordinate reference system to which they are referenced; without this CRS reference the description of position is ambiguous. The UML package for coordinates and their metadata is described in Clause 7. This includes aspects of coordinate operations required to change coordinate values when the coordinate reference system is changed.

A coordinate reference system is usually comprised of two components, one coordinate system and one datum. In modern geodetic terminology the datum is referred to as a reference frame. Some geodetic concepts underpinning spatial referencing by coordinates are given in Annex B. The information required to fully specify a coordinate reference system is described in Clauses 9 to 11, with attributes common to all three packages described in Clause 8.

Some coordinate reference systems have a third component, a defining coordinate conversion from another pre-existing CRS. In this document a CRS having this third component is a derived CRS. The specification for describing coordinate operations, including a defining coordinate conversion, is described in Clause 12.

Further context for the requirements of Clauses 8 to 12 is given in Annexes C and D. Examples illustrating how the specifications of this document can be applied when defining a coordinate reference system or a coordinate operation are given in Annex E. Recommendations for referencing to classes defined in this document are given in Annex F. Changes between this document and the previous version ISO 19111:2007 are described in Annex G.

7 Coordinates package

7.1 Relationship between coordinates and coordinate reference system

In this document, a *coordinate* is one of *n* scalar values that define the position of a single point. In other contexts, the term *ordinate* is used for a single value and coordinate for multiple ordinates. Such usage is not part of this document.

A *coordinate tuple* is an ordered list of coordinates that define the position of a single point. The coordinates within a coordinate tuple are mutually independent. The number of coordinates in a tuple is equal to the dimension of the coordinate space.

A *coordinate set* is a collection of coordinate tuples referenced to the same coordinate reference system. For a coordinate set, one CRS identification or definition may be associated with the coordinate set and then all coordinate tuples in that coordinate set inherit that association. If only one point is being described, the association between coordinate tuple and coordinate reference system is direct.

The concepts of dynamic and static coordinate reference systems are outlined in B.3. If the coordinate reference system is dynamic, operations on the geometry of the tuples within the coordinate set are valid only if all tuples are referenced to the same coordinate epoch. In this document all coordinate tuples in a spatial coordinate set are referenced to one specified coordinate epoch.

Together the coordinate reference system and the coordinate epoch are the coordinate metadata.

Coordinate sets referenced to one CRS may be referenced to another CRS through the application of a coordinate operation. A coordinate operation operates on coordinates, not on coordinate reference systems. A coordinate operation may be single or concatenated: refer to Clause 12. The high level conceptual model for changing coordinates is shown in Figure 3.
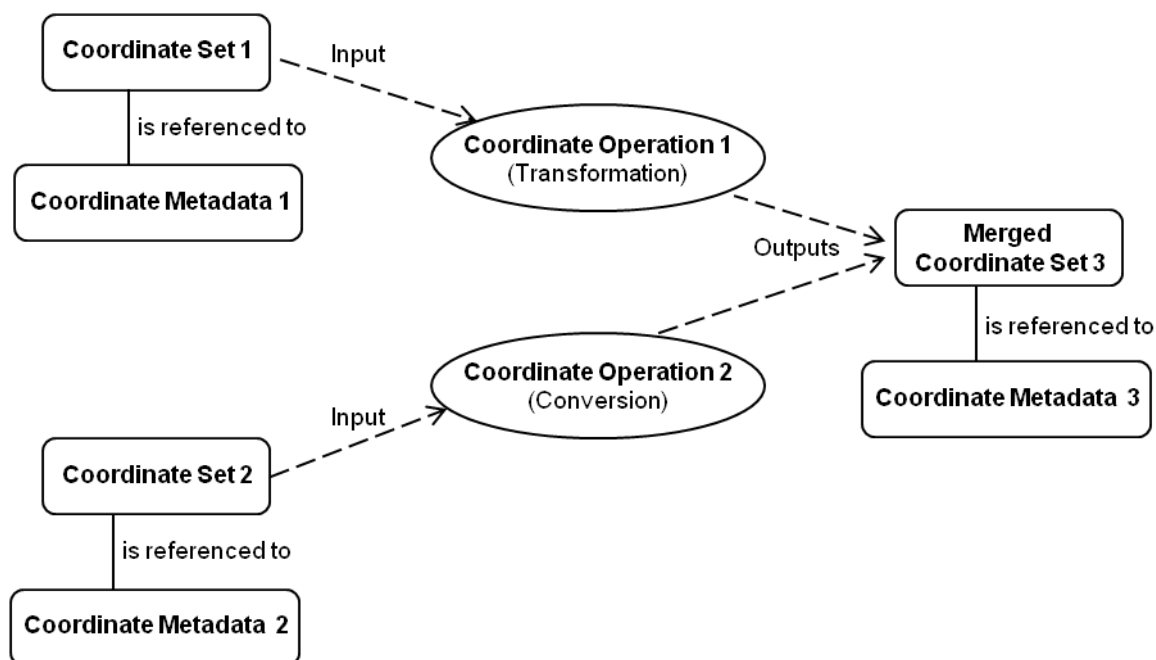
Figure 3 — Conceptual model for coordinate operations to produce a merged coordinate set

Coordinate sets referenced to a dynamic CRS at a given coordinate epoch t_1 may be converted to another coordinate epoch t_2 through a point motion coordinate operation that includes time evolution, often described using velocities, as shown schematically in Figure 4.
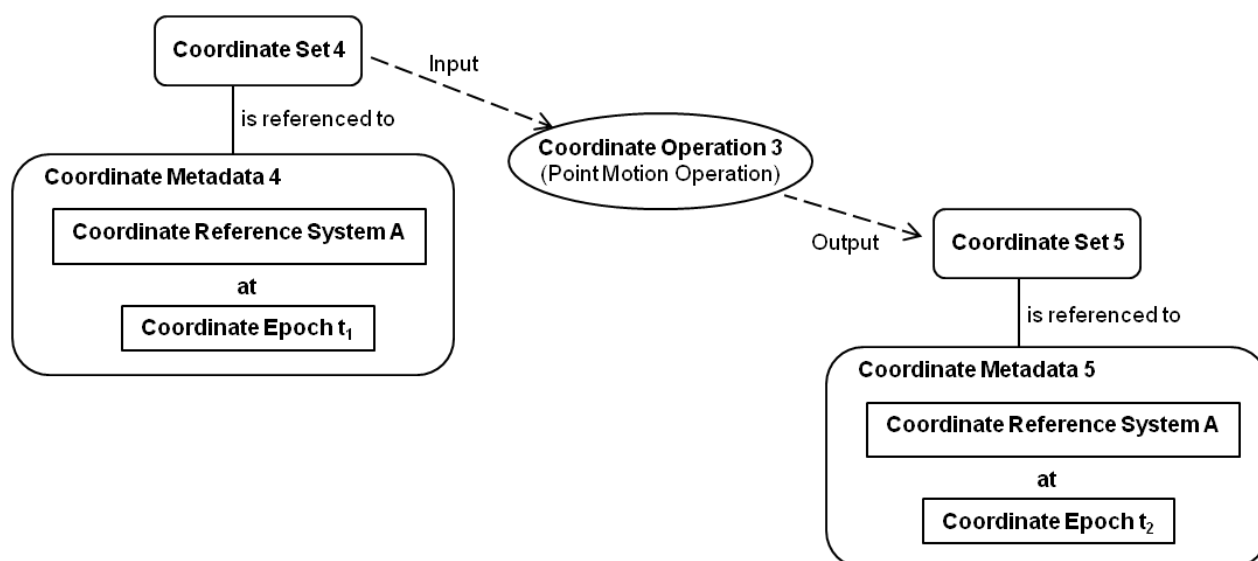
Figure 4 — Conceptual model for a coordinate operation to change coordinate epoch

It is also possible to change coordinates from being referenced to one dynamic CRS at one coordinate epoch to being referenced to another dynamic CRS at another coordinate epoch, or to change coordinates between a dynamic CRS and a static CRS or vice-versa. Further information is in C.1 and C.5.

The description of quality of coordinates is covered by the provisions of ISO 19157[8].

7.2   Coordinate reference system identification

The elements required for the definition of coordinate reference systems and coordinate operations are described in Clauses 8 to 12.

CRS or coordinate operation identification may be through:

a)   a full description, as defined in this document; or

b)   reference to a full description in a register of geodetic parameters (the reference is made to the register and to the identifier of the object description within that register); or

c)   both a full description and a reference to a full description in a register. If there is a conflict between the two, the object full description should prevail over the reference to a register.

a) and b) are alternative means of providing a full description. b) is recommended for simplicity, but if it is not available from a register the description is required to be given explicitly and in full. In both methods, the order of coordinates in each coordinate tuple is required to be as given in the coordinate reference system's coordinate system description.

When using method b), reference to a register, applications that are required only to confirm the *identification* of a CRS or coordinate operation can do so through the register citation and the identifier from that register. They do not need to retrieve the elements that constitute the full description from the register unless there is a need to quote these or to perform a coordinate operation on the coordinate set.

7.3   Requirements for coordinate metadata

7.3.1   Requirements class: static CRS coordinate metadata

Requirement 1: All coordinate tuples in a coordinate set shall be referenced to the same coordinate reference system.

7.3.2   Requirements class: dynamic CRS coordinate metadata

CRS is described in Clause 9 and datum or reference frame in Clause 11. The following subtypes of CRS may have a dynamic reference frame and therefore may be dynamic CRSs: geodetic, geographic, vertical, projected and derived variants of these subtypes. Implementers are warned that CRSs of these subtypes are not necessarily dynamic; their reference frame attributes need to be examined to clarify this.

Requirement 2: When the coordinate reference system to which a coordinate set is referenced is dynamic, all coordinate tuples in the coordinate set shall be referenced to the same coordinate epoch.

7.4   UML schema for the Coordinates package

Figure 5 shows the UML class diagram for coordinate metadata. The definition of the classes in the package are provided in Tables 2 to 4.
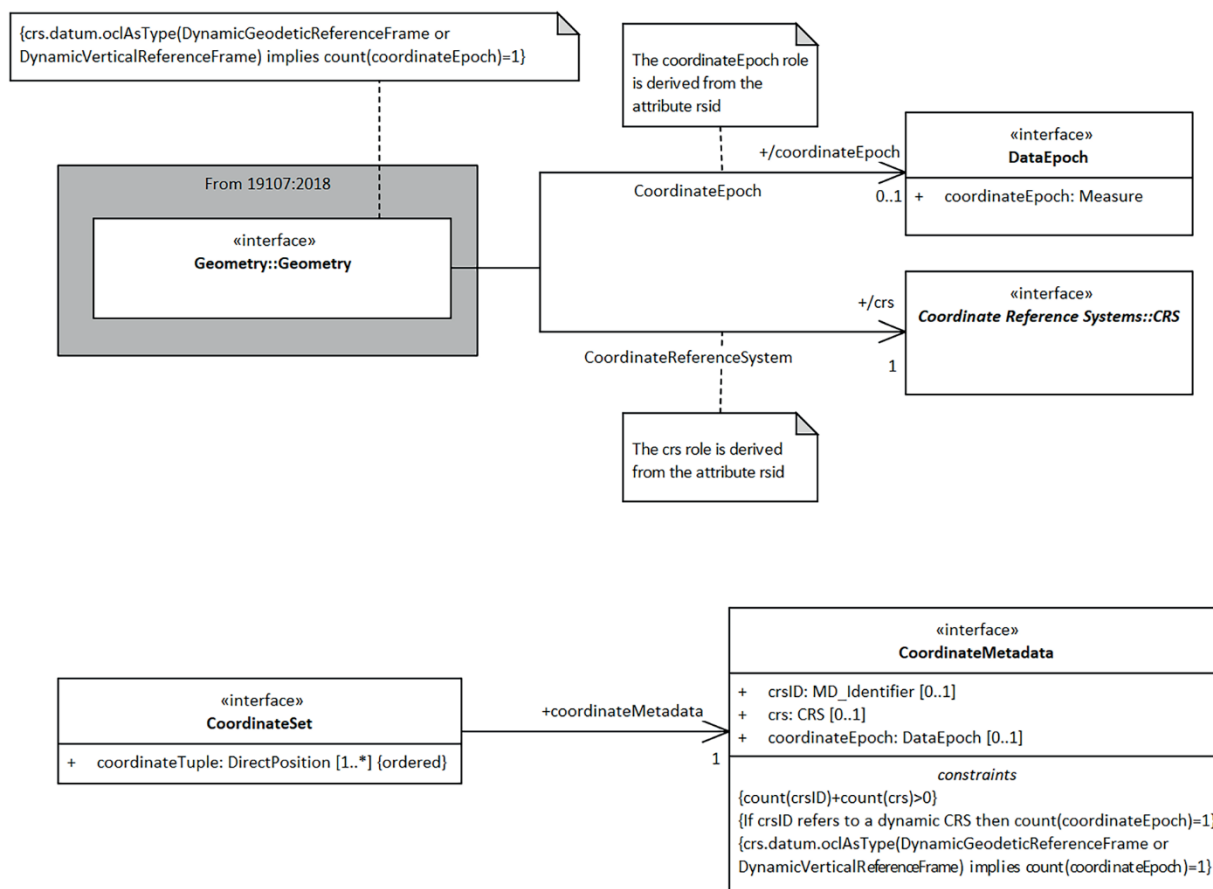
Figure 5 — UML diagram — Relationship of coordinates and coordinate metadata

Table 2 — Defining elements of Coordinates::CoordinateMetadata class

Definition:	metadata required to reference coordinates
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	(none)

Public attributes:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute definition
CRS ID	crsID	MD_Identifier	C	1	identifier of the coordinate reference system to which a coordinate set is referenced
CRS definition	crs	CRS	C	1	full definition of the coordinate reference system to which a coordinate set is referenced
Coordinate epoch	coordinateEpoch	DataEpoch	C	1	epoch at which a coordinate set referenced to a dynamic CRS is valid Note: Required if the CRS is dynamic.

Constraints:	{count(crsID)+count(CRS)>0} Remarks: See 7.2 {crs.datum.oclAsType(DynamicGeodeticReferenceFrame or DynamicVerticalReferenceFrame) implies count(coordinateEpoch)=1} {if crsID refers to a dynamic CRS then count(coordinateEpoch)=1} Remarks: The constraint provides the conditionallity for coordinate epoch.

The association of a coordinate set to a coordinate reference system (including the special case of a coordinate set containing only one tuple) is mandatory. The defining elements of the coordinate reference system class are described in Clause 9.

The constraint on coordinate metadata (repeated on geometry) specifies that if the coordinate reference system is dynamic then the coordinate set additionally is required to be related to a specified coordinate epoch. This enforces the conditionality of the coordinateEpoch attribute. Whether the CRS is dynamic is determined through the CRS's reference frame definition (Clause 11).

Table 3 — Defining elements of Coordinates::CoordinateSet class

Definition:	description of the coordinate tuples in a coordinate set Note: A single coordinate tuple is treated as a special case of coordinate set containing only one member.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	(none)

Association roles:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	coordinateMetadata	CoordinateMetadata	M	1	coordinate metadata to which this coordinate set is referenced

Public attributes:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute definition
Coordinate tuple	coordinateTuple	DirectPosition {ordered}	M	N	position described by a coordinate tuple

Table 4 — Defining elements of Coordinates::DataEpoch class

Definition:	time attribute of a coordinate set that is referenced to a dynamic CRS
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	(none)
Used by:	Coordinates::CoordinateMetadata
Association roles:	(Note attached to association from Geometry::Geometry: "The coordinateEpoch role is derived from the attribute rsid". See ISO 19107).

Public attributes:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute definition
Coordinate epoch	coordinateEpoch	Measure	M	1	date at which coordinates are referenced to a dynamic coordinate reference system, expressed as a decimal year in the Gregorian calendar Example: 2017-03-25 in the Gregorian calendar is epoch 2017.23.

7.5   UML schema for change of coordinates

Coordinates may be changed to be referenced to a different CRS. If the CRS is dynamic, coordinates also may be referenced to a different coordinate epoch, or to both a different CRS and different coordinate epoch.

In this document the CoordinateOperation class has two purposes:
   i)   To define the requirements for describing a coordinate operation
   ii) To apply the coordinate operation to change coordinates.

The defining elements for the CoordinateOperation class and its associated classes and their use in the definition of a coordinate operation are given in Clause 12. Only those attributes relevant to the change of coordinates are elaborated here.

Figure 6 shows the UML class diagram for the application of a coordinate operation to coordinate metadata.

´ interface²
CoordinateOperation

+   transform(CoordinateSet): CoordinateSet

constraints
{transform() pre sourceCRS=CoordinateSet.coordinateMetadata.crs}
{transform() pre count(CoordinateSet.coordinateMetadata.coordinateEpoch)=1 implies
sourceCoordinateEpoch=CoordinateSet.coordinateMetadata.coordinateEpoch}
{transform() post CoordinateSet.coordinateMetadata.crs=targetCRS}
{transform() post count(CoordinateSet.coordinateMetadata.coordinateEpoch)=1 implies
CoordinateSet.coordinateMetadata.coordinateEpoch=targetCoordinateEpoch}

Source   Target   SourceEpoch   TargetEpoch

+sourceCRS   +targetCRS   +sourceCoordinateEpoch   +targetCoordinateEpoch
0..1   0..1   0..1   0..1

´ interface²
Coordinate Reference Systems::CRS

´ interface²
Coordinates::DataEpoch

+   coordinateEpoch: Measure

Figure 6 — UML diagram — Relationship of coordinate operation and coordinate metadata

The operation CoordinateOperation.transform(CoordinateSet) changes coordinates

— from being referenced to one CRS to being referenced to a second CRS, and/or

— in the case of a dynamic CRS, from being referenced to one coordinate epoch to being referenced to a second coordinate epoch.

 Further information regarding commutation is given in C.1.3.

transform(CoordinateSet) has four constraints which collectively require that

— the source CRS and/or the source coordinate epoch shall be those to which the input coordinate set is referenced, and that

— the target CRS and/or the target coordinate epoch shall be those associated with the output coordinate set.

transform(CoordinateSet) operates on coordinate tuples which have the data type DirectPosition and are ordered. This implies that when transform(CoordinateSet) is applied to a coordinate set containing multiple coordinate tuples, the order of the tuples in the coordinate set is preserved.

NOTE   transform(CoordinateSet) operates on coordinate tuples and does not deal with interpolation of the specific geometry types. When a coordinate set is subjected to a coordinate operation, its geometry might or might not be preserved.

8   Common Classes package

8.1   General attributes

8.1.1   Introduction

The Common Classes package contains attributes common to several objects used in referencing by coordinates. These objects – CRS, datum, coordinate system and coordinate operation, together with some of their associated classes – inherit attribute values from the Common Classes package. This facilitates modular programming of names, identifiers and aliases, and of usage (scope and domain of validity).

8.1.2   Name and alias

One of the attributes is the primary name of the object. The object may have alternative names or aliases.

EXAMPLE   A datum name might be "North American Datum of 1983" and its abbreviation "NAD83".

Object primary names have a data type MD_Identifier which is defined in ISO 19115-1:2014. Aliases have a data type GenericName which is defined in ISO 19103.

8.1.3   Identifier

Another attribute is the identifier. This is a unique code used to reference an object in a given place.

EXAMPLE   A geodetic registry might give the NAD83 datum a unique code of "6269".

Identifiers have a data type of MD_Identifier.

In addition to the use of an identifier as a reference to a definition in a geodetic registry, it may also be included in an object definition to allow reference to that object.

8.1.4   Scope and Domain of Validity

Scope is a description of the primary purpose or purposes to which a coordinate reference system, datum or coordinate operation is applied.

DomainOfValidity is described in ISO 19115-1:2014, the introductory text of which is repeated here for convenience:

> The datatype in this [EX_Extent] package is an aggregate of the metadata elements that describe the spatial and temporal extent of resources, objects, events, or phenomena. The EX_Extent class contains information about the geographic (EX_GeographicExtent), temporal (EX_TemporalExtent) and the vertical (EX_VerticalExtent) extent of something. EX_GeographicExtent can be subclassed as EX_BoundingPolygon, EX_GeographicBoundingBox and EX_GeographicDescription. The combined spatial and temporal extent (EX_SpatialTemporalExtent) is an aggregate of EX_GeographicExtent. EX_SpatialTemporalExtent is a subclass of EX_TemporalExtent. The full package is specified in ISO 19115-1:2014 Figure 19.
>
> The EX_Extent class has three optional roles named "geographicElement", "temporalElement", and "verticalElement" and an element called "description". At least one of the four shall be used. The data dictionary for this diagram is located in ISO 19115-1:2014 Table B.15.

In this document Scope and DomainOfValidity are paired through the ObjectUsage.domain attribute. This facilitates descriptions of usage such as 'Purpose 1 in area A, purpose 2 in area B'.

Scope and DomainOfValidity are optional to facilitate a succinct CRS description using Well-Known Text in accordance with ISO 19162[10]. However it is strongly recommended that, in geodetic registries, the entries for coordinate reference systems, datums and coordinate operations should include at least one Scope-DomainOfValidity pairing. Additional Scope-DomainOfValidity pairings may optionally be given.

8.2   UML schema for the Common Classes package

Figure 7 shows the UML class diagram of the Common Classes package. The definition of the classes in the package are provided in Tables 5 to 7.
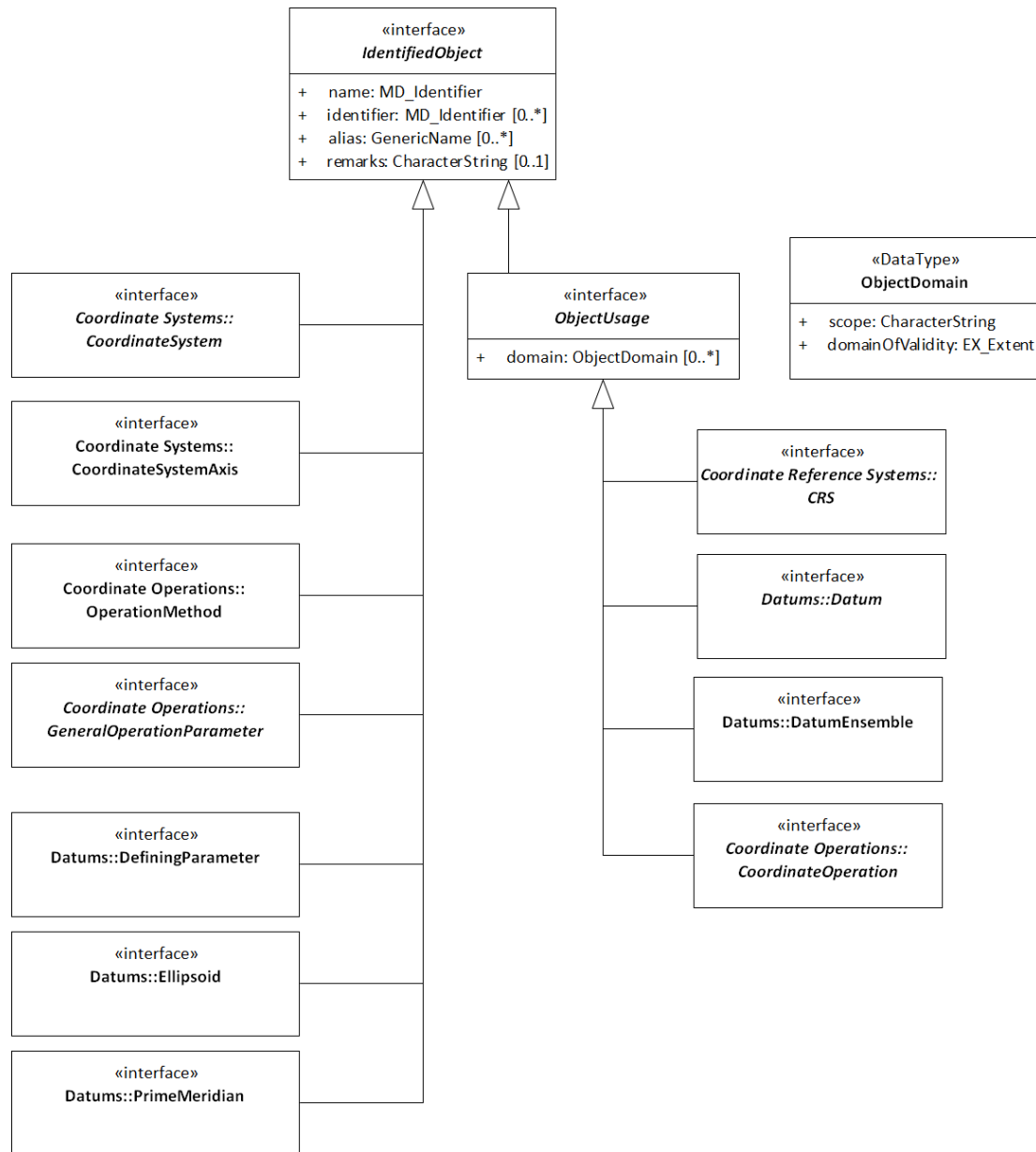
Figure 7 — UML diagram — Common Classes package

The data types MD_Identifier and EX_Extent are defined in ISO 19115-1:2014. The UML class diagram for the attributes in these classes which are of particular relevance to this document are shown in Figure 8. The EX_Extent class contains information about the geographic, vertical and temporal extent.

EX_GeographicExtent can be subclassed as EX_BoundingPolygon, EX_GeographicBoundingBox and EX_GeographicDescription.
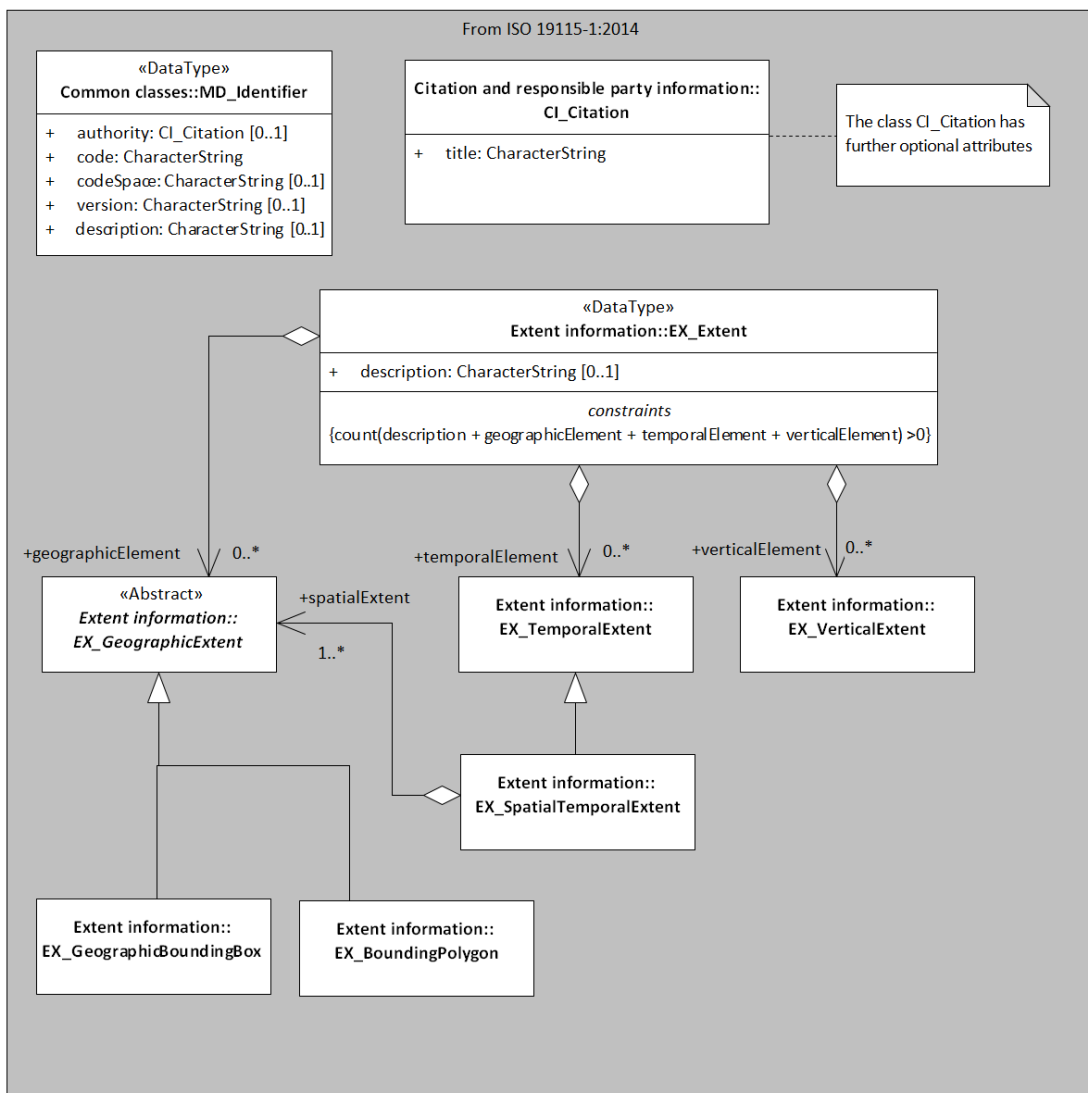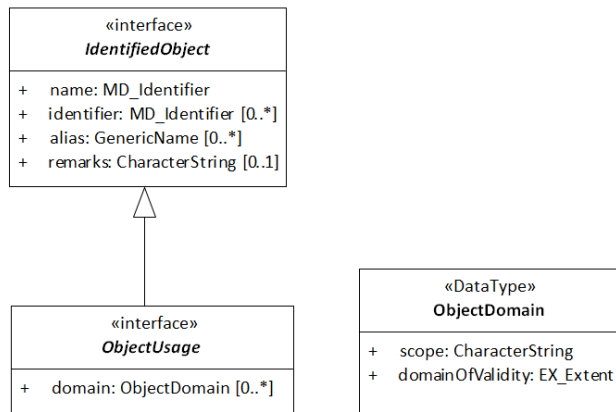
Figure 8 — UML diagram — Data types from ISO 19115-1:2014 (Metadata)

Table 5 — Defining elements of Common Classes::IdentifiedObject class

Definition:	identifications of a CRS-related object				
Stereotype:	Interface				
Class attribute:	Abstract				
Inheritance from:	(none)				
Generalization of:	ObjectUsage, Coordinate Systems::CoordinateSystem, Coordinate Systems:: CoordinateSystemAxis Datums::DefiningParameter, Datums::Ellipsoid, Datums::PrimeMeridian Coordinate Operations::GeneralOperationParameter, Coordinate Operations::OperationMethod				
Public attributes:					
Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Object name	name	MD_Identifier	M	1	primary name by which this object is identified
Object identifier	identifier	MD_Identifier	O	N	identifier which references elsewhere the object's defining information; alternatively an identifier by which this object can be referenced
Object alias	alias	GenericName	O	N	alternative name by which this object is identified
Object remarks	remarks	CharacterString	O	1	comments on or information about this object, including data source information

Table 6 — Defining elements of Common Classes::ObjectUsage class

Definition:	usage of a CRS-related object				
Stereotype:	Interface				
Class attribute:	Abstract				
Inheritance from:	IdentifiedObject				
Generalization of:	Coordinate Reference Systems::CRS, Datums::Datum, Datums::DatumEnsemble, Coordinate Operations::CoordinateOperation				
Public attributes:	4 attributes (name, identifier, alias and remarks) inherited from IdentifiedObject, plus:				
Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Object usage	domain	ObjectDomain	O	N	scope and validity of a CRS-related object

Table 7 — Defining elements of Common Classes::ObjectDomain class

Definition:	scope and validity of a CRS-related object				
Stereotype:	DataType				
Class attribute:	Concrete				
Inheritance from:	(none)				
Used by:	ObjectUsage				
Public attributes:					
Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Object scope	scope	CharacterString	M	1	description of usage, or limitations of usage, for which this object is valid Note: If unknown, enter "not known".
Object validity	domainOfValidity	EX_Extent	M	1	spatial and temporal extent in which this object is valid

9   Coordinate Reference Systems package

9.1   Coordinate reference system

9.1.1   General

In this document, a coordinate reference system (CRS) definition generally consists of two components: one coordinate system (CS, Clause 10) and either one datum or one datum ensemble (Clause 11). Derived coordinate reference systems have a third component: a coordinate conversion (Clause 12). Each of these components has a number of attributes.

A datum (in modern geodesy, a reference frame) specifies the relationship of a coordinate system to an object, thus ensuring that the abstract mathematical concept "coordinate system" can be applied to the practical problem of describing positions of features by means of coordinates. The object will generally, but not necessarily, be the Earth or a feature on the Earth such as a building. For certain coordinate reference systems, the object may be a moving platform such as a car, ship, aircraft or spacecraft.

In this document, the *definition* of a coordinate reference system does not change with time. For coordinate reference systems where the object to which they are related is a moving platform, the transformation from the platform CRS to an Earth-fixed CRS may include a time element. For a dynamic coordinate reference system, locations on or near the surface of the Earth will move (very slowly) within the CRS due to crustal motion or deformation and then the CRS's reference frame definition may include time evolution and/or the CRS may have an associated crustal deformation model.

9.1.2   Principal subtypes of coordinate reference system

The classification criteria for the subtyping of coordinate reference system is firstly by type of datum associated with the coordinate reference system, and, in some cases, secondly by type of coordinate system. The following principal subtypes of coordinate reference system are distinguished:

a)   **Geodetic** – a two- or three-dimensional coordinate reference system used to describe spatial location over the whole Earth or substantial parts of it.

It has one subtype, **geographic**, when its coordinate system type is ellipsoidal.

b)   **Engineering** – a coordinate reference system used locally for which three broad categories are recognised:

1)   coordinate reference systems used to describe spatial location over small areas of the Earth using a flat-Earth approximation of the Earth's surface: corrections for Earth-curvature are not applied. Typical applications are for civil engineering construction and building information management.

NOTE 1 these applications are not restricted to using engineering CRSs: they often utilise projected and sometimes geodetic CRSs.

2)   coordinate reference systems used to describe spatial location on moving objects such as road vehicles, vessels, aircraft or spacecraft.

3)   coordinate reference systems used to describe spatial location internally on an image.

NOTE 2 The CRS internal to the image is not geo-referenced. The image can be georeferenced by relating the engineering CRS to a geodetic or projected CRS through a coordinate transformation. In this document engineering coordinate reference systems for images have continuous axes. Grids based on these CRSs are described in ISO 19123[6].

c) **Vertical** – a one-dimensional coordinate reference system making use of the direction of gravity to define height or depth.

d) **Parametric** – a one-dimensional coordinate reference system that uses a parameter or function as a coordinate.

EXAMPLE   pressure used as a vertical coordinate.

e) **Temporal** – a one-dimensional coordinate reference system that describes time.

These principal subtypes of spatial coordinate reference system are described further in C.2.1.

9.2   Derived coordinate reference system

9.2.1   General

A derived coordinate reference system is defined by applying a coordinate conversion to another pre-existing coordinate reference system which is referred to as the base CRS. The derived CRS inherits its datum (reference frame) or datum ensemble (clause 11) from its base CRS. Consequently most derived CRSs are of the same CRS type as their base CRS. Most derived CRSs have a coordinate system which must be of the same CS type as is allowed for principal CRSs of that CRS type.

EXAMPLE 1   A derived geographic CRS will have an ellipsoidal CS because a geographic CRS must have an ellipsoidal CS.

EXAMPLE 2   A derived parametric CRS will have a parametric CS because a parametric CRS must have a parametric CS.

NOTE   An exception is a CRS derived from a projected CRS - see 9.2.2.

Further information on derived coordinate reference systems is given in C.2.2.2.

9.2.2   Projected coordinate reference system

A projected CRS is a coordinate reference system which is derived from a base geographic CRS by applying the coordinate conversion known as a map projection to latitude and longitude ellipsoidal coordinate values. Projected CRSs are modelled as a special case of derived CRS because of their importance in geographic information. A projected CRS is constrained to have a Cartesian coordinate system. In the 3D case the ellipsoidal height from the base CRS is retained to form a three-dimensional Cartesian coordinate system.

A projected CRS may act as the base CRS for a derived projected CRS. A derived projected CRS is not constrained to have a Cartesian coordinate system: it may have one of several other types of coordinate system.

NOTE   The term 'derived projected CRS' is used for consistency in the UML modelling. A derived projected CRS is not a projected CRS - 'derived from projected CRS' would be a more accurate description. However, in addition to inheriting its datum or reference frame from its base projected CRS, a derived projected CRS inherits the projection distortions of its base projected CRS.

9.3   Compound coordinate reference system

9.3.1   General

A compound coordinate reference system is a non-repeating sequence of two or more coordinate reference systems none of which can itself be compound.

EXAMPLE 1   A projected CRS having easting and northing coordinates with a vertical CRS having a gravity-related height as a coordinate.

EXAMPLE 2   A geographic CRS having latitude and longitude coordinates with a parametric CRS having pressure as a coordinate.

Nesting of compound coordinate reference systems is not be permitted; the individual single systems are aggregated together. Further information on compound coordinate reference system is given in C.2.2.3.

9.3.2   Spatial compound coordinate reference system

For spatial coordinates, a number of constraints exist for the construction of compound CRSs. Coordinate reference systems that are combined are required to not contain any duplicate or redundant axes. Valid combinations shall be the following.

a)   Geographic 2D + Vertical.

b)   Geographic 2D + Engineering 1D (near vertical).

c)   Projected 2D + Vertical.

d)   Projected 2D + Engineering 1D (near vertical).

e)   Engineering (horizontal 2D) + Vertical.

f)   Engineering (1D linear) + Vertical.

9.3.3   Spatio-temporal compound coordinate reference system

Any single spatial coordinate reference system, or any of the combinations of spatial compound coordinate reference systems listed in 9.3.2, may be associated with a temporal coordinate reference system to form a spatio-temporal compound coordinate reference system. More than one temporal coordinate reference system may be included if these axes represent different time quantities: examples are given in E.4.4 and E.4.5.

9.3.4   Spatio-parametric compound coordinate reference system

A spatio-parametric coordinate reference system is a compound CRS in which one component is a geographic 2D, projected 2D or engineering 2D CRS, supplemented by a parametric CRS to create a three-dimensional CRS: an example is included in E.3.3. More than one parametric coordinate reference system may be included if these represent independent parametric quantities.

9.3.5   Spatio-parametric-temporal compound coordinate reference system

Any of the above-listed combinations of spatial, parametric and temporal CRSs may be associated to form a spatio-parametric-temporal compound coordinate reference system.

9.4   UML schema for the Coordinate Reference Systems package

Figure 9 shows the UML class diagram of the Coordinate Reference Systems package. Subtypes of derived CRS are detailed in Figure 10. The definition of the object classes of the package are provided in Tables 8 to 25.

The CRS package UML class diagram shows an association named CoordinateSystem from the SingleCRS class to the CoordinateSystem class. This association is included to indicate that all of the subclasses of SingleCRS have a direct association to CoordinateSystem or one of its subclasses, as later detailed in Clause 10. Constraints on associations between CRSs and coordinate systems are detailed in Clause 10.

The CRS UML class diagram also shows an association named DefiningDatum from the SingleCRS class to the Datum class. This association indicates that many, but not all, of the subclasses of SingleCRS have a direct association to Datum or to one of its subclasses. A single CRS may alternatively be associated with a *datum ensemble* rather than with a datum. Constraints on associations between CRSs and datums or datum ensembles are detailed in Clause 11. Derived CRSs do not use this association to datum or datum ensemble: instead a Derived CRSs inherits its datum or datum ensemble from the base CRS from which it has been derived.

The CRS UML diagram additionally shows an association named Definition from the DerivedCRS class to the Conversion class. This will usually be implemented as a coordinate conversion embedded within the derived CRS definition. A coordinate conversion is a type of coordinate operation. The UML model for coordinate operations is detailed in Clause 12.

Further information on the modelling of CRSs is given in C.2.
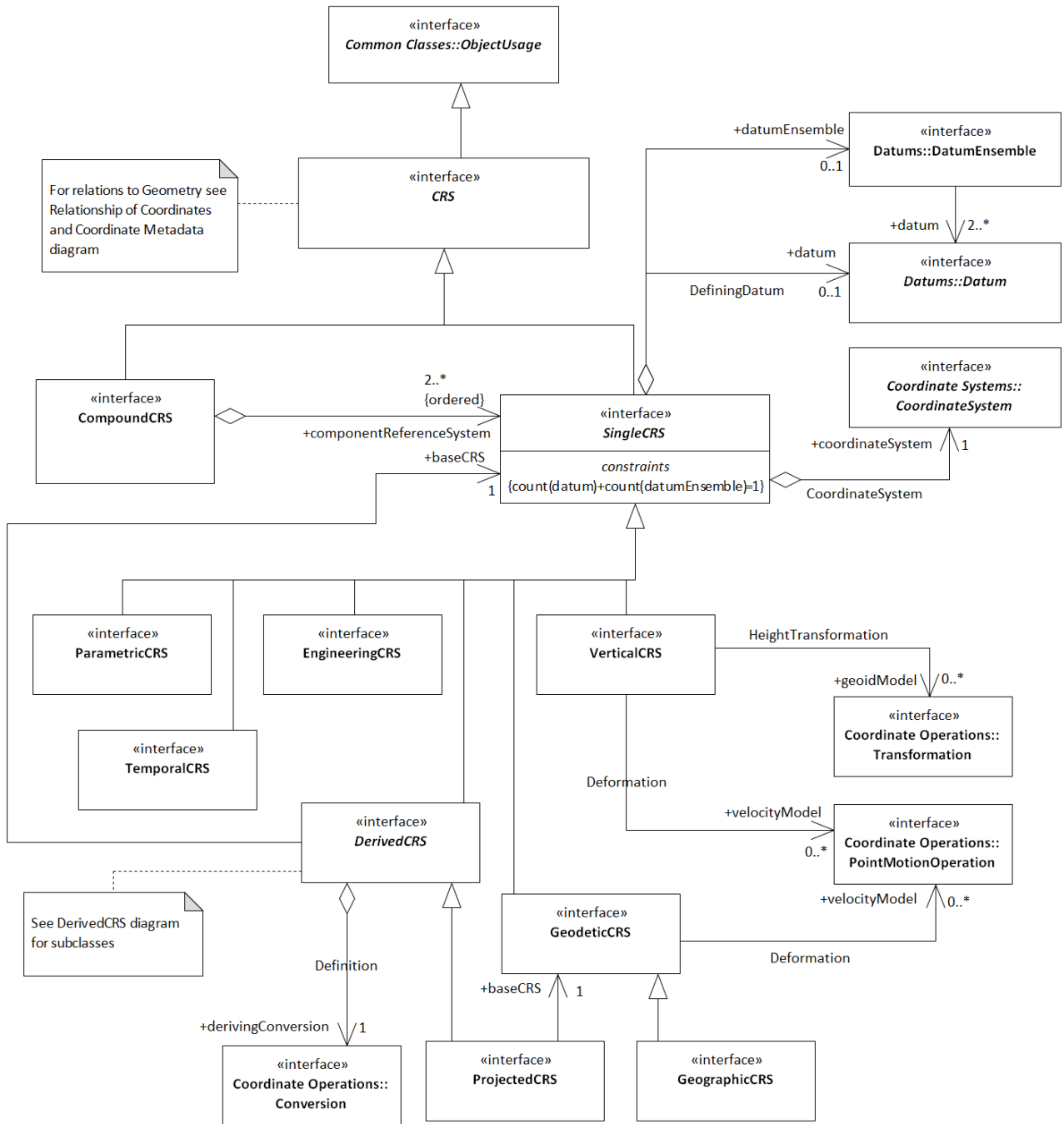
Figure 9 — UML diagram — Coordinate Reference Systems package
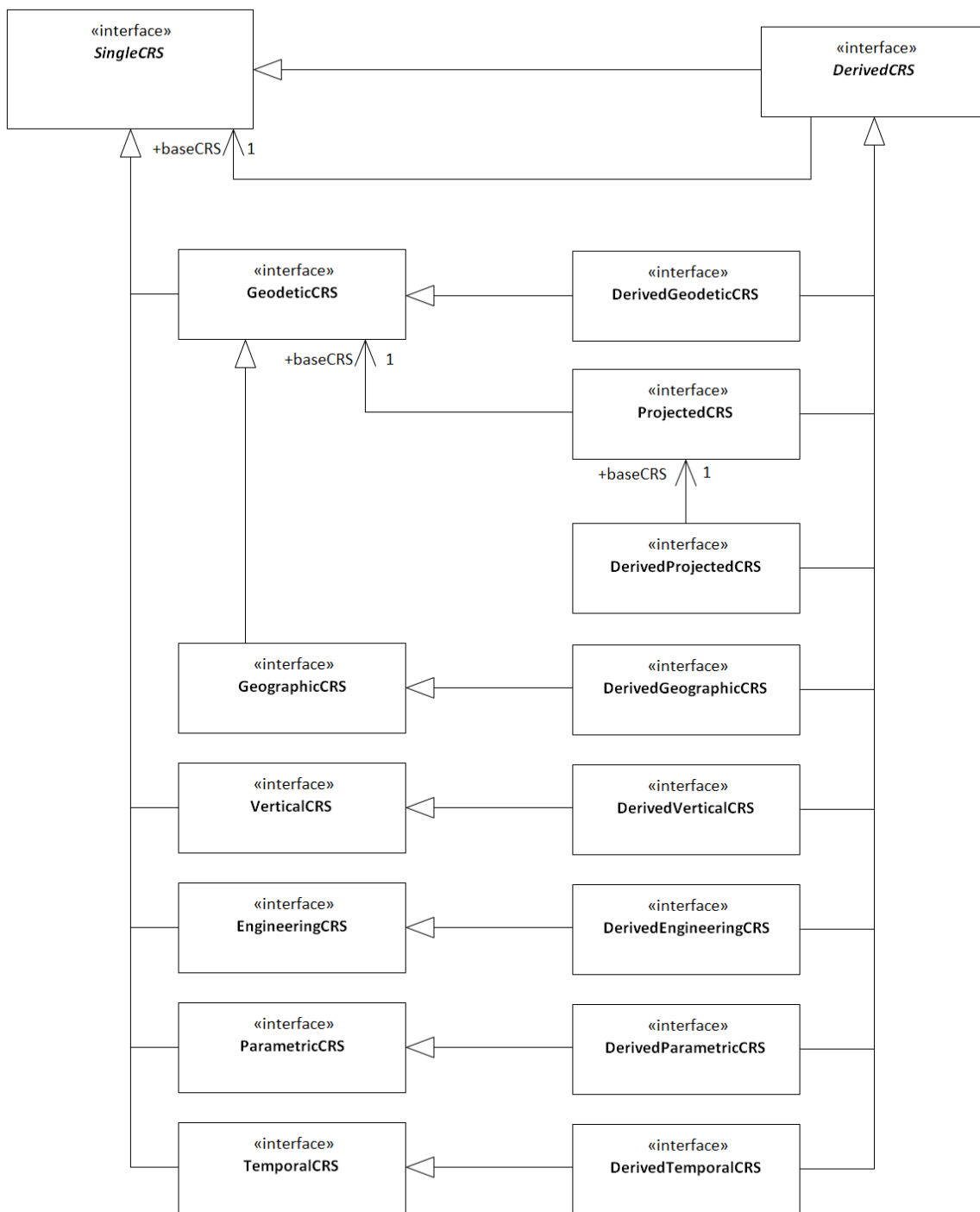
Figure 10 — UML diagram — Derived Coordinate Reference Systems

Table 8 — Defining elements of Coordinate Reference Systems::CRS class

Definition:	coordinate reference system which is usually single but may be compound
Stereotype:	Interface
Class attribute:	Abstract
Inheritance from:	Common Classes::ObjectUsage
Generalization of:	SingleCRS, CompoundCRS
Association roles:	(Note attached to association from Geometry::Geometry: "The crs role is derived from the attribute rsid". See ISO 19107)
Public attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.

Table 9 — Defining elements of Coordinate Reference Systems::SingleCRS class

Definition:	coordinate reference system consisting of one coordinate system and either one datum or one datum ensemble
Stereotype:	Interface
Class attribute:	Abstract
Inheritance from:	CRS
Generalization of:	GeodeticCRS, VerticalCRS, ParametricCRS, EngineeringCRS, TemporalCRS, DerivedCRS

Association roles: associations inherited from CRS, plus:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
Coordinate System	(aggregation) coordinateSystem	CoordinateSystems:: CoordinateSystem	M	1	coordinate system that is a component of this single coordinate reference system
Defining Datum	(aggregation) datum	Datums::Datum	C	1	datum that is a component of this single coordinate reference system
(not named)	(aggregation) datumEnsemble	Datums:: DatumEnsemble	C	1	datum ensemble that is a component of this single coordinate reference system

Constraints:	{count(datum) + count(datumEnsemble) = 1}
Remarks:	The constraint requires a singleCRS to be associated with either a datum (reference frame) or a datum ensemble.
Public attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.

Table 10 — Defining elements of Coordinate Reference Systems::GeodeticCRS class

Definition:	coordinate reference system associated with a geodetic reference frame and a three-dimensional Cartesian or spherical coordinate system Note: If the geodetic reference frame is dynamic then the geodetic CRS is dynamic, else it is static.				
Stereotype:	Interface				
Class attribute:	Concrete				
Inheritance from:	SingleCRS				
Generalization of:	GeographicCRS, DerivedGeodeticCRS				
Association roles:	associations inherited from SingleCRS, plus:				
Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
Coordinate System	(aggregation) coordinateSystem	CoordinateSystems:: GeodeticCS	M	1	coordinate system that is a component of this geodetic coordinate reference system
Defining Datum	(aggregation) datum	Datums:: Geodetic ReferenceFrame	O	1	geodetic reference frame that is a component of this geodetic coordinate reference system
Deformation	velocityModel	CoordinateOperations:: PointMotionOperation	O	N	velocity model(s) or deformation grid(s) that may be applied to this geodetic coordinate reference system
Constraints:	constraints inherited from SingleCRS, plus: {coordinateSystem.ocl As Type(EllipsoidalCS) implies count(datum.ellipsoid)=1				
Remarks:	The constraint enforces the requirement on geographicCRS to be associated with an ellipsoid. It is made through the GeodeticCRS class because GeographicCRS is related to Datum and hence Ellipsoid only through its subtyping from the GeodeticCRS class. GeodeticCRSs should be associated with a Cartesian coordinate system or with a spherical coordinate system.				
Public Attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.				

Table 11 — Defining elements of Coordinate Reference Systems::GeographicCRS class

Definition:	coordinate reference system associated with a geodetic reference frame and a two- or three-dimensional ellipsoidal coordinate system Note: If the geodetic reference frame is dynamic then the geographic CRS is dynamic, else it is static.				
Stereotype:	Interface				
Class attribute:	Concrete				
Inheritance from:	GeodeticCRS				
Generalization of:	DerivedGeographicCRS				
Association roles:	associations inherited from GeodeticCRS, plus:				
Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
Coordinate System	(aggregation) coordinateSystem	CoordinateSystems:: EllipsoidalCS	M	1	ellipsoidal coordinate system that is a component of this geographic coordinate reference system
Constraints:	constraints inherited from GeodeticCRS				
Remarks:	The constraint {coordinateSystem.ocl As Type(EllipsoidalCS) implies count(datum.ellipsoid)=1} which is inherited from geodeticCRS enforces the requirement on GeographicCRS to be associated with an ellipsoid. It is made through the GeodeticCRS class because GeographicCRS is related to Datum and hence Ellipsoid only through its subtyping from the GeodeticCRS class.				
Public Attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.				

Table 12 — Defining elements of Coordinate Reference Systems::VerticalCRS class

Definition:	coordinate reference system having a vertical reference frame and a one-dimensional vertical coordinate system used for recording gravity-related heights or depths; vertical CRSs make use of the direction of gravity to define the concept of height or depth, but the relationship with gravity may not be straightforward.
	Note 1: If the vertical reference frame is dynamic then the vertical CRS is dynamic, else it is static.
	Note 2: Ellipsoidal heights cannot be captured in a vertical coordinate reference system. They exist only as an inseparable part of a 3D coordinate tuple defined in a geographic 3D coordinate reference system.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	SingleCRS
Generalization of:	DerivedVerticalCRS

Association roles: associations inherited from SingleCRS, plus:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
Coordinate System	(aggregation) coordinateSystem	CoordinateSystems:: GeodeticCS	M	1	vertical coordinate system that is a component of this vertical coordinate reference system
Defining Datum	(aggregation) datum	Datums::Geodetic ReferenceFrame	O	1	vertical reference frame that is a component of this vertical coordinate reference system
Height Transformation	geoidModel	from DerivedCRS	O	N	geoid model or height correction model that is associated with this vertical coordinate reference system
Deformation	velocityModel	CoordinateOperations:: PointMotionOperation	O	N	velocity model or deformation grid that is applied to this vertical coordinate reference system

Public Attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage..

Table 13 — Defining elements of Coordinate Reference Systems::ParametricCRS class

Definition:	coordinate reference system having a parametric datum and a one-dimensional parametric coordinate system which uses parameter values or functions
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	SingleCRS
Generalization of:	DerivedParametricCRS

Association roles: associations inherited from SingleCRS, plus:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
Coordinate System	(aggregation) coordinateSystem	CoordinateSystems:: ParametricCS	M	1	parametric coordinate system that is a component of this parametric coordinate reference system
Defining Datum	(aggregation) datum	Datums:: ParametricDatum	O	1	parametric datum that is a component of this parametric coordinate reference system

Public attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and CommonClasses::ObjectUsage.

Table 14 — Defining elements of Coordinate Reference Systems::EngineeringCRS class

Definition:	contextually local coordinate reference system associated with an engineering datum and which is applied either to activities on or near the surface of the Earth without geodetic corrections, or on moving platforms such as road vehicles, vessels, aircraft or spacecraft, or as the internal CRS of an image				
Stereotype:	Interface				
Class attribute:	Concrete				
Inheritance from:	SingleCRS				
Generalization of:	DerivedEngineeringCRS				
Association roles: associations inherited from SingleCRS, plus:					
Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
Coordinate System	(aggregation) coordinateSystem	CoordinateSystems:: EngineeringCS	M	1	coordinate system that is a component of this engineering coordinate reference system
Defining Datum	(aggregation) datum	Datums:: EngineeringDatum	O	1	engineering datum that is a component of this engineering coordinate reference system
Public Attributes: 6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.					

Table 15 — Defining elements of Coordinate Reference Systems::TemporalCRS class

Definition:	coordinate reference system associated with a temporal datum and a one-dimensional temporal coordinate system				
Stereotype:	Interface				
Class attribute:	Concrete				
Inheritance from:	SingleCRS				
Generalization of:	DerivedTemporalCRS				
Association roles: associations inherited from SingleCRS, plus:					
Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
Coordinate System	(aggregation) coordinateSystem	CoordinateSystems:: TemporalCS	M	1	temporal coordinate system that is a component of this temporal coordinate reference system
Defining Datum	(aggregation) datum	Datums:: TemporalDatum	O	1	temporal datum that is a component of this temporal coordinate reference system
Public attributes: 6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.					

Table 16 — Defining elements of Coordinate Reference Systems::DerivedCRS class

Definition:	single coordinate reference system that is defined through the application of a specified coordinate conversion to the definition of a previously established single coordinate reference system referred to as the base CRS Note: A derived coordinate reference system inherits its datum (or datum ensemble) from its base CRS. The coordinate conversion between the base and derived coordinate reference system is implemented using the parameters and formula(s) specified in the definition of the coordinate conversion.
Stereotype:	Interface
Class attribute:	Abstract
Inheritance from:	SingleCRS
Generalization of:	ProjectedCRS, DerivedProjectedCRS, DerivedGeodeticCRS, DerivedGeographicCRS, DerivedVerticalCRS DerivedEngineeringCRS, DerivedParametricCRS, DerivedTemporalCRS

Association roles: associations inherited from SingleCRS, including …
… *(aggregation) coordinateSystem to CoordinateSystems::CoordinateSystem [1], association named CoordinateSystem)*, plus:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	baseCRS	SingleCRS	M	1	coordinate reference system that is the baseCRS for this derived coordinate reference system
Definition	(aggregation) derivingConversion	CoordinateOperations:: Conversion	M	1	conversion that is a component of this derived coordinate reference system

Constraints:	{count(baseCRS.datum)=1 implies datum=baseCRS.datum} {count(baseCRS.datumEnsemble)=1 implies datumEnsemble=baseCRS.datum}
Remarks:	The constraints require the derived CRS to take the datum or datum ensemble (whichever one is applicable) of its base CRS.
Public attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from CommonClasses::IdentifiedObject and CommonClasses::ObjectUsage.

Table 17 — Defining elements of Coordinate Reference Systems::ProjectedCRS class

Definition:	derived coordinate reference system which has a geodetic (usually geographic) coordinate reference system as its base CRS, thereby inheriting a geodetic reference frame, is converted using a map projection, and has a Cartesian coordinate system, usually two-dimensional but may be three-dimensional Note: In the 3D case the base geographic CRSs ellipsoidal height is passed through unchanged and forms the vertical axis of the projected CRS's Cartesian coordinate system.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	DerivedCRS

Association roles: associations inherited from DerivedCRS, plus:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	baseCRS	GeodeticCRS	M	1	geodetic or geographic coordinate reference system that is the baseCRS for this projected coordinate reference system
Coordinate System	(aggregation) coordinateSystem	CoordinateSystems:: CartesianCS	M	1	Cartesian coordinate system that is a component of this projected coordinate reference system

Public Attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.

Table 18 — Defining elements of Coordinate Reference Systems::DerivedProjectedCRS class

Definition:	derived coordinate reference system which has a projected coordinate reference system as its base CRS, thereby inheriting a geodetic reference frame, but also inheriting the distortion characteristics of the base projected CRS
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	DerivedCRS

Association roles: associations inherited from DerivedCRS, plus:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	baseCRS	ProjectedCRS	M	1	projected coordinate reference system that is the baseCRS for this derived projected coordinate reference system
Coordinate System	(aggregation) coordinateSystem	CoordinateSystems:: DerivedProjectedCS	M	1	coordinate system that is a component of this derived projected coordinate reference system

Public Attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.

Table 19 — Defining elements of Coordinate Reference Systems::DerivedGeodeticCRS class

Definition:	derived coordinate reference system which has either a geodetic or a geographic coordinate reference system as its base CRS, thereby inheriting a geodetic reference frame, and associated with a 3D Cartesian or spherical coordinate system
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	GeodeticCRS DerivedCRS
Public Attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.

Table 20 — Defining elements of Coordinate Reference Systems::DerivedGeographicCRS class

Definition:	derived coordinate reference system which has either a geodetic or a geographic coordinate reference system as its base CRS, thereby inheriting a geodetic reference frame, and an ellipsoidal coordinate system Note: A derived geographic CRS can be based on a geodetic CRS only if that geodetic CRS definition includes an ellipsoid.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	GeographicCRS DerivedCRS Note: Constraints inherited through GeographicCRS include: Ellipsoid is mandatory.
Public Attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.

Table 21 — Defining elements of Coordinate Reference Systems::DerivedVerticalCRS class

Definition:	derived coordinate reference system which has a vertical coordinate reference system as its base CRS, thereby inheriting a vertical reference frame, and a vertical coordinate system
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	VerticalCRS DerivedCRS
Public Attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.

Table 22 — Defining elements of Coordinate Reference Systems::DerivedParametricCRS class

Definition:	derived coordinate reference system which has a parametric coordinate reference system as its base CRS, thereby inheriting a parametric datum, and a parametric coordinate system
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	ParametricCRS DerivedCRS
Public Attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.

Table 23 — Defining elements of Coordinate Reference Systems::DerivedEngineeringCRS class

Definition:	derived coordinate reference system which has an engineering coordinate reference system as its base CRS, thereby inheriting an engineering datum, and is associated with one of the coordinate system types within the engineeringCS class
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	EngineeringCRS DerivedCRS
Public Attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.

Table 24 — Defining elements of Coordinate Reference Systems::DerivedTemporalCRS class

Definition:	derived coordinate reference system which has a temporal coordinate reference system as its base CRS, thereby inheriting a temporal datum, and is associated with a temporal coordinate system
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	TemporalCRS DerivedCRS
Public Attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.

Table 25 — Defining elements of Coordinate Reference Systems::CompoundCRS class

Definition:	coordinate reference system describing the position of points through two or more independent single coordinate reference systems Note: two coordinate reference systems are independent of each other if coordinate values in one cannot be converted or transformed into coordinate values in the other.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	CRS

Association roles: associations inherited from CRS, plus

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	(aggregation) componentReference System	SingleCRS (ordered)	M (minimum 2)	N	coordinate reference system that is a component of this compound coordinate reference system

Public Attributes:	6 attributes (CRS name, CRS alias, CRS identifier, CRS scope, CRS validity and CRS remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage.

10 Coordinate Systems package

10.1 Coordinate system - General

In this document, the Coordinate Systems package models two main concepts: coordinate system and coordinate system axis. A coordinate system has to be composed of a non-repeating sequence of coordinate system axes. One coordinate system may be used by multiple coordinate reference systems. The dimensions of the coordinate space, the names, the units of measure, the directions and sequence of the axes all form part of the coordinate system definition. The number of axes is required to be equal to the dimensions of the space. The number of coordinates in a coordinate tuple is required to be equal to the number of coordinate axes in the coordinate system. Coordinates in coordinate tuples are required to be supplied in the order in which the coordinate system's axes are defined.

In this document, coordinate systems are divided into subtypes by the geometric properties of the coordinate space spanned and the geometric properties of the axes themselves (straight or curved; perpendicular or not). Certain subtypes of coordinate system are required to be used only with specific subtypes of coordinate reference system as shown in Table 26 and figures 12 and 13.

Coordinate systems are described further in C.3.

10.2 Parametric coordinate system

A coordinate system is of type parametric if a physical or material property or function is used as the dimension. The parameter can be measured or could be a function defined in other contexts, but in parametric coordinate systems it forms the coordinate system axis.

EXAMPLE 1   Pressure in meteorological applications.

EXAMPLE 2   Density (isopycnals) in oceanographic applications.

A parametric coordinate system is required to be one-dimensional and have one axis.

10.3 Temporal coordinate system

This document supports three forms of temporal coordinate system:

— DateTimeTemporalCS: coordinate values are dateTimes in the proleptic Gregorian calendar as described in ISO 8601.

— TemporalCountCS: coordinate values are integer numbers having units, they are counts of a temporal quantity.

— TemporalMeasureCS: coordinate values are real numbers having units, they are measures of a temporal quantity.

A temporal coordinate system is required to be one-dimensional and to have one axis. Further information is provided in Annex D.

Table 26 — Subtypes of coordinate system and constraints in its relationship with coordinate reference system

CS subtype	Description	Used with CRS type(s)
affine	two- or three-dimensional coordinate system in Euclidean space with straight axes that are not necessarily orthogonal.	engineering derivedEngineering derivedProjected
Cartesian	two- or three-dimensional coordinate system in Euclidean space which gives the position of points relative to orthogonal straight axes. All axes are required to have the same unit of measure.	geodetic projected engineering derivedGeodetic derivedProjected derivedEngineering
cylindrical	three-dimensional coordinate system in Euclidean space consisting of a polar coordinate system extended by a straight coordinate axis perpendicular to the plane spanned by the polar coordinate system.	engineering derivedEngineering derivedProjected
ellipsoidal	two- or three-dimensional coordinate system in which position is specified by geodetic latitude, geodetic longitude and (in the three-dimensional case) ellipsoidal height.	geographic derivedGeographic
linear	one-dimensional coordinate system that consists of the points that lie on the single axis described. Example: usage of the line feature representing a pipeline to describe points on or along that pipeline. This document only lends itself to be used for simple (=continuous) linear systems. For a more extensive treatment of the subject, particularly as applied to the transportation industry, refer to ISO 19148[6].	engineering derivedEngineering
ordinal	an n-dimensional coordinate system using integer indexing.	engineering derivedEngineering derivedProjected
parametric	one-dimensional coordinate system where the axis units are parameter values which are not inherently spatial.	parametric derivedParametric
polar	two-dimensional coordinate system in Euclidean space in which position is specified by distance from the origin and the angle between the line from origin to point and a reference direction.	engineering derivedEngineering derivedProjected
spherical	three-dimensional coordinate system in Euclidean space with one distance, measured from the origin, and two angular coordinates. Note: not to be confused with an ellipsoidal coordinate system based on an ellipsoid 'degenerated' into a sphere.	geodetic engineering derivedGeodetic derivedEngineering derivedProjected
temporal	one-dimensional coordinate system where the axis is time.	temporal derivedTemporal
vertical	one-dimensional coordinate system used to record the heights (or depths) of points dependent on the Earth's gravity field. An exact definition is deliberately not provided as the complexities of the subject fall outside the scope of this document.	vertical derivedVertical

10.4 Coordinate system axis

A coordinate system is composed of a non-repeating sequence of coordinate system axes. Each of its axes is completely characterized by a unique combination of axis name, axis abbreviation, axis direction and axis unit.

Aliases for these attributes may be used as described in Clause 7.

EXAMPLE 1   The combination {Latitude, φ, north, degree} would lead to one instance of the object class "coordinate system axis"; the combination {Latitude, φ, north, radian} to another instance, the axis unit being different.

EXAMPLE 2   The combination {Easting, E, east, metre} would lead to one instance of the object class "coordinate system axis"; the combination {Easting, X, east, metre} to another instance, the axis abbreviation being different.

In this document, usage of coordinate system axis names is constrained by geodetic custom, depending on the coordinate reference system type. These constraints are shown in Table 27. This constraint is required to work in two directions.

EXAMPLE 3   As "geodetic latitude" and "geodetic longitude" are used as names for coordinate axes forming a geographic coordinate reference system, these terms cannot be used in another context.

Aliases for these constrained names are permitted.

Table 27 — Naming constraints for coordinate system axis

CS type	When used in CRS type	Permitted coordinate system axis names
Cartesian	geodetic	geocentric X, geocentric Y, geocentric Z
Cartesian	projected	northing or southing, easting or westing, [ellipsoidal height (if 3D)]
ellipsoidal	geographic	geodetic latitude, geodetic longitude, [ellipsoidal height (if 3D)]
spherical	geodetic	spherical latitude, spherical longitude, geocentric radius or geocentric latitude, geodetic longitude, geocentric radius
vertical	vertical	depth or gravity-related height

Parametric, temporal and engineering coordinate reference systems may make use of names specific to the local context or custom.

Coordinate system axis is described further in C.3.3.

10.5 UML schema for the Coordinate Systems package

Figure 11 shows the UML class diagram of the Coordinate Systems package. There are restrictions on the associations between Coordinate Reference System subtypes and Coordinate System subtypes which are shown in the UML class diagram in Figure 12. The definitions of the object classes of the Coordinate System package are provided in Tables 28 to 49.
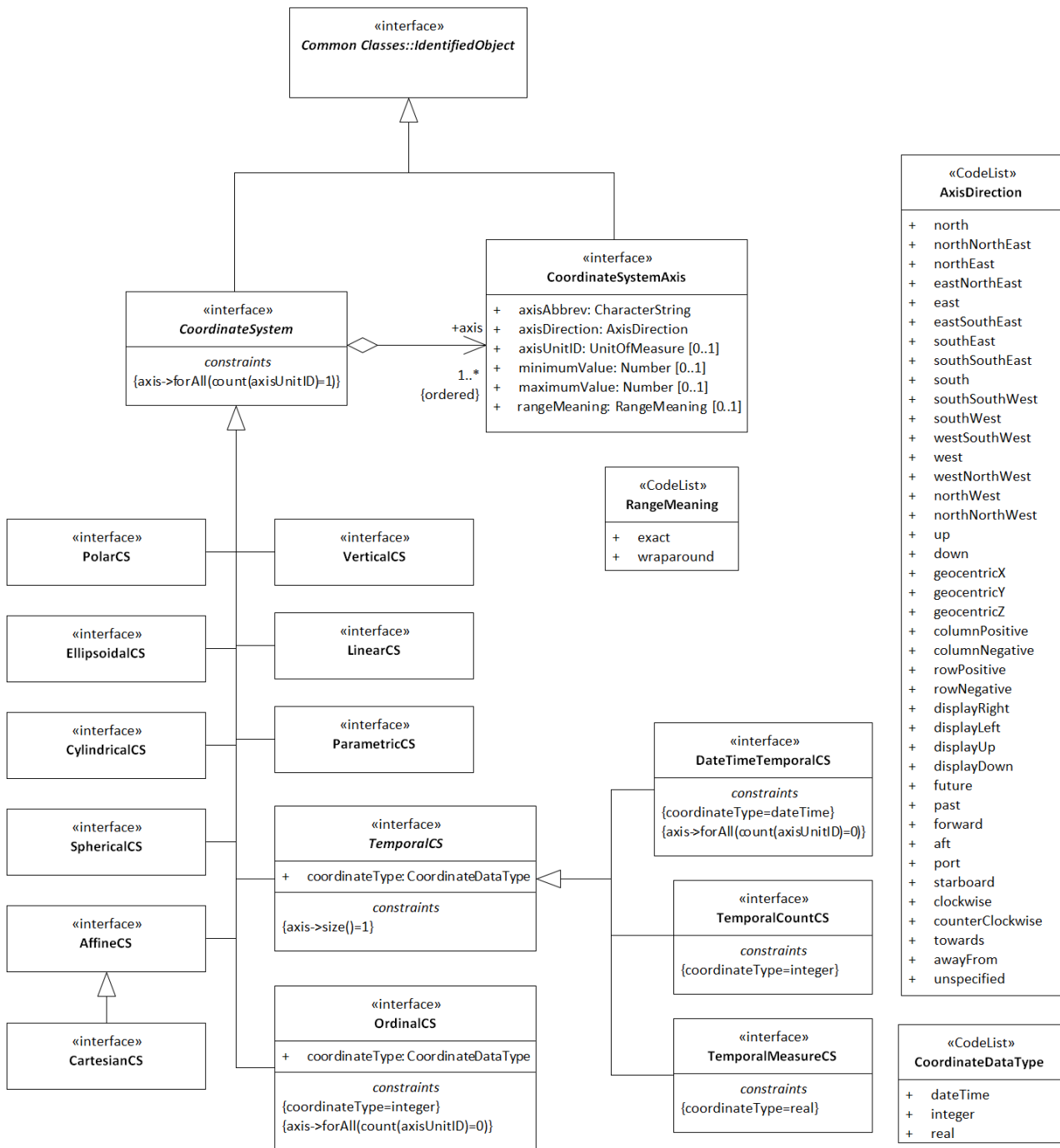
Figure 11 — UML diagram — Coordinate Systems package
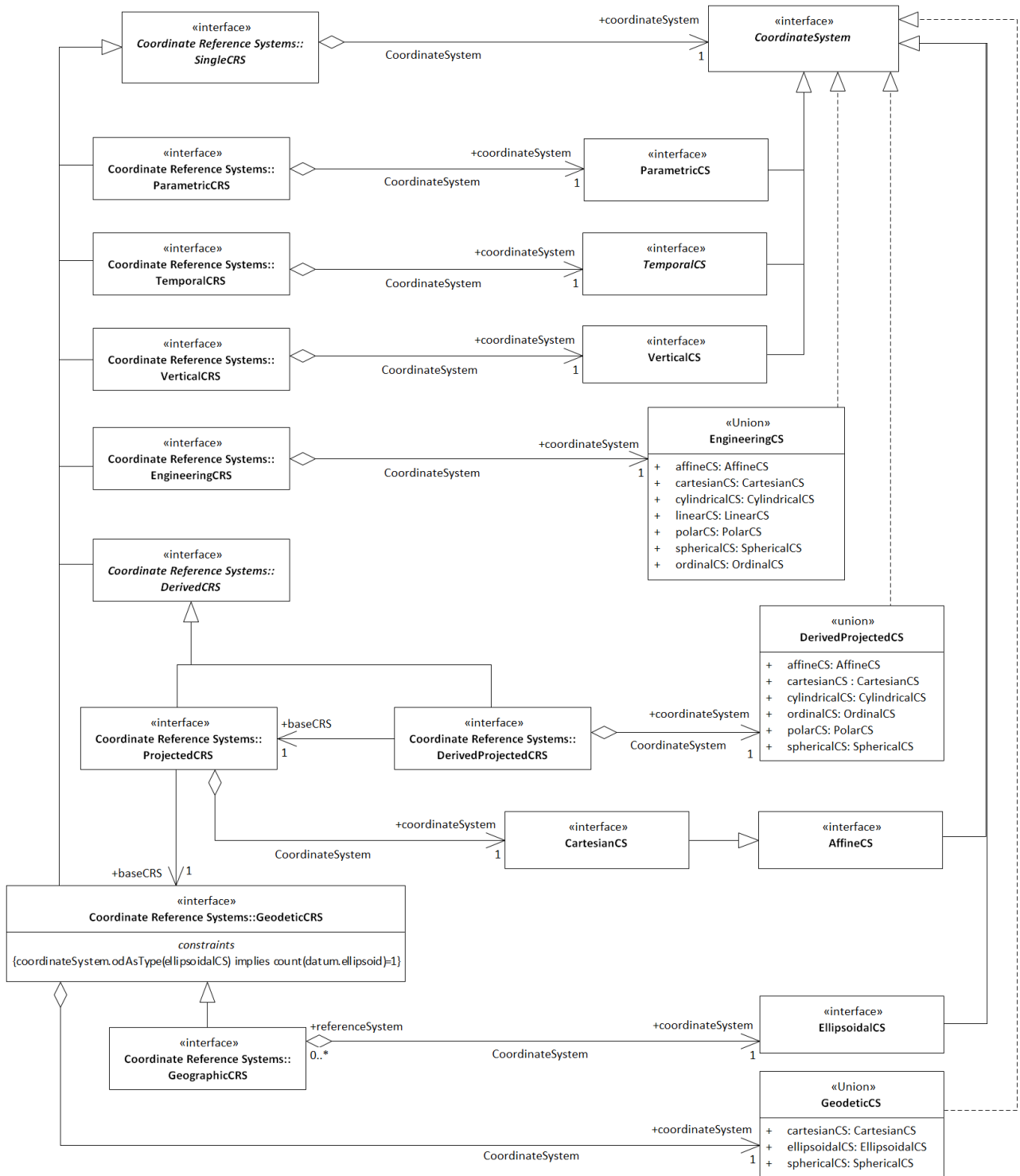
Figure 12 — UML diagram — Coordinate System type associations with Coordinate Reference System type

Table 28 — Defining elements of CoordinateSystems::CoordinateSystem class

Definition:	non-repeating sequence of coordinate system axes that spans a given coordinate space Note: A coordinate system is derived from a set of mathematical rules for specifying how coordinates in a given space are to be assigned to points. The coordinate values in a coordinate tuple shall be recorded in the order in which the coordinate system axes associations are recorded.
Stereotype:	Interface
Class attribute:	Abstract
Inheritance from:	Common Classes::IdentifiedObject
Generalization of:	AffineCS, CylindricalCS, EllipsoidalCS, LinearCS, OrdinalCS, ParametricCS, PolarCS, SphericalCS, TemporalCS, VerticalCS, DerivedProjectedCS, EngineeringCS, GeodeticCS

Association roles:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	(aggregation) axis	SingleCRS (ordered)	M	N	coordinate system axis that is a component of this coordinate system

Constraints: Remarks:	{axis->forAll(count(axis.axisUnitID)=1)} This constraint requies all axes to include unit information. The constraint is modified by the ordinalCS and dateTimeTemporalCS classes.
Public attributes:	4 attributes (CS name, CS alias, CS identifier and CS remarks) inherited from Common Classes::IdentifiedObject.

Table 29 — Defining elements of CoordinateSystems::AffineCS class

Definition:	two- or three-dimensional coordinate system in Euclidean space with straight axes that are not necessarily orthogonal Note: The number of associations shall equal the dimension of the coordinate system.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	CoordinateSystem
Generalization of:	Cartesian CS
Used by:	DerivedProjectedCS EngineeringCS
Public attributes:	4 attributes (CS name, CS alias, CS identifier and CS remarks) inherited from Common Classes::IdentifiedObject.

Table 30 — Defining elements of CoordinateSystems::CartesianCS class

Definition:	two- or three-dimensional coordinate system in Euclidean space with orthogonal straight axes Note: All axes shall have the same length unit. A CartesianCS shall have two or three axis associations; the number of associations shall equal the dimension of the coordinate system.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	AffineCS
Used by:	DerivedProjectedCS EngineeringCS GeodeticCS
Public attributes:	4 attributes (CS name, CS alias, CS identifier and CS remarks) inherited from Common Classes::IdentifiedObject.

Table 31 — Defining elements of CoordinateSystems::CylindricalCS class

Definition:	three-dimensional coordinate system in Euclidean space consisting of a polar coordinate system extended by a straight coordinate axis perpendicular to the plane spanned by the polar coordinate system Note: A CylindricalCS shall have three axis associations.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	CoordinateSystem
Used by:	DerivedProjectedCS EngineeringCS
Public attributes:	4 attributes (CS name, CS alias, CS identifier and CS remarks) inherited from Common Classes::IdentifiedObject.

Table 32 — Defining elements of CoordinateSystems::EllipsoidalCS class

Definition:	two- or three-dimensional coordinate system in which position is specified by geodetic latitude, geodetic longitude, and (in the three-dimensional case) ellipsoidal height Note: An EllipsoidalCS shall have two or three associations; the number of associations shall equal the dimension of the coordinate system.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	CoordinateSystem
Used by:	GeodeticCS
Public attributes:	4 attributes (CS name, CS alias, CS identifier and CS remarks) inherited from Common Classes::IdentifiedObject.

Table 33 — Defining elements of CoordinateSystems::LinearCS class

Definition:	one-dimensional coordinate system that consists of the points that lie on the single axis described Note: The associated coordinate is the distance – with or without offset – from the origin point, specified through the datum definition, to the point along the axis. Example: usage of the line feature representing a pipeline to describe points on or along that pipeline. A LinearCS shall have one axis association.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	CoordinateSystem
Used by:	EngineeringCS
Public attributes:	4 attributes (CS name, CS alias, CS identifier and CS remarks) inherited from Common Classes::IdentifiedObject.

Table 34 — Defining elements of CoordinateSystems::OrdinalCS class

Definition:	n-dimensional coordinate system in which every axis uses integers Note: The number of associations shall be equal the dimension of the coordinate system.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	CoordinateSystem
Used by:	EngineeringCS DerivedProjectedCS
Constraints:	{coordinateType=integer} {axis->forAll(count(axis.axisUnitID)=0)}
Remarks:	Coordinates in an OrdinalCS are sequential counts. The constraints require that coordinates referenced to an ordinal coordinate system have coordinate values with a dataType of integer. No units are required.

Public attributes: 4 attributes (CS name, CS alias, CS identifier and CS remarks) inherited from Common Classes::IdentifiedObject, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Coordinate data type	coordinateType	CoordinateDataType	M	1	datatype of coordinate values

Table 35 — Defining elements of CoordinateSystems::ParametricCS class

Definition:	one-dimensional coordinate reference system which uses parameter values or functions that may vary monotonically with height Note: A ParametricCS shall have one axis association.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	CoordinateSystem
Public attributes:	4 attributes (CS name, CS alias, CS identifier and CS remarks) inherited from Common Classes::IdentifiedObject.

Table 36 — Defining elements of CoordinateSystems::PolarCS class

Definition:	two-dimensional coordinate system in Euclidean space in which position is specified by the distance from the origin and the angle between the line from the origin to a point and a reference direction Note: A PolarCS shall have two axis associations.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	CoordinateSystem
Used by:	DerivedProjectedCS EngineeringCS
Public attributes:	4 attributes (CS name, CS alias, CS identifier and CS remarks) inherited from Common Classes::IdentifiedObject.

Table 37 — Defining elements of CoordinateSystems::SphericalCS class

Definition:	three-dimensional coordinate system in Euclidean space with one distance measured from the origin and two angular coordinates Note: Not to be confused with an ellipsoidal coordinate system based on an ellipsoid "degenerated" into a sphere. A SphericalCS shall have three axis associations.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	CoordinateSystem
Used by:	DerivedProjectedCS EngineeringCS GeodeticCS
Public attributes:	4 attributes (CS name, CS alias, CS identifier and CS remarks) inherited from Common Classes::IdentifiedObject.

Table 38 — Defining elements of CoordinateSystems::TemporalCS class

Definition:	one-dimensional coordinate system used to record time Note: A TemporalCS shall have one axis association.				
Stereotype:	Interface				
Class attribute:	Abstract				
Inheritance from:	CoordinateSystem				
Generalization of:	DateTimeTemporalCS TemporalCountCS TemporalMeasureCS				
Constraints: Remarks:	{axis -> size()=1} The constraint enforces the constraint on CoordinateSystem onto the subtypes of TemporalCS (but this is overridden in the case of DateTimeTemporalCS).				
Public attributes:	4 attributes (CS name, CS alias, CS identifier and CS remarks) inherited from CommonClasses::IdentifiedObject, plus.				
Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Coordinate data type	coordinateType	CoordinateDataType	M	1	datatype of coordinate values

Table 39 — Defining elements of CoordinateSystems::DateTimeTemporalCS class

Definition:	one-dimensional coordinate system used to record time in dateTime representation as defined in ISO 8601. Note: A DateTimeTemporalCS shall have one axis association. It does not use axisUnitID; the temporal quantities are defined through the ISO 8601 representation.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	TemporalCS
Constraints: Remarks:	{coordinateType=dateTime} {count(axis.axisUnitID)=0} The constraints require that coordinates referenced to a dateTime temporal coordinate system have coordinate values with a dataType of dateTime. Units are implied by the dateTime representation.
Public attributes:	5 attributes (CS name, CS alias, CS identifier, CS remarks and coordinateType) inherited from CommonClasses::IdentifiedObject and TemporalCS, one of which is constrained.

Table 40 — Defining elements of CoordinateSystems::TemporalCountCS class

Definition:	one-dimensional coordinate system used to record time as an integer count Note: A TemporalCountCS shall have one axis association.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	TemporalCS
Constraints:	{coordinateType=integer}
Public attributes:	5 attributes (CS name, CS alias, CS identifier, CS remarks and coordinateType) inherited from CommonClasses::IdentifiedObject and TemporalCS, one of which is constrained.

Table 41 — Defining elements of CoordinateSystems::TemporalMeasureCS class

Definition:	one-dimensional coordinate system used to record a time as a real number Note: A TemporalMeasureCS shall have one axis association.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	TemporalCS
Constraints:	{coordinateType=real}
Public attributes:	5 attributes (CS name, CS alias, CS identifier, CS remarks and coordinateType) inherited from CommonClasses::IdentifiedObject and TemporalCS, one of which is constrained.

Table 42 — Defining elements of CoordinateSystems::VerticalCS class

Definition:	one-dimensional coordinate system used to record the heights or depths of points, usually dependent on the Earth's gravity field Note: An exact definition is deliberately not provided as the complexities of the subject fall outside the scope of this document. A VerticalCS shall have one axis association.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	CoordinateSystem
Public attributes:	4 attributes (CS name, CS alias, CS identifier and CS remarks) inherited from Common Classes::IdentifiedObject.

Table 43 — Defining elements of CoordinateSystems::DerivedProjectedCS class

Definition:	coordinate system used by a DerivedProjected CRS, one of an affine coordinate system, a Cartesian coordinate system, a cylindrical coordinate system, an ordinal coordinate system, a polar coordinate system or a spherical coordinate system
Stereotype:	Union
Realization of:	CoordinateSystem. As such, it must implement all inherited operations and associations. Furthermore, it must support all inherited attributes, at least as "read only".
Association roles:	associations inherited from CoordinateSystem, plus: (aggregation) affineCS to AffineCS [1] (aggregation) cartesianCS to CartesianCS [1] (aggregation) cylindricalCS to CylindricalCS [1] (aggregation) ordinalCS to OrdinalCS [1] (aggregation) polarCS to PolarCS [1] (aggregation) sphericalCS to SphericalCS [1] union (one of) constraint on affineCS, cartesianCS, cylindricalCS, ordinalCS, polarCS and sphericalCS associations
Public attributes:	(none)

Table 44 — Defining elements of CoordinateSystems::EngineeringCS class

Definition:	coordinate system used by an engineering coordinate reference system, one of an affine coordinate system, a Cartesian coordinate system, a cylindrical coordinate system, a linear coordinate sytem, an ordinal coordinate system, a polar coordinate system or a spherical coordinate system
Stereotype:	Union
Realization of:	CoordinateSystem. As such, it must implement all inherited operations and associations. Furthermore, it must support all inherited attributes, at least as "read only".
Association roles:	associations inherited from CoordinateSystem, plus: (aggregation) affineCS to AffineCS [1] (aggregation) cartesianCS to CartesianCS [1] (aggregation) cylindricalCS to CylindricalCS [1] (aggregation) linearCS to LinearCS [1] (aggregation) ordinalCS to OrdinalCS [1] (aggregation) polarCS to PolarCS [1] (aggregation) sphericalCS to SphericalCS [1] union (one of) constraint on affineCS, cartesianCS, cylindricalCS, linearCS, ordinalCS, polarCS and sphericalCS associations
Public attributes:	(none)

Table 45 — Defining elements of CoordinateSystems::GeodeticCS class

Definition:	coordinate system used by a Geodetic CRS, one of a Cartesian coordinate system or a spherical coordinate system
Stereotype:	Union
Realization of:	CoordinateSystem. As such, it must implement all inherited operations and associations. Furthermore, it must support all inherited attributes, at least as "read only".
Association roles:	associations inherited from CoordinateSystem, plus: (aggregation) cartesianCS to CartesianCS [1] (aggregation) ellipsoidalCS to EllipsoidalCS [1] *(see remarks)* (aggregation) sphericalCS to SphericalCS [1] union (one of) constraint on affineCS, cartesianCS, cylindricalCS, linearCS, ordinalCS, polarCS and sphericalCS associations
Remarks:	ellipsoidalCS is included in the GeodeticCS class so that it may be used by the GeographicCRS subtype of Geodetic CRS. GeodeticCRSs should use only CartesianCS or sphericalCS.
Public attributes:	(none)

Table 46 — Defining elements of CoordinateSystems::CoordinateDataType class

Definition:	datatype of coordinate values
Stereotype:	CodeList
Inheritance from:	(none)
Used by:	OrdinalCoordinateSystem TemporalCoordinateSystem

Public attributes:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Integer	integer	integer	C	1	quantity expressed as a count used for a temporal or ordinal coordinate system axis.
Real	real	measure	C	1	quantity expressed as a measure used for a temporal coordinate system axis
DateTime	dateTime	dateTime	C	1	compound quantity representable as a character string conformant with ISO 8601 used for a temporal coordinate system axis

Condition: One and only one of the listed attributes shall be supplied.

Table 47 — Defining elements of CoordinateSystems::CoordinateSystemAxis class

Definition:	definition of a coordinate system axis
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	Common Classes::IdentifiedObject

Public attributes:	4 attributes (CS axis name, CS axis alias, CS axis identifier and CS axis remarks) inherited from Common Classes::IdentifiedObject, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Coordinate system axis abbreviation	axisAbbrev	CharacterString	M	1	abbreviation used for this coordinate system axis; this abbreviation is also used to identify the coordinates in the coordinate tuple Examples: *X* and *Y*. Note: when the standard symbol is a Greek character (see 3.2) the abbreviation may differ from the symbol (for example to constrain symbols to other character sets).
Coordinate system axis direction	axisDirection	AxisDirection	M	1	direction of this coordinate system axis (or in the case of Cartesian projected coordinates, the direction of this coordinate system axis locally) Examples: north or south, east or west, up or down. Note: Within any set of coordinate system axes, only one of each pair of terms can be used. For Earth-fixed CRSs, this direction is often approximate and intended to provide a human interpretable meaning to the axis. When a geodetic reference frame is used, the precise directions of the axes may therefore vary slightly from this approximate direction. Note that an EngineeringCRS often requires specific descriptions of the directions of its coordinate system axes.
Coordinate system axis unit	axisUnitID	UnitOfMeasure	C	0..1	spatial unit or temporal quantity used for this coordinate system axis Note: The value of a coordinate in a coordinate tuple shall be recorded using this unit. This element shall be omitted if this axis is part of a DateTimeTemporalCS or an OrdinalCS, and be provided in all other cases.
Coordinate system axis minimum value	minimumValue	Number	O	1	minimum value normally allowed for this axis, in the unit for the axis
Coordinate system axis maximum value	maximumValue	Number	O	1	maximum value normally allowed for this axis, in the unit for the axis
Coordinate system axis range meaning	rangeMeaning	RangeMeaning	C	1	meaning of axis value range specified by minimumValue and maximumValue Note: This element shall be omitted when both minimumValue and maximumValue are omitted. It may be included when minimumValue and/or maximumValue are included. If this element is omitted when minimumValue or maximumValue are included, the meaning is unspecified.

Table 48 — Defining elements of CoordinateSystems::AxisDirection class

Definition:	direction of positive increase in the coordinate value for a coordinate system axis
Stereotype:	CodeList
Derived from:	(none)
Used by:	CoordinateSystemAxis

Public attributes:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
north	north	CharacterString	C	1	axis positive direction is north Note: In a geodetic, geographic or projected CRS, north is defined through the geodetic reference frame. In an engineering CRS, north may be defined with respect to an engineering object rather than a geographical direction.
north-north-east	northNorthEast	CharacterString	C	1	axis positive direction is approximately north-north-east
north-east	northEast	CharacterString	C	1	axis positive direction is approximately north-east
east-north-east	eastNorthEast	CharacterString	C	1	axis positive direction is approximately east-north-east
east	east	CharacterString	C	1	axis positive direction is $\pi/2$ radians clockwise from north
east-south-east	eastSouthEast	CharacterString	C	1	axis positive direction is approximately east-south-east
south-east	southEast	CharacterString	C	1	axis positive direction is approximately south-east
south-south-east	southSouthEast	CharacterString	C	1	axis positive direction is approximately south-south-east
south	south	CharacterString	C	1	axis positive direction is π radians clockwise from north
south-south-west	southSouthWest	CharacterString	C	1	axis positive direction is approximately south-south-west
south-west	southWest	CharacterString	C	1	axis positive direction is approximately south-west
west-south-west	westSouthWest	CharacterString	C	1	axis positive direction is approximately west-south-west
west	west	CharacterString	C	1	axis positive direction is $3\pi/2$ radians clockwise from north
west-north-west	westNorthWest	CharacterString	C	1	axis positive direction is approximately west-north-west
north-west	northWest	CharacterString	C	1	axis positive direction is approximately north-west
north-north-west	northNorthWest	CharacterString	C	1	axis positive direction is approximately north-north-west
up	up	CharacterString	C	1	axis positive direction is up relative to gravity
down	down	CharacterString	C	1	axis positive direction is down relative to gravity
Geocentric X	geocentricX	CharacterString	C	1	axis positive direction is in the equatorial plane from the centre of the modelled Earth towards the intersection of the equator with the prime meridian
Geocentric Y	geocentricY	CharacterString	C	1	axis positive direction is in the equatorial plane from the centre of the modelled Earth towards the intersection of the equator and the meridian $\pi/2$ radians eastwards from the prime meridian
Geocentric Z	geocentricZ	CharacterString	C	1	axis positive direction is from the centre of the modelled Earth parallel to its rotation axis and towards its north pole

(Public attributes continued on next page)

Table 48 (continued)

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
column-positive	columnPositive	CharacterString	C	1	axis positive direction is towards higher pixel column
column-negative	columnNegative	CharacterString	C	1	axis positive direction is towards lower pixel column
row-positive	rowPositive	CharacterString	C	1	axis positive direction is towards higher pixel row
row-negative	rowNegative	CharacterString	C	1	axis positive direction is towards lower pixel row
display-right	displayRight	CharacterString	C	1	axis positive direction is right in display
display-left	displayLeft	CharacterString	C	1	axis positive direction is left in display
display-up	displayUp	CharacterString	C	1	axis positive direction is towards top of approximately vertical display surface
display-down	displayDown	CharacterString	C	1	axis positive direction is towards bottom of approximately vertical display surface
forward	forward	CharacterString	C	1	axis positive direction is forward Note: For an observer at the centre of the object this is will be towards its front, bow or nose.
aft	aft	CharacterString	C	1	axis positive direction is aft Note: For an observer at the centre of the object this will be towards its back, stern or tail.
port	port	CharacterString	C	1	axis positive direction is port Note: For an observer at the centre of the object this will be towards its left.
starboard	starboard	CharacterString	C	1	axis positive direction is starboard Note: For an observer at the centre of the object this will be towards its right.
clockwise	clockwise	CharacterString	C	1	axis positive direction is clockwise from a specified direction
counter-clockwise	counterClockwise	CharacterString	C	1	axis positive direction is counter clockwise from a specified direction
towards	towards	CharacterString	C	1	axis positive direction is towards the object
away-from	awayFrom	CharacterString	C	1	axis positive direction is away from the object
future	future	CharacterString	C	1	temporal axis positive direction is towards the future
past	past	CharacterString	C	1	temporal axis positive direction is towards the past
unspecified	unspecified	CharacterString	C	1	axis positive direction is unspecified
Condition:	One and only one of the listed attributes shall be supplied.				

Table 49 — Defining elements of CoordinateSystems::RangeMeaning class

Definition:	meaning of the axis value range specified through minimumValue and maximumValue
Stereotype:	CodeList
Inheritance from:	(none)
Used by:	CoordinateSystemAxis

Public attributes:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute description
Exact	exact	CharacterString	C	1	any value between and including minimumValue and maximumValue is valid
Wraparound	wraparound	CharacterString	C	1	axis is continuous with values wrapping around at the minimumValue and maximumValue Note: Values with the same meaning repeat modulo the difference between maximumValue and minimumValue.
Condition:	One and only one of the listed attributes shall be supplied.				

11 Datums (reference frames) package

11.1 Types of datum and reference frame

A datum, in geodesy now usually referred to as a reference frame, relates a coordinate system to an object. For geodetic and vertical coordinate reference systems, the reference frame relates the coordinate system to the Earth or other planetary object. With other subtypes of coordinate reference systems, the datum may relate the coordinate system to another physical or virtual object. For an engineering CRS the datum may relate the coordinate system to a feature on the Earth such as a building, or to an origin point on an image. In other applications of an engineering CRS, the object may be a platform that is moving relative to the Earth. In these applications the datum itself is not time-dependent, but any transformations of the associated coordinates to an Earth-fixed or other coordinate reference system shall contain time-dependent parameters. For parametric CRSs the object may be a measurement such as an atmospheric pressure level.

In this document, several subtypes of the Datum class are recognized. Each subtype can be associated only with specific subtypes of coordinate reference systems. Constraints on Datum are detailed below.

Datums and reference frames are described further in C.4.

11.2 Geodetic reference frame

11.2.1 Prime meridian

If the subtype of Datum is geodetic reference frame, the description of the origin from which longitude values are specified – the prime meridian – is mandatory. Default values for the prime meridian name and Greenwich Longitude are "Greenwich" and 0, respectively. If the prime meridian name is "Greenwich" then the value of Greenwich Longitude is required to be 0 degrees. If the subtype of Datum is geodetic reference frame and the prime meridian specification is omitted, it is assumed to be the default.

A prime meridian specification is not permitted to be provided if the Datum subtype is not geodetic reference frame.

Prime meridian is described further in C4.2.2.

11.2.2 Ellipsoid

If the subtype of Datum is geodetic reference frame and the associated geographic CRS's coordinate system type is ellipsoidal, the description of one associated reference ellipsoid is mandatory. If the subtype of Datum is geodetic reference frame and the associated geodetic CRS's coordinate system type is Cartesian or spherical, the association to reference ellipsoid is optional; however if there is a recommended reference ellipsoid for the reference frame then it is advised that its description be included - see C.4.2.3.

A reference ellipsoid specification is not permitted to be provided if the Datum subtype is not geodetic reference frame.

11.3 Dynamic reference frame

If the subtype of Datum is geodetic or vertical, the frame-defining parameters may include time evolution to describe the motions of points used to define the reference frame. Then the geodetic or vertical reference frame is *dynamic*; the inclusion of the frame reference epoch is a mandatory attribute. Further information is provided in B.4.

11.4   Datum ensemble

A Datum Ensemble is a construct to facilitate the merging of realizations of the same Terrestrial Reference System or Vertical Reference System for lower accuracy spatial manipulation. In this document, datum ensemble is a collection of two or more reference frames that are realizations of one Terrestrial or Vertical Reference System and which for all but the highest accuracy requirements may be considered to be insignificantly different from each other. Datasets referenced to the various realizations may be merged without change of coordinates.

NOTE   For rigorous spatial positioning requirements the realizations must be treated individually. See C.4.7.

In the construction of a CRS, a datum ensemble may take the place of an individual datum. Single CRSs are constrained to have either a datum (or reference frame) or a datum ensemble.

11.5   Temporal datum

A temporal datum consists of a temporal origin and a calendar. In this document only the proleptic Gregorian calendar is explicitly recognised. Default value for the calendar is "propleticGregorian". The *proleptic Gregorian calendar* is produced by extending the *Gregorian calendar* backwards to dates preceding its official introduction in 1582. If the calendar specification is omitted, it is assumed to be the default.

The temporal origin is required to be expressed as a DateTime representation. The notation for a DateTime representation is defined in ISO 8601.

11.6   UML schema for the Datums package

Figure 13 shows the UML class diagram for the Datums package. There are restrictions on the associations between Coordinate Reference System subtypes and Datum subtypes which are shown in the UML class diagram in Figure 14.

The definition of the object classes of this package is provided in Tables 50 to 64.
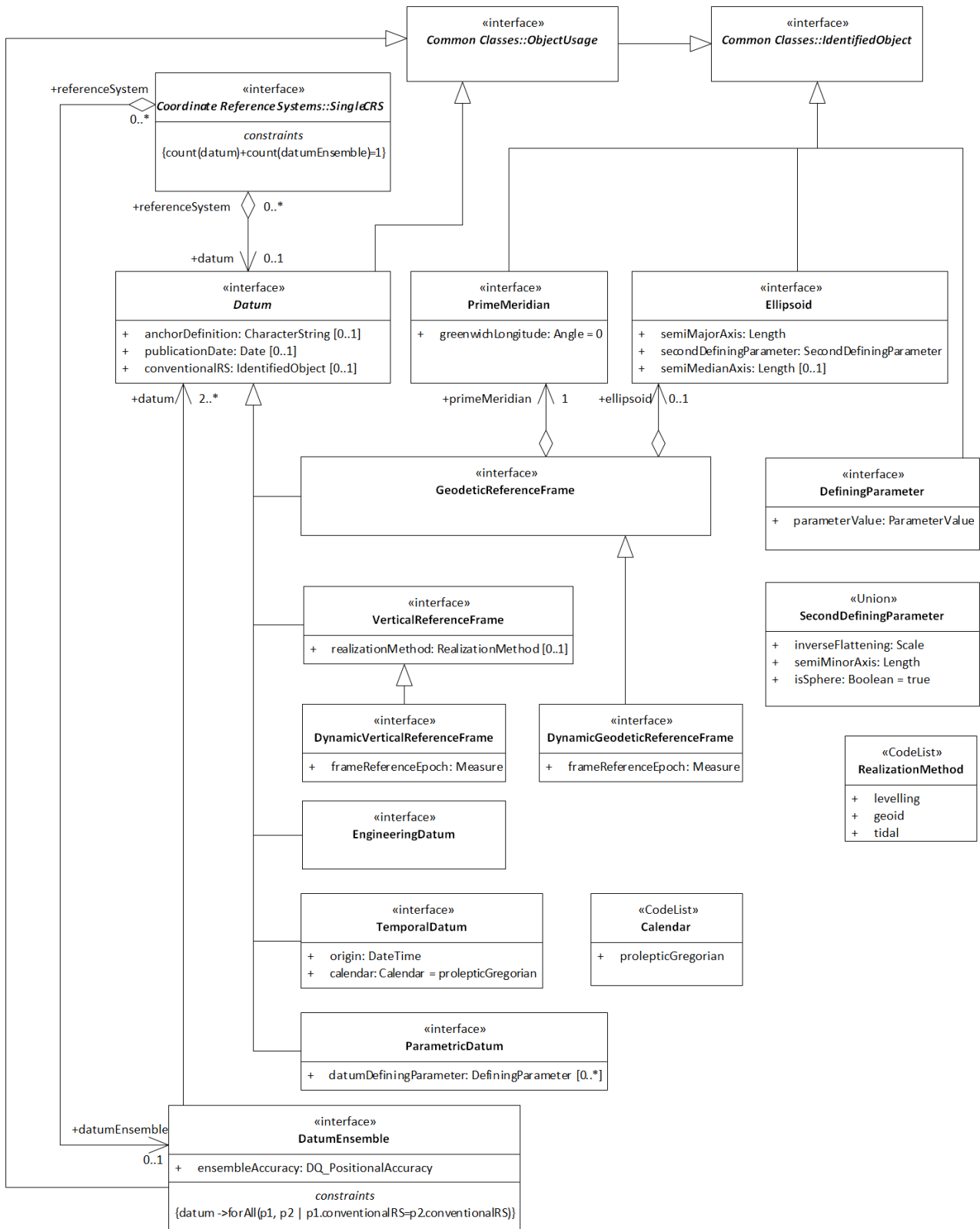
Figure 13 — UML diagram — Datums package
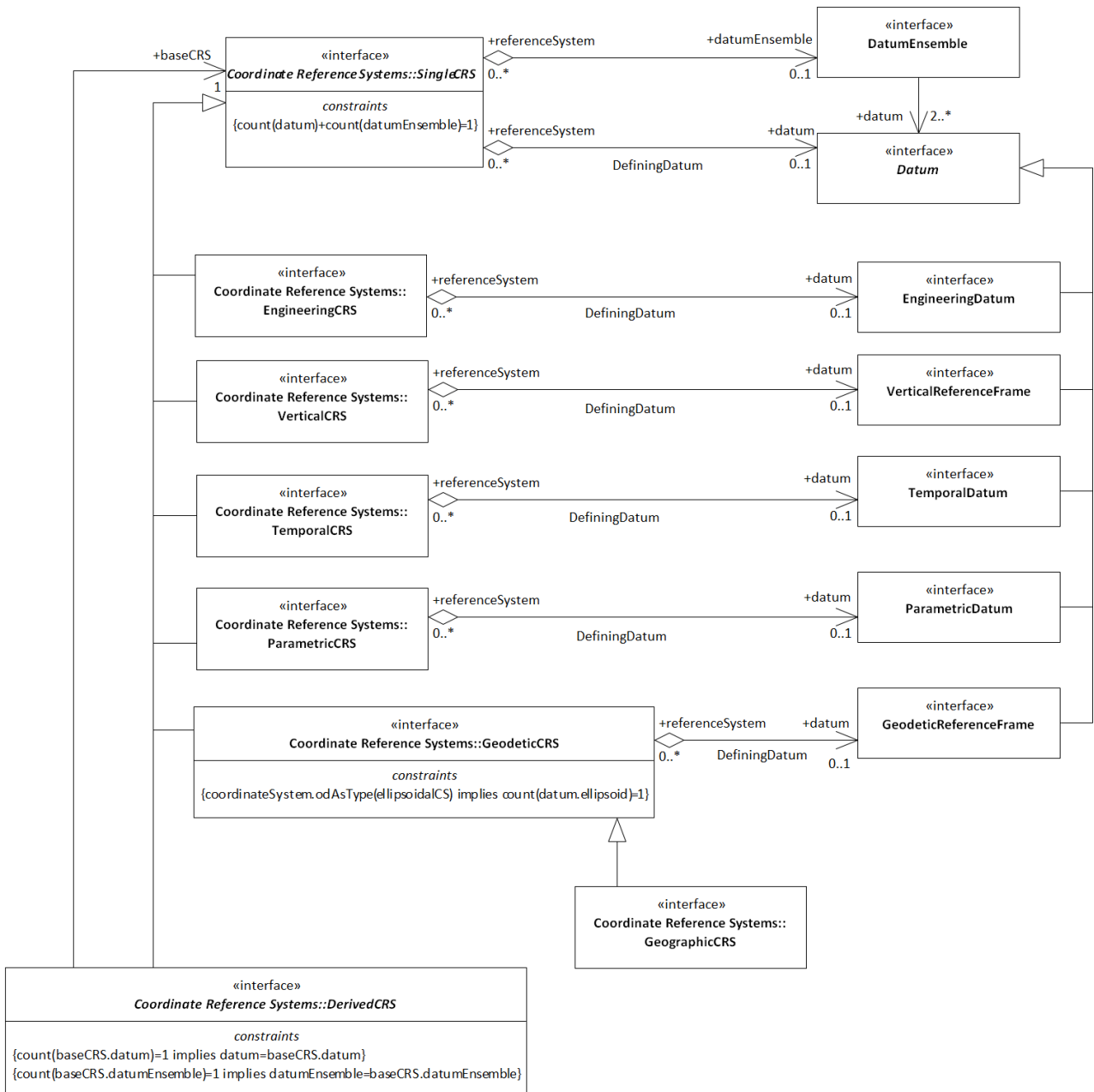
Figure 14 — UML diagram — Datum type associations with Coordinate Reference System type

Table 50 — Defining elements of Datums::Datum class

Definition:	specification of the relationship of a coordinate system to an object, thus creating a coordinate reference system Note: For geodetic and vertical coordinate reference systems, it relates a coordinate system to the Earth. With other types of coordinate reference systems, the datum may relate the coordinate system to another physical or virtual object. A datum uses a parameter or set of parameters that determine the location of the origin of the coordinate reference system. Each datum subtype can be associated with only specific types of coordinate reference systems.
Stereotype:	Interface
Class attribute:	Abstract
Inheritance from:	Common Classes::ObjectUsage
Generalization of:	EngineeringDatum GeodeticReferenceFrame ParametricDatum TemporalDatum VerticalReferenceFrame

Public attributes:	6 attributes (datum name, datum alias, datum identifier, datum scope, datum validity and datum remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage, plus:				
Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Datum anchor definition	anchorDefinition	CharacterString	O	1	description, possibly including coordinates of an identified point or points, of the relationship used to anchor a coordinate system to the Earth or alternate object Note: For modern geodetic reference frames the anchor may be a set of station coordinates; if the reference frame is dynamic it will also include coordinate velocities. For a traditional geodetic datum, the anchor may be a point known as the fundamental point, which is traditionally the point where the relationship between geoid and ellipsoid is defined, together with a direction from that point. - For a vertical reference frame the anchor may be the zero level at one or more defined locations or a conventionally defined surface. - For an engineering datum, the anchor may be an identified physical point with the orientation defined relative to the object.
Datum publication date	publicationDate	CI_Date	O	1	date on which the datum definition was published
Conventional reference system	conventionalRS	CommonClasses::IdentifiedObject	O	1	name, identifier, alias and remarks for the terrestrial reference system or vertical reference system realized by this reference frame Examples: "ITRS" for ITRF88 through ITRF2008 and ITRF2014, or "EVRS" for EVRF2000 and EVRF2007.

Remarks:	The constraint on the SingleCRS class *{count(datum) +count(datumEnsemble) = 1}* requires a single CRS to have either a datum or a datum ensemble. The constraint on the DatumEnsemble class *{datum → forAll(p1, p2 | p1.conventionalRS = p2.conventionalRS)}* requires that all reference frames that are members of a specified datum ensemble shall have the same terrestrial or vertical reference system.

Table 51 — Defining elements of Datums::GeodeticReferenceFrame class

Definition:	definition of the position, scale and orientation of a geocentric Cartesian 3D coordinate system relative to the Earth Note 1: It may also identify a defined ellipsoid (or sphere) that approximates the shape of the Earth and which is centred on and aligned to this geocentric coordinate system. Older geodetic datums define the location and orientation of a defined ellipsoid (or sphere) that approximates the shape of the earth. Note 2: In 19111:2007 this class was called GeodeticDatum.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	Datum
Generalization of:	DynamicGeodeticReferenceFrame

Association roles:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	(aggregation) ellipsoid	Ellipsoid	O	1	ellipsoid which is a component of this geodetic reference frame
(not named)	(aggregation) primeMeridian	PrimeMeridian	M	1	prime meridian which is a component of this geodetic reference frame
Remarks:		The constraint on GeodeticCRS of *{coordinateSystem.oclAsType(EllipsoidalCS) implies count(datum.ellipsoid)=1}* requires that if the CRS using the geodetic reference frame includes ellipsoidal coordinates then an association to Ellipsoid is mandatory. This constraint on GeodeticCRS is inherited by GeographicCRS.			

Public attributes:	9 attributes (datum name, datum alias, datum identifier, datum remarks, datum scope, datum validity, datum anchorDefinition, datum publicationDate and conventionalRS) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage and Datum.

Table 52 — Defining elements of Datums::DynamicGeodeticReferenceFrame class

Definition:	geodetic reference frame in which some of the parameters describe time evolution of defining station coordinates Example: defining station coordinates having linear velocities to account for crustal motion.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	GeodeticReferenceFrame

Public attributes:	9 attributes (datum name, datum alias, datum identifier, datum remarks, datum scope, datum validity, datum anchorDefinition, datum publication date and conventionalRS) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage, and Datum, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Frame reference epoch	frameReferenceEpoch	Measure	M	1	epoch to which the coordinates of stations defining the dynamic geodetic reference frame are referenced, usually given as a decimal year Example: 2016.47.

Table 53 — Defining elements of Datums::PrimeMeridian class

Definition:	origin meridian from which longitude values are determined Note: The default value for prime meridian name is "Greenwich". When the default applies, the value for the greenwichLongitude shall be 0 (degrees).
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	Common Classes::IdentifiedObject

Public attributes: 4 attributes (prime meridian name, prime meridian alias, prime meridian identifier and prime meridian remarks) inherited from Common Classes::IdentifiedObject, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Prime meridian Greenwich longitude	greenwichLongitude	Angle	M	1	longitude of the prime meridian measured from the internationally-recognised reference meridian ('Greenwich meridian'), positive eastward Note 1: Default value: 0 degrees. Note 2: If the value of the prime meridian *name* is "Greenwich" then the value of greenwichLongitude shall be 0 degrees.

Table 54 — Defining elements of Datums::Ellipsoid class

Definition:	geometric reference surface embedded in 3D Euclidean space formed by an ellipse that is rotated about a main axis Note: For the Earth the ellipsoid is bi-axial with rotation about the polar axis. This results in an oblate ellipsoid with the midpoint of the foci located at the nominal centre of the Earth.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	Common Classes::IdentifiedObject
Association roles:	
Remarks:	The constraint on GeodeticCRS of *{coordinateSystem.oclAsType(EllipsoidalCS) implies count(datum.ellipsoid)=1}* requires that if the CRS using the geodetic reference frame includes an ellipsoidal coordinate system then an association to ellipsoid is mandatory. This constraint on GeodeticCRS is inherited by GeographicCRS.

Public attributes: 4 attributes (ellipsoid name, ellipsoid alias, ellipsoid identifier and ellipsoid remarks) inherited from Common Classes::IdentifiedObject, plus

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Length of semi-major axis	semiMajorAxis	Length	M	1	length of the semi-major axis of the ellipsoid
Second defining parameter	secondDefiningParameter	SecondDefining Parameter	M	1	definition of the second parameter that describes the shape of this ellipsoid
Length of semi-median axis	semiMedianAxis	Length	O	1	length of the semi-median axis of a triaxial ellipsoid Note: This parameter is not required for a biaxial ellipsoid.

Table 55 — Defining elements of Datums::SecondDefiningParameter class

Definition:	definition of the second parameter that defines the shape of a biaxial ellipsoid, or the third parameter that defines a triaxial ellipsoid Note: A biaxial ellipsoid requires two defining parameters: a semi-major axis and inverse flattening or a semi-major axis and a semi-minor axis. When the reference body is a sphere rather than an ellipsoid, only a single defining parameter is required, namely the radius of the sphere; in that case, the semi-major axis "degenerates" into the radius of the sphere.
Stereotype:	Union
Inheritance from:	(none)
Used by:	Ellipsoid

Public attributes:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Inverse flattening	inverseFlattening	Scale	C	1	inverse flattening value of the ellipsoid
Length of semi-minor axis	semiMinorAxis	Length	C	1	length of the semi-minor axis of the ellipsoid
"Ellipsoid = Sphere" indicator	isSphere	Boolean	C	1	ellipsoid is degenerate and is actually a sphere Note: The sphere is completely defined by the semi-major axis, which is the radius of the sphere. This attribute has the value "true" if the figure is a sphere.

Condition:	union (one of) constraint on inverseFlattening, semiMinorAxis and Sphere attributes. One and only one of these three elements shall be supplied.
Remarks:	In the case of a triaxial ellipsoid (when the semi-median axis attribute is provided), the SecondDefiningParameter element supplied should be the semiMinorAxis.

Table 56 — Defining elements of Datums::VerticalReferenceFrame class

Definition:	textual description and/or a set of parameters identifying a particular reference level surface used as a zero-height or zero-depth surface, including its position with respect to the Earth Note: In 19111:2007 this class was called VerticalDatum.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	Datum
Generalization of:	DynamicVerticalReferenceFrame
Public attributes:	9 attributes (datum name, datum alias, datum identifier, datum remarks, datum scope, datum validity, datum anchorDefinition, datum publication date and conventionalRS) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage and Datum, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Realization method	realizationMethod	RealizationMethod	O	1	method through which this vertical reference frame is realized

Table 57 — Defining elements of Datums:: RealizationMethod class

Definition:	specification of the method by which the vertical reference frame is realized
Stereotype:	CodeList
Inheritance from:	(none)
Used by:	VerticalReferenceFrame

Public attributes:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Levelling-based	levelling	CharacterString	C	1	realization is by adjustment of a levelling network fixed to one or more tide gauges
Geoid-based	geoid	CharacterString	C	1	realization is through a geoid height model or a height correction model Note: This is applied to a specified geodetic CRS.
Tidal	tidal	CharacterString	C	1	realization is through a tidal model or by tidal predictions
Condition:	One and only one of the listed attributes shall be supplied.				

Table 58 — Defining elements of Datums::DynamicVerticalReferenceFrame class

Definition:	vertical reference frame in which some of the defining parameters have time dependency Example: Defining station heights have velocity to account for post-glacial isostatic rebound motion.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	VerticalReferenceFrame
Public attributes:	10 attributes (datum name, datum alias, datum identifier, datum remarks, datum scope, datum validity, datum anchor, datum publication date, conventionalRS and vertical reference frame realization method) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage, Datum and VerticalReferenceFrame plus:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Frame reference epoch	frameReferenceEpoch	Measure	M	1	epoch to which the coordinates of stations defining the dynamic vertical reference frame are referenced, usually given as a decimal year Example: 2016.47.

Table 59 — Defining elements of Datums::ParametricDatum class

Definition:	textual description and/or a set of parameters identifying a particular reference surface used as the origin of a parametric coordinate system, including its position with respect to the Earth
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	Datum
Public attributes:	9 attributes (datum name, datum alias, datum identifier, datum remarks, datum scope, datum validity, datum anchor, datum publication date and conventionalRS) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage and Datum, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Datum Defining Parameter	datumDefiningParameter	DefiningParameter	O	N	parameter used to define the parametric datum

Table 60 — Defining elements of Datums::DefiningParameter class

Definition:	parameter value, an ordered sequence of values, or a reference to a file of parameter values that define a paramtric datum
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	Common Classes::IdentifiedObject
Used by:	ParametricDatum
Association roles:	(none)

Public attributes:	4 attributes (parameter name, parameter alias, parameter identifier and parameter remarks) inherited from Common Classes::IdentifiedObject, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Parameter value	parameterValue	CoordinateOperations:: ParameterValue	M	1	value of the datum-defining parameter

Table 61 — Defining elements of Datums::EngineeringDatum class

Definition:	definition of the origin and orientation of an engineering coordinate reference system Note: The origin can be fixed with respect to the Earth (such as a defined point at a construction site), or be a defined point on a moving vehicle (such as on a ship or satellite), or a defined point of an image.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	Datum
Public attributes:	9 attributes (datum name, datum alias, datum identifier, datum remarks, datum validity, datum scope, datum anchor, datum publication date and conventionalRS) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage and Datum.

Table 62 — Defining elements of Datums::TemporalDatum class

Definition:	definition of the relationship of a temporal coordinate system to an object Note: The object is normally time on the Earth.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	Datum

Public attributes:	9 attributes (datum name, datum alias, datum identifier, datum remarks, datum scope, datum validity, datum anchor, datum publication date and conventionalRS) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage and Datum, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Temporal origin	origin	dateTime	M	1	date and time to which temporal coordinates are referenced, expressed in conformance with ISO 8601
Calendar	calendar	Calendar	M	1	calendar to which the temporal origin is referenced Note: Default value is prolepticGregorian.

Table 63 — Defining elements of Datums::Calendar class

Definition:	specification of the calendar to which a temporal origin is referenced				
Stereotype:	CodeList				
Inheritance from:	(none)				
Used by:	TemporalDatum				
Public attributes:					
Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Proleptic Gregorian	prolepticGregorian	CharacterString	C	1	proleptic Gregorian calendar as defined in ISO 8601 Note: This is the default value.
Condition:	Only one attribute shall be supplied.				

Table 64 — Defining elements of Datums::DatumEnsemble class

Definition:	collection of two or more geodetic or vertical reference frames (or if not geodetic or vertical reference frame, a collection of two or more datums) which for all but the highest accuracy requirements may be considered to be insignificantly different from each other Note: Every frame or datum within the datum ensemble must be a realization of the same reference system or datum.				
Stereotype:	Interface				
Class attribute:	Concrete				
Inheritance from:	Common Classes::ObjectUsage				
Association roles:					
Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	datum	Datum	M (minimum 2)	N	datum or reference frame which is a member of this datum ensemble
Constraint:	{datum → forAll(p1, p2 | p1.conventionalRS = p2.conventionalRS)}				
Remarks:	The constraint requires that reference frames (datums) that are members of the same datum ensemble shall all have the same conventionalRS.				
Public attributes:	6 attributes (datum ensemble name, datum ensemble alias, datum ensemble identifier, datum ensemble scope, datum ensemble validity and datum ensemble remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage, plus:				
Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Ensemble Accuracy	ensembleAccuracy	DQ_PositionalAccuracy	M	1	inaccuracy introduced through use of this collection of reference frames or datums Note: It is an indication of the differences in coordinate values at all points between the various realizations that have been grouped into this datum ensemble.
Note:	The constraint on the SingleCRS class of *{count(datum) + count datumEnsemble) = 1}* requires a single CRS to have either a datum or a datum ensemble.				

12 Coordinate Operations package

12.1 General characteristics of coordinate operations

In this document the following subtypes of coordinate operation are recognized:

g) *a single coordinate operation*. A single coordinate operation has a method - the mathematical formula it uses - together with parameters used in the formula. In an instance of a coordinate operation, the parameter values are specific to that instance. In an implementation this will be through either a coordinate conversion, a coordinate transformation or a point motion operation.

   1) A *coordinate conversion* (CC) changes coordinates from one coordinate reference system to another coordinate reference system based on the same datum (reference frame).

   2) A *coordinate transformation* (CT) changes coordinates from one coordinate reference system to another coordinate reference system which is based on another datum (reference frame).

   3) A *point motion operation* (PMO) changes coordinates within one coordinate reference system to account for the motion of the point within the CRS over a period of time.

h) A *concatenated coordinate operation* is a non-repeating sequence of single coordinate operations.

   EXAMPLE Changing coordinates from being referenced to CRS A to being referenced to CRS B through coordinate transformation CRS A to CRS C followed by coordinate transformation CRS C to CRS B.

   The sequence of coordinate operations is constrained by the requirement that the source coordinate reference system of step (n + 1) is required to be the same as the target coordinate reference system of step (n). The source coordinate reference system of the first step and the target coordinate reference system of the last step are the source and target coordinate reference system associated with the concatenated coordinate operation. For a concatenated coordinate operation sequence of n coordinate operations:

   1) sourceCRS (concatenated coordinate operation) = sourceCRS (coordinate operation step 1)

   2) targetCRS (coordinate operation step i) = sourceCRS (coordinate operation step i + 1); i = 1 ... ($n-1$)

   3) target CRS (concatenated coordinate operation) = target CRS (coordinate operation step n)

   Instead of a forward coordinate operation, an inverse coordinate operation may be used for one or more of the coordinate operation steps mentioned above, but only if the inverse coordinate operation is uniquely defined by the forward coordinate operation method.

i) A *pass-through coordinate operation* allows a subset of a coordinate tuple to be subjected to a coordinate operation; coordinates in the coordinate tuple other than the subset remain unchanged.

   EXAMPLE An operation to derive coordinates in a projected 3D CRS from coordinates in a three-dimensional geodetic CRS. The geodetic latitude and geodetic longitude coordinates in the base CRS are converted to easting and northing coordinates in the projected CRS but the base CRS's vertical coordinate (ellipsoidal height) is unchanged.

Coordinate operations are further described in C.5.

12.2   UML schema for the Coordinate Operations package

Figures 15 and 16 contain the two parts of the UML class diagram for the Coordinate Operations package. As indicated by the note in Figure 15, Figure 16 shows additional classes and associations from the SingleOperation class shown in Figure 15. The definition of the object classes of the Coordinate Operations package is provided in Tables 65 to 82.

In this document the CoordinateOperation class has two purposes:

   i)   To describe a coordinate operation;

   ii)   To apply a change of coordinates.

An operation CoordinateOperation.*transform(CoordinateSet)* applies a coordinate operation to the coordinates within a coordinate set. It and its constraints are shown in Figure 6 and discussed in Clause 7. Only those attributes relevant to the description of a coordinate operation are shown in Figures 15 and 16 and in the following tables.

The Coordinate Operations package UML class diagram shows two associations named Source and Target from the CoordinateOperation class to the CRS class. These indicate the CRS from which coordinates are changed and the CRS to which coordinates are changed respectively; they form part of the definition of a coordinate operation. They should not be confused with the transform within the CoordinateOperation class which acts on coordinates and is described in Clause 7. The Source and Target associations are mandatory for all subtypes of coordinate operation except coordinate conversion. Coordinate conversions that are part of the definition of a derived CRS do not use these associations; the source and target CRSs for such a defining coordinate conversion are identified through the association from DerivedCRS to SingleCRS with the baseCRS having the role of sourceCRS for the coordinate conversion.

The Coordinate Operations package UML class diagram also shows an additional association from the CoordinateOperation class to the CRS class, named Interpolation. Some single coordinate operations employ methods which include interpolation within a grid to derive the values of operation parameters. The CRS to be used for the interpolation may be different from either the sourceCRS or the targetCRS. The Interpolation association specifies the CRS to be used for the interpolation.

EXAMPLE   Vertical offsets between two vertical CRSs interpolated from a grid. The source and target CRSs will both be vertical CRSs, the interpolation CRS is a geographic CRS to which the grid is referenced.

   An example is given in C.5.1.
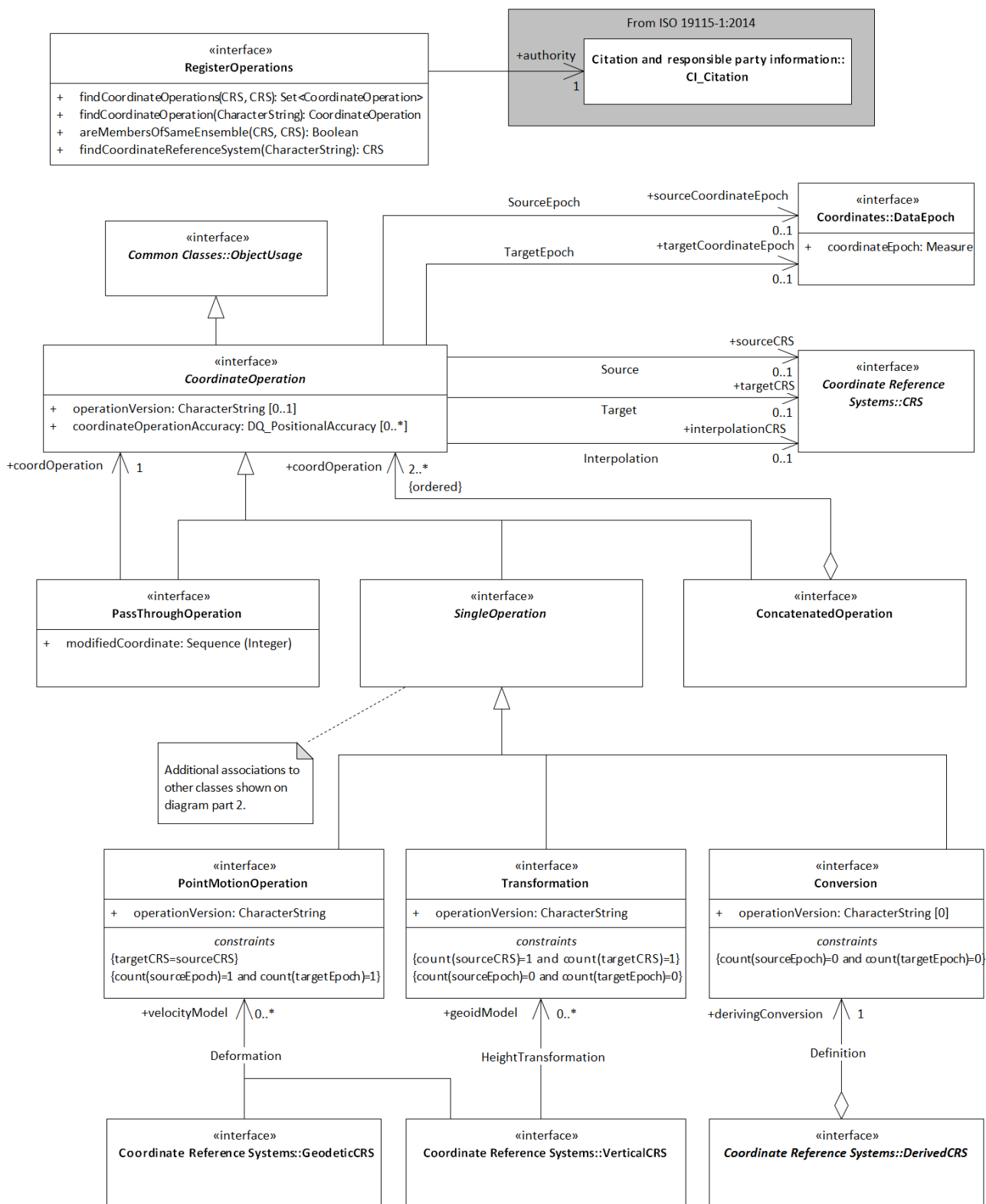
Figure 15 — UML diagram — Coordinate Operations package part 1
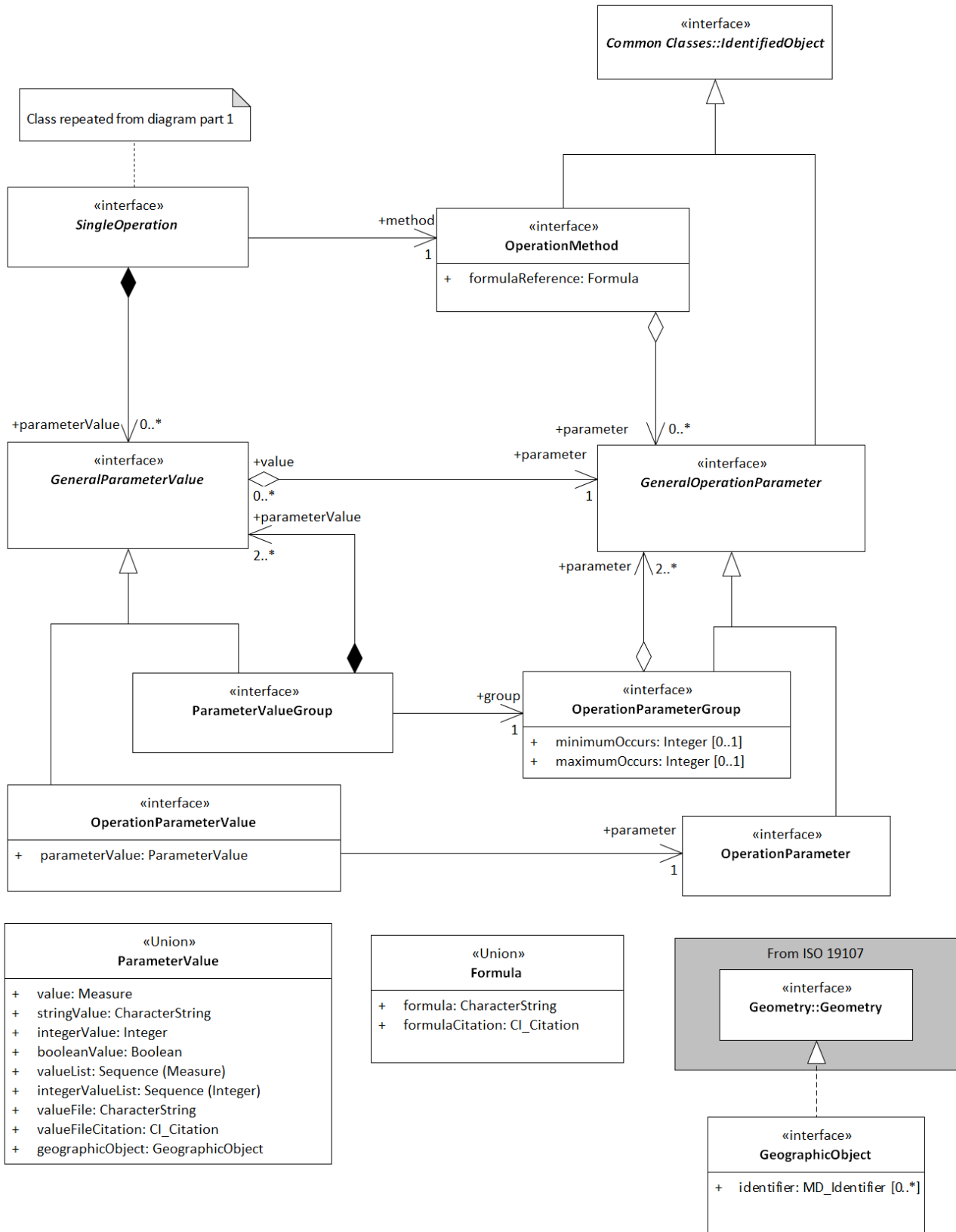
Figure 16 — UML diagram — Coordinate Operations package part 2

Table 65 — Defining elements of Coordinate Operations::CoordinateOperation class

Definition:	mathematical operation (a) on coordinates that transforms or converts them from one coordinate reference system to another coordinate reference system, or (b) that decribes the change of coordinate values within one coordinate reference system due to the motion of the point between one coordinate epoch and another coordinate epoch Note: Many but not all coordinate operations (from CRS A to CRS B) also uniquely define the inverse coordinate operation (from CRS B to CRS A). In some cases, the coordinate operation method algorithm for the inverse coordinate operation is the same as for the forward algorithm, but the signs of some coordinate operation parameter values have to be reversed. In other cases, different algorithms are required for the forward and inverse coordinate operations, but the same coordinate operation parameter values are used. If (some) entirely different parameter values are needed, a different coordinate operation shall be defined.
Stereotype:	Interface
Class attribute:	Abstract
Inheritance from:	IdentifiedObject::ObjectUsage
Generalization of:	ConcatenatedOperation PassThroughOperstion SingleOperation

Association roles:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
Source	sourceCRS	CoordinateReferenceSystems:: CRS	O	1	coordinate reference system to which the coordinate set input into this coordinate operation is referenced
Target	targetCRS	CoordinateReferenceSystems:: CRS	O	1	coordinate reference system to which the coordinate set output from this coordinate operation is referenced
Interpolation	interpolationCRS	CoordinateReferenceSystems:: CRS	O	1	coordinate reference system to which gridded data files are referenced which this coordinate operation uses to transform coordinates between two other coordinate reference systems Note: InterpolationCRS is only used when it is different from both sourceCRS and targetCRS.
Source Epoch	sourceCoordinate Epoch	Coordinates::DataEpoch	O	1	coordinate epoch of the coordinate set input into this coordinate operation
Target Epoch	target Coordinate Epoch	Coordinates::DataEpoch	O	1	coordinate epoch of the coordinate set output from this coordinate operation

Public attributes: 6 attributes (coordinate operation name, coordinate operation alias, coordinate operation identifier, coordinate operation scope, coordinate operation validity and coordinate operation remarks) inherited from Common Classes::IdentifiedObject and Common Classes::ObjectUsage, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Coordinate operation version	operationVersion	CharacterString	C	1	version of the coordinate transformation (i.e. instantiation due to the stochastic nature of the parameters) Condition: Mandatory when describing a coordinate transformation or point motion operation, and should not be supplied for a coordinate conversion.
Coordinate operation accuracy	coordinateOperation Accuracy	DQ_Positional Accuracy	O	N	estimate(s) of the impact of this coordinate operation on point accuracy Note: Gives position error estimates for target coordinates of this coordinate operation, assuming no errors in source coordinates.

(Table continued on following page)

Table 65 (continued)

Operation name	UML identifier	Arguments	Output	Operation Definition
Transform coordinate set	transform(CoordinateSet) {constraints}	CoordinateSet	CoordinateSet	operation that changes coordinate values of all coordinate tuples in a coordinate set from being referenced to one CRS to being referenced to another CRS and/or from being referenced to one coordinate epoch to being referenced to another coordinate epoch

Constraints:	{transform() pre sourceCRS=CoordinateSet.coordinateMetadata.crs}
	{transform() pre count(CoordinateSet.coordinateMetadata.coordinateEpoch)=1 implies sourceCoordinateEpoch=CoordinateSet.coordinateMetadata.coordinateEpoch}
	{transform() post CoordinateSet.coordinateMetadata.crs =targetCRS}
	{transform() post count(CoordinateSet.coordinateMetadata.coordinateEpoch)=1 implies CoordinateSet.coordinateMetadata.coordinateEpoch=sourceCoordinateEpoch}

Remarks: The application of transform(CoordinateSet) and the constraints on the Coordinate Operation class are described in Clause 7.

Table 66 — Defining elements of Coordinate Operations::PassThroughOperation class

Definition:	specification of a subset of coordinate tuples that is subject to a coordinate operation
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	CoordinateOperation

Association roles: associations inherited from CoordinateOperation, plus:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	(aggregation) coordOperation	CoordinateOperation	M	1	subset of a coordinate tuple that the coordinate operation will operate upon

Public attributes: 8 attributes (coordinate operation name, coordinate operation alias, coordinate operation identifier, coordinate operation scope, coordinate operation validity, coordinate operation remarks, coordinate operation version and coordinate operation accuracy) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage and CoordinateOperation, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Modified coordinates	modifiedCoordinate	Sequence<Integer>	M	1	ordered sequence of positive integers defining the positions in a source coordinate tuple of the coordinates affected by this pass-through operation

Table 67 — Defining elements of Coordinate Operations::ConcatenatedOperation class

Definition:	ordered sequence of two or more single coordinate operations
	Note: The sequence of coordinate operations is constrained by the requirement that the source coordinate reference system of step (n + 1) shall be the same as the target coordinate reference system of step (n). The source coordinate reference system of the first step and the target coordinate reference system of the last step are the source and target coordinate reference system associated with the concatenated coordinate operation. For a concatenated coordinate operation sequence of n coordinate operations:
	source CRS (concatenated coordinate operation) = source CRS (coordinate operation step 1)
	target CRS (coordinate operation step i) = source CRS (coordinate operation step i + 1); i = 1 …(n − 1)
	target CRS (concatenated coordinate operation) = target CRS (coordinate operation step n)
	Instead of a forward coordinate operation, an inverse coordinate operation may be used for one or more of the coordinate operation steps mentioned above, if the inverse coordinate operation is uniquely defined by the forward coordinate operation method.

Stereotype:	Interface

Class attribute:	Concrete

Inheritance from:	CoordinateOperation

Association roles: associations inherited from CoordinateOperation, plus:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	(aggregation) coordOperation	CoordinateOperation {ordered}	M (minimum 2)	N	coordinate operation that is a step in the sequence forming this concatenated coordinate operation

Public attributes:	8 attributes (coordinate operation name, coordinate operation alias, coordinate operation identifier, coordinate operation scope, coordinate operation validity, coordinate operation remarks, coordinate operation version and coordinate operation accuracy) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage and CoordinateOperation.

Table 68 — Defining elements of Coordinate Operations::SingleOperation class

Definition:	single (not concatenated) coordinate operation

Stereotype:	Interface

Class attribute:	Abstract

Inheritance from:	CoordinateOperation

Generalization of:	Conversion
	Transformation
	PointMotionOperation

Association roles: associations inherited from CoordinateOperation, plus:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	method	OperationMethod	M	1	algorithm or procedure used by this single operation
(not named)	(composition) parameterValue	GeneralParameterValue	O	N	parameter value or parameter value group used by this single operation

Public attributes:	8 attributes (coordinate operation name, coordinate operation alias, coordinate operation identifier, coordinate operation scope, coordinate operation validity, coordinate operation remarks, coordinate operation version and coordinate operation accuracy) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage and CoordinateOperation.

Table 69 — Defining elements of Coordinate Operations::Transformation class

Definition:	mathematical operation on coordinates in which parameters are empirically derived from data containing the coordinates of a series of points in both coordinate reference systems Note: This computational process is usually "over-determined", allowing derivation of error (or accuracy) estimates for the coordinate transformation. Also, the stochastic nature of the parameters may result in multiple (different) versions of the same coordinate transformations between the same source and target CRSs. Any single coordinate operation in which the input and output coordinates are referenced to different datums (reference frames) will be a coordinate transformation.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	SingleOpereration
Constraints:	{count(sourceCRS)=1 and count(targetCRS)=1} {count(sourceEpoch)=0 and count(targetEpoch)=0}
Remarks:	The constraints enforce that for a Transformation the "sourceCRS" and "targetCRS" associations are mandatory, the "sourceEpoch" and "targetEpoch" associations are not applicable.
Public attributes:	8 attributes (coordinate operation name, coordinate operation alias, coordinate operation identifier, coordinate operation scope, coordinate operation validity, coordinate operation remarks, coordinate operation version and coordinate operation accuracy) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage and CoordinateOperation, one of which is modified:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Coordinate operation version	operationVersion	CharacterString	M	1	version of the coordinate transformation (i.e. instantiation due to the stochastic nature of the parameters)

Table 70 — Defining elements of Coordinate Operations::Conversion class

Definition:	mathematical operation on coordinates in which the parameter values are defined rather than empirically derived; application of the coordinate conversion introduces no error into output coordinates Note: The best-known example of a coordinate conversion is a map projection. For coordinate conversions the output coordinates are referenced to the same datum as are the input coordinates. Coordinate conversions forming a component of a derived CRS have a source CRS and a target CRS that are NOT specified through the source and target associations, but through associations from DerivedCRS to SingleCRS.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	SingleOperation
Constraints:	{count(sourceEpoch)=0 and count(targetEpoch)=0}
Remarks:	Conversion inherits associations *SourceEpoch* and *TargetEpoch* from the CoordinateOperation class. This constraint enforces that these associations are not applicable in a conversion.
Public attributes:	8 attributes (coordinate operation name, coordinate operation alias, coordinate operation identifier, coordinate operation scope, coordinate operation validity, coordinate operation remarks, coordinate operation version and coordinate operation accuracy) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage and CoordinateOperation, one of which is modified:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Coordinate operation version	operationVersion	CharacterString	O	0	(not applicable) Note: This attribute is not used in a coordinate conversion.

Remarks:	The "sourceCRS" and "targetCRS" associations are mandatory for describing coordinate conversions which are not part of the definition of a derived CRS. However coordinate conversions defining a derivedCRS have a source CRS and a target CRS that are NOT specified through these associations but through associations from DerivedCRS to SingleCRS. See C.5.1.

Table 71 — Defining elements of Coordinate Operations::PointMotionOperation class

Definition:	mathematical operation that decribes the change of coordinate values within one coordinate reference system due to the motion of the point between one coordinate epoch and another coordinate epoch Note: In this document the motion is due to tectonic plate movement or deformation.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	SingleOpereration
Constraints:	(targetCRS = sourceCRS) (count(sourceEpoch)=1 and count(targetEpoch)=1}
Remarks:	The constraints enforce that for PointMotionOperation the "sourceEpoch" and "targetEpoch" associations are mandatory. The PointMotionOperation operates within a CRS so the source CRS and the target CRS associations must be to the same.

Public attributes: 8 attributes (coordinate operation name, coordinate operation alias, coordinate operation identifier, coordinate operation scope, coordinate operation validity, coordinate operation remarks, coordinate operation version and coordinate operation accuracy) inherited from Common Classes::IdentifiedObject, Common Classes::ObjectUsage and CoordinateOperation, one of which is modified:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Coordinate operation version	operationVersion	CharacterString	M	1	version of the point motion operation (i.e. instantiation due to the stochastic nature of the parameters) Note: This attribute is mandatory in a point motion operation.

Table 72 — Defining elements of Coordinate Operations::OperationMethod class

Definition:	method (algorithm or procedure) used to perform the coordinate operation
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	Common Classes::IdentifiedObject

Association roles:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	(aggregation) parameter	GeneralOperationParameter	O	N	parameter or parameter group used by this coordinate operation method

Public attributes: 4 attributes (coordinate operation method name, coordinate operation method alias, coordinate operation method identifier and coordinate operation method remarks) inherited from Common Classes::IdentifiedObject, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Coordinate operation method formula reference	formulaReference	CC_Formula	M	1	formula(s) or procedure used by this coordinate operation method Note: This may be a reference to a publication. Note that the operation method may not be analytic, in which case this attribute references or contains the procedure, not an analytic formula.

Table 73 — Defining elements of Coordinate Operations::Formula class

Definition:	specification of the coordinate operation method formula				
Stereotype:	Union				
Inheritance from:	(none)				
Used by:	OperationMethod				
Public attributes:					
Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Coordinate operation method formula	formula	CharacterString	C	1	formula(s) or procedure used by the coordinate operation method
Coordinate operation method formula citation	formulaCitation	CI_Citation	C	1	reference to a publication giving the formula(s) or procedure used by the coordinate operation method
Condition: union (one of) constraint on formula and formulaCitation attributes. One and only one of the listed attributes shall be supplied.					

Table 74 — Defining elements of Coordinate Operations::GeneralOperationParameter class

Definition:	definition of a parameter or group of parameters used by a coordinate operation method
Stereotype:	Interface
Class attribute:	Abstract
Inheritance from:	Common Classes::IdentifiedObject
Generalization of:	OperationParameter OperationParameterGroup
Public attributes:	4 attributes (coordinate operation parameter name, coordinate operation parameter alias, coordinate operation parameter identifier and coordinate operation parameter remarks) inherited from Common Classes::IdentifiedObject.

Table 75 — Defining elements of Coordinate Operations::OperationParameterGroup class

Definition:	definition of a group of related parameters used by a coordinate operation method				
Stereotype:	Interface				
Class attribute:	Concrete				
Inheritance from:	GeneralOperationParameter				
Association roles:	associations inherited from GeneralOperationParameter, plus:				
Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	parameter	GeneralOperationParameter	M (minimum 2)	N	parameter that is a member of this parameter group
Public attributes:	4 attributes (coordinate operation parameter name, coordinate operation parameter alias, coordinate operation parameter identifier and coordinate operation parameter remarks) inherited from Common Classes::IdentifiedObject, plus:				
Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Minimum occurrences	minimumOccurs	Integer	O	1	minimum number of times that values for this parameter group or parameter is required Note: If this attribute is omitted, the minimum number is one.
Maximum occurrences	maximumOccurs	Integer	O	1	maximum number of times that values for this parameter group or parameter can be included Note: If this attribute is omitted, the maximum number is one.

Table 76 — Defining elements of Coordinate Operations::OperationParameter class

Definition:	definition of a parameter used by a coordinate operation method Note: Most parameter values are numeric, but other types of parameter values are possible.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	GeneralOperationParameter
Public attributes:	4 attributes (coordinate operation parameter name, coordinate operation parameter alias, coordinate operation parameter identifier, and coordinate operation parameter remarks) inherited from Common Classes::IdentifiedObject.

Table 77 — Defining elements of Coordinate Operations::GeneralParameterValue class

Definition:	parameter value or group of parameter values
Stereotype:	Interface
Class attribute:	Abstract
Inheritance from:	(none)
Generalization of:	OperationParameterValue ParameterValueGroup

Association roles:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	(aggregation) parameter	GeneralOperationParameter	M	1	parameter or parameter group which has this value or value group

Public attributes:	(none)

Table 78 — Defining elements of Coordinate Operations::ParameterValueGroup class

Definition:	group of related parameter values Note: The same group can be repeated more than once in a coordinate operation or higher level ParameterValueGroup, if those instances contain different values of one or more ParameterValues which suitably distinguish among those groups.
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	GeneralParameterValue

Association roles: associations inherited from GeneralParameterValue, plus:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	(composition) parameterValue	GeneralParameterValue	M (minimum 2)	N	value in this value group
(not named)	group	OperationParameterGroup	M	1	parameter group associated with this value group

Public attributes:	(none)

Table 79 — Defining elements of Coordinate Operations::OperationParameterValue class

Definition:	parameter value, ordered sequence of values, or reference to a file of parameter values				
Stereotype:	Interface				
Class attribute:	Concrete				
Inheritance from:	GeneralParameterValue				
Association roles:	associations inherited from GeneralParameterValue, plus:				
Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	parameter	(aggregation) OperationParameter	M	1	parameter which has this value
Public attributes:					
Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Parameter value	parameterValue	CoordinateOperations:: ParameterValue	M	1	value of the coordinate operation parameter

Table 80 — Defining elements of Coordinate Operations::ParameterValue class

Definition:	value of the coordinate operation parameter
Stereotype:	Union
Inheritance from:	(none)
Used by:	OperationParameterValue

Public attributes:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Operation parameter numeric value	value	Measure	C	1	numeric value of the coordinate operation parameter with its associated unit
Operation parameter string value	stringValue	CharacterString	C	1	string value of a coordinate operation parameter Note: A string value does not have an associated unit.
Operation parameter integer value	integerValue	Integer	C	1	positive integer value of a coordinate operation parameter, usually used for a count Note: An integer value does not have an associated unit.
Operation parameter Boolean value	booleanValue	Boolean	C	1	boolean value of a coordinate operation parameter Note: A Boolean value does not have an associated unit.
Operation parameter value list	valueList	Sequence<Measure>	C	1	ordered collection, i.e. sequence, of two or more numeric values of a coordinate operation parameter list, where each value has the same associated unit.
Operation parameter integer value list	integerValueList	Sequence<Integer>	C	1	ordered collection, i.e. sequence, of two or more integer values of a coordinate operation parameter list, usually used for counts Note: These integer values do not have an associated unit.
Operation parameter file reference	valueFile	CharacterString	C	1	reference to a file or a part of a file containing one or more parameter values Note: When referencing a part of a file, that file shall contain multiple identified parts, such as an XML encoded document. Furthermore, the referenced file or part of a file can reference another part of the same or different files, as allowed in XML documents.
Operation parameter file reference citation	valueFileCitation	CI_Citation	C	1	citation for a reference to a file or a part of a file containing one or more parameter values Note: When referencing a part of a file, that file shall contain multiple identified parts, such as an XML encoded document. Furthermore, the referenced file or part of a file can reference another part of the same or different files, as allowed in XML documents.
Geographic object	geographicObject	GeographicObject	C	1	identifier of a geographic feature of which the coordinates are used as operation parameters

Condition: union (one of) constraint on these attributes. One and only one of the listed attributes shall be supplied.

Table 81 — Defining elements of Coordinate Operations::GeographicObject class

Definition:	identification of an object used as a parameter in a coordinate transformation, point motion operation or coordinate conversion
Stereotype:	Interface
Class attribute:	Concrete
Realization of:	Geometry::Geometry (ISO 19107). As such, it must implement all inherited operations and associations. Furthermore, it must support all inherited attributes, at least as "read only".

Public attributes:

Attribute name	UML identifier	Data type	Obligation	Maximum Occurrence	Attribute Definition
Geographic object identifier	identifier	MD_Identifier	O	N	identifier of the geographic object

Table 82 — Defining elements of Coordinate Operations::RegisterOperations class

Definition:	operations supported in the Coordinate Operations package
Stereotype:	Interface
Class attribute:	Concrete
Inheritance from:	(none)

Association roles:

Association name	UML identifier	Association with	Obligation	Maximum Occurrence	Association definition
(not named)	authority	Citation and responsible party information::CI_Citation	M	1	citation used by this register operation

Note: CI_Citation is described in ISO 19115.

Public attributes:

Operation name	UML identifier	Arguments	Output	Operation Definition
Find coordinate operations	findCoordinateOperations	(CRS,CRS)	Set<CoordinateOperation>	operation to find any coordinate operations for which the given CRSs are the source and target, in that order Note: This is the reverse navigation for the associations *Source* and *Target* from CoordinateOperation to Coordinate Reference Systems::CRS.
Find Coordinate Operation	findCoordinateOperation	(CharacterString)	CoordinateOperation	operation to extract Coordinate Operation details from a registry
Find Coordinate ReferenceSystem	findCoordinateReferenceSystem	(CharacterString)	CRS	operation to extract CRS details from a registry
Members Of Same Datum Ensemble	areMembersOfSameEnsemble	(CRS,CRS)	Boolean	operation to determine whether two coordinate reference systems are members of one ensemble Note: If this returns true then for low accuracy purposes coordinate sets referenced to these CRSs may be merged without coordinate transformation. The attribute 'DatumEnsemble.ensembleAccuracy' gives some indication of the inaccuracy introduced through such merger.

Annex A: Abstract test suite (normative)

A.1 Conformance — General

To verify whether a coordinate reference system or coordinate operation is in conformance with this document, check that it satisfies the requirements for the appropriate conformance class given in A.2 to A.4. Conformance shall be tested against the mandatory and conditional elements (where the condition is true) that are described in Clauses 7 to 12.

Conformance categories are shown in Table A.1.

Table A.1 — Categories of conformance

Category	Requirements in
Conformance for coordinate metadata	A.2
Conformance of a CRS definition	A.3
Conformance of a coordinate operation definition	A.4

In each of the conformance classes below the following apply:

1) Test purpose: To determine whether all of the relevant entities and elements which are specified to be mandatory or mandatory under the conditions specified have been provided in the definition.

2) Test case identifier: Completeness test

3) Test type: capability

4) Test method: Check the entity description to ensure that it includes, as a minimum, all of the elements indicated as mandatory for that type of system and that it uses the appropriate data types for, and occurrences of, those elements.

A.2 Conformance for coordinate metadata

Requirements for conformance of a reference to a coordinate reference system are shown in Table A.2.

Table A.2 — Conformance of a coordinate set reference to a coordinate reference system

Conformance class	Description	Requirements
1	Coordinate set reference when CRS has a static reference frame or datum.	Clause 7.3.1, requirement 1.
2	Coordinate set reference when CRS has a dynamic reference frame.	Clause 7.3.1, requirement 1, and Clause 7.3.2, requirement 2.

A.3 Conformance of a CRS definition

Requirements for conformance of the definition of a coordinate reference system are shown in Table A.3. The principle requirement is shown through the table given in Column 4, but this inherits further requirements for associations, attributes and/or constraints from the tables given in Column 5. Conformance requires adherence to the requirements in all of the tables listed for that conformance class.

Table A.3 — Conformance of the definition of a coordinate reference system

Conformance class	Description	Requirements		
		Described in		Dependencies described in
		Clause	Table	Tables
3	Definition of a static geodetic CRS	9 8 10 11	10 45 51	9 5, 6, 7 28, 30, 47, 48. Also table 37 if spherical CS supported. 50, 53. Also tables 54 and 55 if ellipsoid supported.
4	Definition of a dynamic geodetic CRS	9 8 10 11	10 45 52	9 5, 6, 7 28, 30, 47, 48. Also table 37 if spherical CS supported. 50, 51, 53. Also tables 54 and 55 if ellipsoid supported.
5	Definition of a derived geodetic CRS	9 8 10 11 12	19 45 51 70	9, 10, 11, 16 5, 6, 7 28, 30, 47, 48. Also table 37 if spherical CS supported. 50, 52, 53. Also tables 54 and 55 if ellipsoid supported. 65, 68, 72, 73, 76, 79, 80
6	Definition of a static geographic CRS	9 8 10 11	11 32 51	9, 10 5, 6, 7 28, 47, 48 50, 53, 54, 55
7	Definition of a dynamic geographic CRS	9 8 10 11	11 32 52	9, 10 5, 6, 7 28, 47, 48 50, 51, 53, 54, 55
8	Definition of a derived geographic CRS	9 8 10 11 12	20 32 51 70	9, 10, 11, 16 5, 6, 7 28, 47, 48 50, 52, 53, 54, 55 65, 68, 72, 73, 76, 79, 80
9	Definition of a projected CRS	9 8 10 11 12	17 30 51 70	9, 10, 11, 16 5, 6, 7 28, 47, 48 50, 52, 53, 54, 55 65, 68, 72, 73, 76, 79, 80

Conformance class	Description	Requirements		
		Described in		Dependencies described in
		Clause	Table	Tables
10	Definition of a derived projected CRS	9 8 10 11 12	18 43 51 70	9, 10, 11, 16, 17 5, 6, 7 28, 47, 48 and at least one of 29, 30, 31, 34, 36, 37. If 34 then also 46. 50, 52, 53, 54, 55 65, 68, 72, 73, 76, 79, 80
11	Definition of a static vertical CRS	9 8 10 11	12 42 56	9 5, 6, 7 28, 47, 48 50, 57
12	Definition of a dynamic vertical CRS	9 8 10 11	12 42 58	9 5, 6, 7 28, 47, 48 50, 56, 57
13	Definition of a derived vertical CRS	9 8 10 11 12	21 42 56 70	9, 12, 16 5, 6, 7 28, 47, 48 50, 57, 58 65, 68, 72, 73, 76, 79, 80
14	Definition of a parametric CRS	9 8 10 11	13 35 59	9 5, 6, 7 28, 47, 48 50, 60
15	Definition of a derived parametric CRS	9 8 10 11 12	22 35 59 70	9, 13, 16 5, 6, 7 28, 47, 48 50, 60 65, 68, 72, 73, 76, 79, 80
16	Definition of an engineering CRS	9 8 10 11	14 44 61	9 5, 6, 7 28, 47, 48 and at least one of 29, 30, 31, 33, 34, 36, 37. If 34 then also 46. 50
17	Definition of a derived engineering CRS	9 8 10 11 12	23 44 61 70	9, 14, 16 5, 6, 7 28, 47, 48 and at least one of 29, 30, 31, 33, 34, 36, 37. If 34 then also 46. 50 65, 68, 72, 73, 76, 79, 80
18	Definition of a temporal CRS using dateTime	9 8 10 11	15 39 62	9 5, 6, 7 28, 38, 46, 47, 48 50, 63
19	Definition of a temporal CRS with temporal count	9 8 10 11	15 40 62	9 5, 6, 7 28, 38, 46, 47, 48 50, 63
20	Definition of a temporal CRS with temporal measure	9 8 10 11	15 41 62	9 5, 6, 7 28, 38, 46, 47, 48 50, 63

Conformance class	Description	Requirements		
		Described in		Dependencies described in
		Clause	Table	Tables
21	Definition of a derived temporal CRS	9 8 10 11 12	24 40/41 62 70	9, 15, 16 5, 6, 7 28, 38, 46, 47, 48 50, 63 65, 68, 72, 73, 76, 79, 80
22	Definition of a CRS with datum ensemble	11 9 8	64	50 9 5, 6, 7 plus the requirements for the relevant conformance classes 3 through 21.
23	Definition of a compound CRS	9 8	25	9 5, 6, 7

A.4 Conformance of a coordinate operation definition

Requirements for conformance of the definition of a coordinate operation are shown in Table A.4. The principle requirement is shown through the table given in Column 4, but this inherits further requirements for associations, attributes and or constraints from the tables given in Column 5. Conformance requires adherence to the requirements in all of these tables. Conformance class 24 pertains to coordinate conversions between two independent coordinate reference systems: conversions used in derived CRS definitions are included in A.3.

Table A.4 — Conformance of the definition of a coordinate operation

Conformance class	Description	Requirements		
		Described in		Dependencies described in
		Clause	Table	Tables
24	Definition of a coordinate conversion (Excludes conversions supporting a derived CRS, requirements for which are in table A.3)	12 8 9	70	65, 68, 72, 73, 76, 79, 80 5, 6, 7 8
25	Definition of a coordinate transformation	12 8 9	69	65, 68, 72, 73, 76, 79, 80 5, 6, 7 8
26	Definition of a point motion operation	12 8 9	71	65, 68, 72, 73, 76, 79, 80 5, 6, 7 8
27	Definition of a concatenated coordinate operation	12 8 9	67	65, 68 5, 6, 7 8
28	Definition of a pass-through coordinate operation	12 8 9	66	65 5, 6, 7 8

Annex B: Spatial referencing by coordinates – Geodetic concepts
(informative)

B.1 Some geodetic concepts

Coordinates are the object of this document. Point positioning is a central technological element for this. The creation and maintenance of hierarchically-ordered geodetic reference systems provides a consistent stable base for positioning and navigation.

Geodesy is the geoscience which deals with the measurement of the size and shape of the Earth, the Earth's rotation and its gravitational field, as well as with mapping its surface. The determination of the size and shape (or "figure") of the Earth includes the study of the solid and fluid Earth surfaces, their changes and deformations through Earth tides and crustal motion. Earth rotation and its temporal variations provide parameters for the transformation between celestial and terrestrial reference systems. The Earth's gravity field is related to mass-density variations within the solid earth; its spatial and temporal variations define the Earth's geocentre.

Spatial reference systems are sustainable central elements of space geodesy and astrometry against which changes are measurable. In modern geodesy the concept is called a reference system and that for the Earth is a *terrestrial reference system*. The concept is turned into actuality, or *realized*, through a *terrestrial reference frame*. Geodetic science's fundamental description of locations on the Earth is the conceptual International Terrestrial Reference System (ITRS). It has been realized through several versions of the International Terrestrial Reference Frame (ITRF).

In the terminology of this document a coordinate reference system is a coordinate system that is referenced to an object. It is the practical geodetic realizations, not their conceptual reference systems, that are called *coordinate reference systems.*

B.2 Geodetic reference surfaces

The surface of the Earth, with its topography, is highly irregular and unsuitable as the basis of spatial reference systems. A more practical reference is a surface where topographic features are removed. Such a surface that best approximates the shape of the Earth is the *geoid*. It is an idealized surface which the oceans would take in the absence of currents, air pressure variations, etc.

Vertical reference surfaces historically have been defined as mean sea level at one or more locations over a particular period of time and extended across the continents by spirit levelling and/or by measurements of gravity. More recently, a value of gravitational equipotential may have been adopted as the vertical reference surface by convention. Heights and depths are measured along the direction of gravity from such vertical reference surfaces. In this document such heights are referred to as gravity-related heights (*H*). Geodetic science distinguishes several different types of gravity-related heights, differentiated by the assumptions made about the Earth's gravity field (orthometric, Normal, Normal-orthometric, geopotential, etc.). How the gravity field is modelled and what physical quantities they represent is beyond the scope of this document.

The geoid is, however, affected by anomalies in the distribution of mass inside the Earth and hence is an irregular surface. These irregularities cause the geoid to be too complicated to readily serve as the computational surface for geometrical problems such as point positioning. To facilitate easier spatial calculations, the shape of the Earth was approximated by the nearest regular geometric solid, an oblate ellipsoid. The reference ellipsoid is a reasonably accurate approximation of the geoid which undulates

around the ellipsoid's surface with variations globally of ± 110 m. The geometrical separation between the geoid and the reference ellipsoid is called the geoid undulation or geoid height, see Figure B.1.
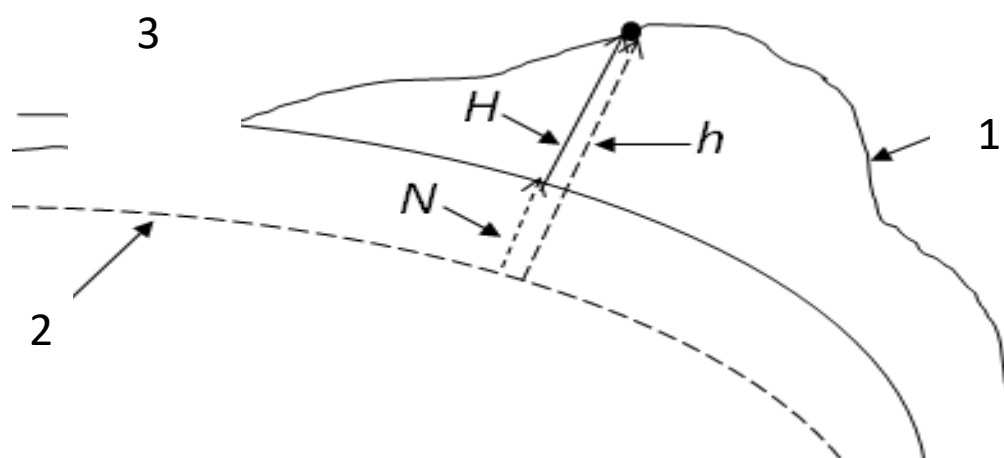
Key

1   surface of the Earth

2   ellipsoid surface

3   vertical reference surface

h ellipsoidal height, measured from ellipsoid along perpendicular passing through point

H gravity-related height, measured along direction of gravity from vertical reference surface

N geoid height or geoid undulation, height of geoid above ellipsoid

$h \approx H + N$

Note: In geodesy the conventional symbol for geoid height is N, as used here. This should not be confused with the cartographic use of the symbol N for northing.

Figure B.1 — Ellipsoidal and gravity-related heights

There is not just one ellipsoid. The size, shape, position and orientation of an ellipsoid are a matter of choice, and therefore many choices are possible. This choice of ellipsoid size, shape, position and orientation with respect to the Earth is captured by the concept of *geodetic datum*. Geodetic datums were traditionally defined such that the ellipsoid matched the surface of the geoid as closely as possible locally, for example in a country. Before the satellite geodesy era, the coordinate systems associated with geodetic datums were intended to be geocentric but, due to local deviations in the direction of the (vertical) plumbline, their origins differed from the geocentre by hundreds of metres. These regional geodetic datums, such as ED50 (European Datum 1950) and NAD27 (North American Datum 1927), have ellipsoids associated with them that are regional "best fits"' to the geoid within their areas of determination.

A change of size, shape, position or orientation of an ellipsoid will result in a change of ellipsoidal coordinates of a point on the Earth. Consequently, ellipsoidal coordinates – latitude and longitude – are only unambiguous when the geodetic datum is identified.

The position of a point relative to an ellipsoid is expressed by means of ellipsoidal coordinates: geodetic latitude (φ) and geodetic longitude (λ). The height above the ellipsoid (h) is an inseparable element of a geographic 3D coordinate tuple.Historically, it has been common practice to describe a location in 3D space through the combination of horizontal ellipsoidal coordinates for horizontal position together with

a gravity-related height for vertical position. Such a combination is an example of a compound coordinate reference system (C.2.2.3).

More recently, the use of ellipsoidal coordinates for geodetic calculations has been replaced by the use of a three-dimensional geocentric Cartesian coordinate system, *X*, *Y* and *Z*. Since the advent of satellite positioning, such coordinate systems are typically *geocentric*: the *Z*-axis is aligned with the Earth's (conventional or instantaneous) rotation axis, the *X*-axis lies within the equatorial plane and the international reference meridian (a realization of the Greenwich observatory's meridian plane, C.4.2.2)), whilst the *Y*-axis forms a right-handed coordinate system.

B.3 Dynamic and static reference frames

Historically national and regional geodetic CRSs have been realized through the coordinates of points on the surface of the Earth. Those points move with the regional tectonic plate and to an observer situated on the tectonic plate the coordinates appear to be unchanging with time. This is called a '*static reference frame*'. Examples include ETRF89 in Europe and GDA2020 in Australia.

Modern geodesy recognises that the surface of the Earth is deforming. The coordinates of points may change with time. The time evolution of coordinates may be included in the geodetic reference frame definition. Alternatively, or additionally, a crustal deformation model may be associated with the reference frame. A frame in which coordinates may change with time is called a '*dynamic reference frame*'.

When a reference frame and its coordinate reference system is fixed to the Earth as a whole, the individual tectonic plates move (albeit slowly, a few centimetres a year) within the system. To an observer on a tectonic plate the coordinates of his location within the earth-fixed Cartesian coordinate system change slowly with time – a dynamic reference frame. Examples include ITRF realizations and systems used by global satellite navigation systems such as WGS 84 used by GPS, PZ-90 by GLONASS, etc.

The tectonic plates may be subject to local crustal deformation caused by processes such as earthquake or post-glacial isostatic rebound. These deformations may be modelled, usually as a velocity grid. A CRS that has an associated deformation model is dynamic, regardless of whether it is plate-fixed or earth-fixed.

A CRS is dynamic either if it has a dynamic reference frame or if it is associated with a velocity model.

For practical application on a non-global basis, a dynamic CRS is not always convenient. Users prefer coordinates of locations on stable parts of tectonic plates to be constant. Regional and national reference frames may be defined to be fixed to the local tectonic plate with their definition adopting ITRF coordinate values at a chosen frame reference epoch. The coincidence of the two frames is only at the chosen epoch. Due to the motion of the tectonic plate to which the plate-fixed frame is fixed, the relationship between the global frame and the plate-fixed regional or national frame will then change slowly with time. Transformations between global and these national systems contain time evolution.

B.4 Epoch

B.4.1 Introduction

In this document "Epoch" is a point in time. It is given as a decimal year in the Gregorian calendar, with yyyy.00 being midnight at the start of the 1st January of year "yyyy". Should it be required to convert the epoch to or from a Gregorian calendar date, assuming any of 365, 365.25 or 366 days as the year length will be satisfactory for dealing with tectonic plate linear motion.

EXAMPLE: 2017-03-25 in the Gregorian calendar is epoch 2017.23.

Several time references are involved in dynamic CRSs:

a) *Frame reference epoch*;

b) *Coordinate epoch*;

c) *Transformation epoch, which in itself has two forms:*

   1) *transformation reference epoch* for transformations which are time-specific;

   2) *parameter reference epoch* for transformations which are time-dependent.

These are defined below.

B.4.2 Frame reference epoch

One of the attributes of a dynamic CRS is the reference epoch at which the frame's station coordinates and velocities are defined. The frame's reference epoch is a choice made in the solution's data processing. It is not necessarily the same as the year in the name of the realization, which may be the last year in which observations were made or be a target date for publication. The date in the name of the realization indicates the sequence of realizations but, other than that, has no significant meaning; it is just a name.

B.4.3 Coordinate epoch

In a dynamic CRS, coordinates of a point on the surface of the Earth may change with time. To be unambiguous the coordinates must always be qualified with the epoch at which they are valid. This is often expressed in the form "<CRS_name> at epoch T", "<CRS_name> epoch T" or "<CRS_name>@T".

EXAMPLES

ITRF2008 at epoch 2017.53

ITRF2008 epoch 2017.53

WGS 84 (G1762) @ 2017.53

It is vital to realise that in all of these examples the suffix "2017.53" refers to the coordinates. It does <u>not</u> belong to the CRS and therefore does not modify in any way the definition of ITRF2008 (which has a frame reference epoch of 2005.0) or of WGS 84 (G1762).

The coordinates of a data set may be changed to any other epoch. Plate motion or other crustal deformation models are often used for this when estimated coordinate velocities are not available. Such models facilitate for example the change of coordinates from being referenced to ITRF2008 at epoch 2017.53 to being referenced to ITRF2008 at epoch 2005.0.

B.4.4 Transformation reference epoch and parameter reference epoch

In the case of spatial coordinates, a change of coordinates between two CRSs often takes the form of a similarity or *Helmert transformation*. In the plane, a Helmert transformation has four parameters; in 3D space it has seven parameters, consisting of a rotation and scaling operation in addition to a simple translation.

For coordinate transformations between dynamic CRSs and between dynamic CRSs and static CRSs, the standard Helmert 7-parameter similarity transformation for 3D space is supplemented in one of the following two ways:

a) the transformation is valid for a specific epoch only, the *transformation reference epoch*. This is sometimes referred to as an 8-parameter transformation: the 7 Helmert parameters plus transformation reference epoch. Supplementary methods to account for crustal motion to the transformation reference epoch are required.

b) the parameters are time-dependent and the *parameter reference epoch* defines the date at which the quoted values of the 7 Helmert parameters are valid. Each of the seven parameters has a rate and has to be adjusted for any difference between the coordinate epoch and the parameter reference epoch before the transformation is applied. This is often referred to as a 14-parameter (7 parameters plus 7 rates) or, more appropriately, a 15-parameter transformation (7 parameters plus 7 rates plus the parameter reference epoch).

B.5 Map projections

Spatial calculations on the surface of an ellipsoid are not straightforward. It is considerably easier to work in plane rectangular coordinates. Such coordinates can be obtained from ellipsoidal coordinates using the artifice of a map projection. It is *not* possible to map the curved surface of an ellipsoid onto a plane map surface without deformation. The compromise most frequently chosen is to preserve angles and length ratios, so small squares are mapped as squares. This is known as a conformal projection. One example of a conformal map projection method is Transverse Mercator. Properties other than those preserved, for example scale, contain errors and the projected coordinate reference system can only be used over areas where these errors can be tolerated. Other projection methods preserve different properties, for example area.

Within the mapping plane, we have rectangular coordinates *x* and *y*. There is no global standard for axis direction: in some communities x is east, in others x is north and yet others x is south. In all cases, the north direction used for reference is the *map* north, not the *local* geodetic north. The difference between the two north directions is called the *meridian convergence*.

Annex C: Spatial referencing by coordinates – Context for modelling
(informative)

C.1 Coordinate metadata

C.1.1 Coordinates

The geometry of spatial features can be expressed in terms of invariant geometric quantities such as shapes and relative positions/orientations (strictly speaking only distance ratios and angles are invariant quantities). However, this would be impractical: performing calculations on spatial data would be a major effort. The expression of the position of a point by using coordinates introduces simplicity in terms of overview and calculus. However, there is a price to be paid for this convenience. To describe a simple shape such as a triangle in a plane, instead of one distance ratio and one angle, six coordinates are required. The inherent degrees of freedom (four in 2D, seven in 3D) have to be satisfied by choosing the origin of the coordinate axes, their unit and the orientations of the axes. This choice underlines the fact that coordinates are human-defined quantities and **not** natural phenomena. Although this may seem self-evident, it is often overlooked and has consequences for the interpretation of coordinates and their error characteristics.

The concept of a *coordinate reference system* (CRS) captures the choice of values for the parameters that constitute the degrees of freedom of the coordinate space. The fact that such a choice has to be made leads to the large number of coordinate reference systems in use around the world. It is also the cause of the little understood fact that the latitude and longitude of a point are not unique. Without the full specification of the coordinate reference system, coordinates are ambiguous at best and meaningless at worst. However, for some interchange purposes, it is sufficient to confirm the identity of the system without necessarily having the full system definition.

C.1.2 Coordinates in a dynamic CRS

Traditionally coordinates describing features on the surface of the Earth have been static, that is they do not change with time. Modern CRSs may be static, but some (including those used by navigation satellite systems) are dynamic, that is the coordinates of points on the surface of the earth may change with time. Static and dynamic CRS concepts are outlined in B.3. In a dynamic CRS the full specification of the coordinate reference system is insufficient to remove ambiguity in coordinates: the epoch for those coordinates is also required. The coordinate epoch is an attribute of the coordinates themselves, it is not part of the coordinate reference system specification.

Calculations on Geometry (ISO 19107[3]) using coordinates referenced to a dynamic CRS can be made only if the coordinates are first reduced to a common coordinate epoch. In this document, coordinates in a spatial dataset are required to be referenced to one coordinate reference system and to one coordinate epoch. This document does not permit individual coordinate tuples within one spatial coordinate set to be associated with different coordinate epochs. Before being combined into a coordinate set they must first be referenced to one chosen coordinate epoch.

C.1.3 Change of coordinate epoch

A change of coordinates from being referenced to CRS 1 at coordinate epoch 1 to being referenced to CRS 2 at coordinate epoch 2 may be achieved through three routes:

a) through two coordinate operations, first changing coordinate epoch and then changing CRS;

b) through two coordinate operations, first changing CRS and then changing coordinate epoch;

c) directly through one coordinate operation combining the operations in a) or b).

In principle these three routes commute. In practice they may not do so because the coordinate operations usually are not error free and the parameter values for each of the possible five coordinate operations may have been estimated independently. Additionally, in practice data may not be available for all of the routes, or may require change of coordinate epoch through some time other than coordinate epoch 1 or coordinate epoch 2, introducing further steps into the procedure.

C.1.4 Coordinate reference system identification

Implementers are warned that in any register, errors in the data may be corrected in accordance with rules specific to that register as defined by the responsible registration authority. The rules for dealing with erroneous data should be recognized by applications referencing the register in order to be able to find the data that is required (usually the most up-to-date register information, but sometimes the erroneous information from the past because historically it was used to transform spatial data that is still in use).

C.2 Coordinate reference system definition

C.2.1 Principal subtypes of coordinate reference system

Subtypes of coordinate reference system are defined in Clause 9.

The classification criterion for sub-typing of principal coordinate reference systems is by reference to the type of datum associated with the coordinate reference system. The following principal subtypes of coordinate reference system are distinguished:

a) **Geodetic**. A spatial coordinate reference system that is associated with a geodetic reference frame. Geodetic coordinate reference systems are either two- or three-dimensional. This document subtypes geodetic CRSs having an ellipsoidal CS as **geographic.** A geographic CRS is required be associated with an ellipsoid. A geodetic CRS with a Cartesian or spherical coordinate system usually will be, but is not necessarily, associated with an ellipsoid. A geographic CRS using 3D ellipsoidal coordinates [latitude, longitude and ellipsoidal height (h)] is used when positions are described on, above or below the ellipsoid. The geographic 2D case ignores ellipsoidal height. Ellipsoidal heights cannot exist independently, but only as an inseparable part of a 3D coordinate tuple defined in a geographic 3D coordinate reference system. A geodetic 3D CRS using three-dimensional Cartesian coordinates is used when describing positions relative to the centre of the Earth. The geodetic CRS may be static or dynamic: refer to B.3.

b) **Vertical**. A spatial coordinate reference system that is associated with a vertical reference frame. Vertical CRSs make use of the direction of gravity to define the concept of height or depth.

> NOTE   Depth is sometimes measured along a line that does not follow the vector of gravity locally. An example is depth in an oil or gas well where it is generally measured along the wellbore path. This path can vary significantly from the local vertical. Nevertheless, the distance along the wellbore path is referred to as "depth".

The surveying method through which the vertical CRS is realized is an optional attribute of the vertical CRS's reference frame. Some vertical reference frames are propagated through a geoid height model - see C.4.3. In this document the geoid height model is described as a coordinate transformation. When this method is used, the vertical reference frame realization method will be 'geoid'. Then the vertical coordinate reference system should be associated with the geoid model coordinate transformation through the HeightTransformation association of the UML model. The geodetic coordinate reference system in which the geoid height model is expressed is the source CRS for the coordinate transformation. The geoid heights may be referenced to more than one geodetic reference frame, in which case there will be more than one HeightTransformation association. The vertical CRS may be static or dynamic: refer to B.3.

c) **Engineering**. A spatial coordinate reference system that is associated with an engineering datum, used only in a contextually local sense. This subtype is used to model the following broad categories of local coordinate reference systems:

   1) systems applied to engineering activities on or near the surface of the Earth but having limited extent;

   2) coordinates on moving platforms such as road vehicles, vessels, aircraft or spacecraft.

   3) internal coordinates on images or in imagery sensors.

   For engineering CRSs on or near the surface of the Earth, "contextually local" is equivalent to "spatially local". These engineering CRSs are commonly based on a simple flat-Earth approximation of the Earth's surface: calculations on coordinates use simple plane arithmetic without any corrections for Earth curvature. A local activity is not necessarily referenced to an engineering CRS - many will be referenced to a geodetic, geographic or projected CRS. It is only those that are not so referenced that use an engineering CRS.

   Engineering CRSs used on moving platforms are usually intermediate coordinate reference systems that are computationally required to calculate coordinates referenced to geodetic or projected CRSs from other CRSs or grids applied to sensors carried on the platform. These engineering coordinate reference systems are subject to all the motions of the platform with which they are associated. In this case, "contextually local" means that the associated coordinates are meaningful only relative to the moving platform. In the spatial sense, their applicability may extend from the immediate vicinity of the platform (e.g. a moving seismic ship) to the entire Earth (e.g. in space applications). The determining factor is the mathematical model deployed in the positioning calculations. Transformation of coordinates from these engineering CRSs on moving platforms to Earth-referenced coordinate reference systems involves time-dependent coordinate operation parameters.

   For Engineering CRSs used to describe internal position on images, "contextually local" means the internal CRS for the image or image sensor. In this document a CRS has axes that have continuous numbering and coordinate increment. An ordinal coordinate system may be used when the coordinates are regularly spaced sequential indexes. An example is given in E.2.8. Grids in general, and specifically irregular grids, are described in ISO 19123[6]. The internal CRS may be georeferenced to location in a geodetic, geographic or projected CRS through a coordinate transformation, either directly or indirectly through an engineering CRS local to the sensor platform.

d) **Parametric**. Scientific communities, especially those concerned with the environmental sciences, frequently express spatial position partially in terms of a parameter or function. Within these communities, this parameter or function is treated as a coordinate. Its relationship with a spatial dimension will usually be non-linear. Examples are widespread, but latitude, longitude and pressure is a commonly encountered example; pressure is used as a proxy for height, but its relationship to height is complex. Examples of parametric coordinate reference systems are given in E.3.

e) **Temporal**. In this document a temporal CRS is defined in the same way as other principal subtypes: a coordinate system and a datum anchoring the coordinate system to an object, usually time on the Earth. This document supports temporal coordinate reference systems sufficient for spatio-temporal referencing. In this document the only recognised calendar is the Gregorian calendar with its proleptic extension as defined in ISO 8601. Other calendars and their conversions to the ISO 8601 Gregorian calendar are not supported. Temporal coordinate reference systems are described further in Annex D. Examples of temporal coordinate reference systems are given in E.4.

C.2.2   Additional subtypes of coordinate reference system

C.2.2.1   Introduction

In addition to the principal subtypes of coordinate reference systems described above, to permit modelling of certain relationships and constraints, additional subtypes are distinguished. These additional subtypes are:

a)   derived coordinate reference system;

b)   projected coordinate reference system, which is a derived coordinate reference system but treated exceptionally because of its importance in geographic information;

c)   compound coordinate reference system.

C.2.2.2   Derived coordinate reference system

Some coordinate reference systems are defined by applying a coordinate conversion to another pre-existing coordinate reference system. An example is one where the axis unit of an existing CRS has been modified. Such a coordinate reference system is called a derived CRS, and the coordinate reference system from which it was derived is called the Base CRS. In principle, all subtypes of single coordinate reference system may take on the role of either Base or Derived CRS. However a *derived CRS inherits its datum or reference frame from its base CRS*. Because CRS type is generally classified by reference to the type of datum, this inheritance means that most derived CRSs are of the same type as their base CRS. For example, if the base CRS has a datum type of parametric, the derived CRS type will inherit this parametric datum and its type will therefore be derived parametric.

A **projected coordinate reference system** is one that is derived from a CRS with a geodetic reference frame by applying the coordinate conversion known as a map projection. ProjectedCRS is modelled as an object class under its own name, rather than as a DerivedCRS of type "projected", to honour common practice which acknowledges projected CRSs as one of the most frequently encountered types of coordinate reference systems used in geographic information. Although in theory the base CRS for a projected CRS may be any geodetic CRS and the UML model shows this, in practice the base CRS of a projected CRS usually will be a geographic CRS. Then the map projection is applied to latitude and longitude ellipsoidal coordinate values.

A projected CRS may act as the base CRS for another derived CRS, (a derived projected CRS), for example the CRS underpinning a seismic bin grid. A derived coordinate reference system which has a projected coordinate reference system as its base CRS inherits the distortion characteristics of the base projected CRS, in addition to the reference frame.

The type of coordinate system that may be associated with a derived CRS usually has to be consistent with the type of coordinate system that may be associated with its baseCRS. For example a derived engineering CRS must have a CS type of engineering. Exceptions to this are CRSs with a geodetic reference frame:

a) a geodetic CRS (with a Cartesian or spherical coordinate system) may act as the base CRS for another geodetic CRS or, if the geodetic CRS definition includes an ellipsoid, for a geographic CRS having an ellipsoidal coordinate system;

b) a geographic CRS (with ellipsoidal coordinate system) may act as a base CRS for either another geographic CRS or a geodetic CRS with a Cartesian or spherical coordinate system;

c) either a geodetic CRS or (more usually) a geographic CRS may act as the base for a projected CRS

- the projected CRS must have a Cartesian coordinate system.

d) a projected CRS may act as the base CRS for a derived projected CRS

- if the derived projected CRS is 2-dimensional it may have either an affine or a Cartesian or an ordinal or a polar coordinate system. If 3D a cylindrical or spherical CS is also permitted.

The type of coordinate system that may be associated with a derived CRS is constrained through the subtyping shown in Figure 10 (Derived CRS) in Clause 9 in conjunction with Figure 12 (CS-CRS associations) in Clause 10.

If the new CRS does not inherit the datum or reference frame of the CRS through which it is defined, then it is not a derived CRS. For example:

— a national geodetic CRSs may be defined relative to one of the International Terrestrial Reference Frames and may be coincident with the ITRF at some defined frame reference epoch. Because the national CRS has its own reference frame and does not inherit that of the ITRF, it is modelled as a principal CRS and not a derived CRS. The national CRS may be related to the ITRF through a coordinate transformation.

— a geoid-based vertical CRS that is realized through the association of a geoid height model with a specified geographic CRS is *not* a derived CRS because the vertical CRS does not inherit the geodetic reference frame of the base geographic CRS but has its own vertical reference frame. In this document the geoid height model is described as a coordinate transformation. The vertical CRS is related to the coordinate transformation through a Height Transformation association, through which the source geographic CRS may be discovered. See example E.2.10 for an example of a description of a geoid-based vertical CRS and example E.5.2 for the description of its associated geoid height model.

C.2.2.3   Compound coordinate reference system

The traditional separation of horizontal and vertical position has resulted in coordinate reference systems that are horizontal (2D) and vertical (1D) in nature, as opposed to truly three-dimensional. It is established practice to combine the horizontal coordinates of a point with a height or depth from a different vertical coordinate reference system.

The coordinate reference system to which these 2D + 1D coordinates are referenced is a sequence of the separate horizontal and vertical coordinate reference systems. A temporal coordinate reference system may be added. Such a system is called a compound coordinate reference system (CCRS). It consists of a non-repeating sequence of two or more single coordinate reference systems, none of which can itself be compound, and which are independent of each other. Coordinate reference systems are independent of

each other if coordinate values in one cannot be converted or transformed into coordinate values in the other. In general, a compound CRS may contain any number of independant CRSs.

The coordinate order within a coordinate tuple that is referenced to a compound CRS follows firstly the order of the component single CRSs, and secondly within each of these the coordinates follow the component single CRSs coordinate system axis order. There is no prescribed order for the sequence of component CRSs but it is recommended that horizontal should precede vertical and that spatial should precede temporal.

When more than two systems are combined to form a compound coordinate reference system, nesting of CCRSs is not permitted; the individual single systems are aggregated together. Table C.1 gives examples of the possible composition of compound coordinate reference systems.

Table C.1 — Compound coordinate reference system

Compound CRS Type	Constituent CRS types
Spatial	Geographic 2D + Vertical
	Geographic 2D + Engineering 1D (near vertical)
	Projected 2D + Vertical
	Projected 2D + Engineering 1D (near vertical)
	Engineering (horizontal 2D) + Vertical
	Engineering (1D linear) + Vertical
Spatio-Temporal	Any horizontal spatial 2D plus temporal, for example: — Geographic 2D + Temporal Including multiple temporal systems that are independent is also permissible.
Spatio-Parametric	Any horizontal spatial 2D plus parametric, for example: — Projected 2D + Parametric Including multiple parametric systems that are independent is also permissible.
Spatio-Parametric-Temporal	Any spatio-parametric plus temporal, for example: — Geographic 2D + Parametric + Temporal

Should there be a requirement to tabulate coordinates that are not independent of each other, these should not be described as a compound CRS but instead be treated as multiple independent CRSs. For example to tabulate four coordinates latitude, longitude, easting and northing, then geographic2D + projected 2D is not permissible as a compound CRS because coordinates referenced to a projected CRS are not independent of coordinates referenced to a geographical CRS: they may be converted or transformed between the systems. A dual tabulation of latitude and longitude referenced to a geographic CRS and easting and northing referenced to a projected CRS should be made, even when the projected CRS has the geographic CRS as its base CRS.

C.3 Coordinate system

C.3.1 General

Coordinate systems are defined in Clause 10.

The coordinates of points are described in a coordinate system. A coordinate system is the set of coordinate system axes that spans the coordinate space. This concept implies the set of mathematical rules that determine how coordinates are associated with invariant quantities such as angles and distances. In other words, a coordinate system implies how coordinates are calculated from geometric elements such as distances and angles and vice versa. The calculus required to derive angles and distances from point coordinates in a mapping plane and vice versa is simple Euclidean 2D geometry. To do the same on the surface of an ellipsoid (curved 2D space) involves more complex ellipsoidal calculus. These rules cannot be specified in detail, but are implied by the geometric properties of the coordinate space.

NOTE   The word "distance" is used loosely in the above description. Strictly speaking distances are not invariant quantities, as they are expressed in the unit defined for the coordinate system. Ratios of distances are invariant.

C.3.2 Cartesian coordinate system

Cartesian coordinate system is modelled as a special case of an affine coordinate system (Figure 11). The UML model of the Coordinate System associations to Coordinate Reference System in Figure 12 shows both affineCS and CartesianCS in the EngineeringCS and DerivedProjectedCS union classes. This is strictly unnecessary as the presence of affine implies its subtype Cartesian. CartesianCS has been included in the EngineeringCS and DerivedProjectedCS classes to emphasise that engineering CRSs and derived projected CRSs may have a CartesianCS.

C.3.3 Coordinate system axis

Coordinate system axes are described in 10.5.

The concept of coordinate axis requires some clarification. Consider an arbitrary x, y, z coordinate system. The x-axis may be defined as the locus of points with $y = z = 0$. This is easily enough understood if the x, y, z coordinate system is a Cartesian system and the space it describes is Euclidean. It becomes a bit more difficult to understand in the case of a strongly curved space, such as the surface of an ellipsoid, where its geometry is described by an ellipsoidal coordinate system (2D or 3D). Applying the same definition by analogy to the curvilinear latitude and longitude coordinates, the latitude axis would be the prime meridian and the longitude axis would be the equator, which is not a satisfactory definition.

Bearing in mind that the order of the coordinates in a coordinate tuple is required to be the same as the defined order of the coordinate axes, the "ith" coordinate axis of a coordinate system is defined as the locus of points for which all coordinates with sequence number not equal to "i", have a constant value locally (whereby $i = 1...n$, and n is the dimension of the coordinate space).

The addition of the word "locally" in this definition apparently adds an element of ambiguity and this is intentional. However, the definition of the coordinate parameter associated with any axis has to be unique. The coordinate axis itself should not be interpreted as a unique mathematical object but the associated coordinate parameter should.

EXAMPLE 1   Geodetic latitude is defined as the "angle from the equatorial plane to the perpendicular to the ellipsoid through a given point, northwards usually treated as positive". However, when used in an ellipsoidal coordinate system the geodetic latitude axis will be described as pointing "north". At two different points on the ellipsoid, the direction "north" will be a spatially different direction, but the concept of latitude is the same.

The specified direction of the coordinate axes is often only approximate. This may lead to the two uses of the coordinate system being slightly rotated with respect to each other:

EXAMPLE 2   Two geodetic coordinate reference systems that make use of the same ellipsoidal coordinate system will usually be associated with the Earth through two different geodetic reference frames with different origins and different orientations.

EXAMPLE 3   A Cartesian coordinate system might be applied at each of two buildings, in each case orientated along one side of the building. If the two buildings are rotated relative to each other, so too will be the two coordinate systems.

The AxisUnit class contains four attributes. One of the quantities temporalCount, temporalMeasure or temporalString is used for a temporal CS axis. AxisUnitID is used for non-temporal coordinate system axes.

C.4 Datum and Reference Frame

C.4.1 General

Datums are defined in Clause 11. Particularly in a geodetic context, the modern term is now reference frame. In older geodetic terminology (pre-dating the satellite era) the origin of a survey network was termed the datum point. This term remains in use for non-geodetic purposes. In this document 'Datum' is used as the name of the generalized class in the UML model for all subtypes of datum and reference frame.

A datum or reference frame specifies the relationship of a coordinate system to an object thus creating a coordinate reference system. The datum or reference frame implicitly (occasionally explicitly) contains the values chosen for the set of parameters that represents the degrees of freedom of the coordinate system, as described in C.1.1. A datum or reference frame therefore implies a choice regarding the origin and orientation of the coordinate system.

C.4.2 Geodetic reference frame

C.4.2.1   General

A geodetic reference frame is used with three-dimensional or horizontal (two-dimensional) coordinate reference systems. It is used to describe large portions of the Earth including the entire Earth. It requires a prime meridian definition and when used in a geographic CRS an ellipsoid definition. When used in a geodetic CRS, the provision of ellipsoid is optional but recommended - see C.4.2.3.

C.4.2.2   Prime meridian

A prime meridian defines the origin from which longitude values are specified. Most modern geodetic reference frames use as their prime meridian the Bureau Internationale de l'Heure (BIH) Zero Meridian, sometimes referred to as the IERS Zero Meridian or the International Reference Meridian. This is a 1980s realization of the meridian through Greenwich, and replaces earlier definitions. In this document the concept is primarily used to describe longitude offsets determined between the then international standard and other national standards, for example the longitude offset between the Greenwich meridian and the Paris meridian. In this document the term 'Greenwich meridian' is a synonym for the then current international meridian.

In this document when the Datum subtype is geodetic reference frame the prime meridian must be identified. It must be explicitly stated when it is not the international standard, for example if it is Ferro or Batavia (Jakarta). The prime meridian need not be explicitly stated when it is the international standard; if not explicitly stated it is assumed to be the international standard, i.e. 'Greenwich'.

C.4.2.3   Ellipsoid

A reference ellipsoid is defined such that it approximates the surface of the Earth. Because of the area for which the approximation is valid – traditionally regionally, but with the advent of satellite positioning often globally – the ellipsoid is typically associated with geodetic or geographic and, indirectly, projected CRSs.

If the geodetic reference frame is combined with an ellipsoidal coordinate system as a geographic CRS, an ellipsoid is required to be specified. An ellipsoid is optional for a geodetic CRS with other types of coordinate system (Cartesian, spherical). However, its inclusion is strongly recommended, because although the definition of a geodetic CRS using a geocentric Cartesian coordinate system apparently obviates the need of an ellipsoid, the ellipsoid may play a role in the definition of the orientation of the coordinate system. Reference to a prime meridian is non-sensical without an ellipsoid.

An ellipsoid model of the Earth can be defined by either its semi-major axis and inverse flattening, or by its semi-major axis and semi-minor axis. The second parameter may be derived from other defining parameters. For some applications, for example small scale mapping in atlases, a spherical approximation of the Earth's surface is used, requiring only the radius of the sphere to be specified.

In the UML model, these options are modelled by a mandatory attribute "semiMajorAxis" in the class "Ellipsoid", plus a "secondDefiningParameter" attribute. That attribute uses the SecondDefiningParameter class with the stereotype "Union", meaning that one, and only one, of its attributes is used by an object. That class allows specification of the semiMinorAxis or inverseFlattening as the second defining ellipsoid parameter, or can specify that a spherical model is used. For a sphere, the attribute "semiMajorAxis" of the "Ellipsoid" class is interpreted as the radius of the sphere.

This document also permits a triaxial reference ellipsoid to be described using an additional semi-median axis attribute. This attribute is for planetry applications and is not used when describing a bi-axial oblate reference ellipsoid model of the Earth. For a triaxial reference ellipsoid it is usual for the secondDefiningParameter to be the ellipsoid's semi-minor axis.

A reference ellipsoid specification is not to be provided if the Datum subtype is not geodetic reference frame or dynamic geodetic reference frame. It is mandatory if the associated CRS's coordinate system is ellipsoidal, for other permitted types of coordinate system the reference ellipsoid specification is optional, but recommended.

C.4.3 Vertical reference frame

A vertical reference frame is a reference surface (the geoid) that is realized through levelling or gravity measurements. Different types of heights can be referenced to the same vertical reference surface. The geodetic distinctions between dynamic heights, orthometric heights, Normal heights and Normal-orthometric heights are not discussed in this document: all are grouped as 'gravity-related'.

The following methods of realization of a vertical reference frame may be distinguished:

a) **Levelling.** The zero value of the associated (vertical) coordinate system axis is defined at one or more tide gauges monitoring sea level over a period of time and then promulgated locally or regionally through a levelling network;

b) **Geoid**. The zero value of the associated (vertical) coordinate system axis is chosen to approximate mean sea level in some way, usually by convention. The vertical reference surface is expressed as a model of the heights of the geoid surface above or below a reference ellipsoid (geoid heights) in one or more geodetic reference frames. When this method is used, in this document the geoid height model is described as a coordinate operation, with the coordinate operation's source CRS as the geodetic reference system to which the model is related. The vertical reference frame is associated with the geoid height model through a HeightTransformation association between the frame's vertical CRS and the coordinate operation - see C.2.1 b).

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

C.4.4 Dynamic reference frames

Geodetic and vertical reference frames may be either *static* or *dynamic*. These are terms are made from the viewpoint of an observer on a tectonic plate on the surface of the Earth. Further information is given in B.3. Both geodetic and vertical reference frame types are modelled with a dynamic subtype which has a mandatory attribute, the frame reference epoch of the realization. By implication, if the geodetic or vertical reference frame subtype is not dynamic then it is static. To be unambiguous, coordinates referenced to a coordinate reference system having a dynamic reference frame must also be qualified by their coordinate epoch.

C.4.5 Parametric datum

If a parameter such as atmospheric pressure is the basis for the definition of the origin of a datum, then the datum type is *parametric*, not vertical.

C.4.6 Engineering datum

An engineering datum is used in a local context only. It describes the origin of an engineering (or local) coordinate reference system. It is stressed that the engineering datum does not necessarily describe the origin of the engineering CRS with respect to the Earth, but only relative to other points in its domain of validity, be that a moving platform, a building or an area on or near the surface of the Earth, or an image. The relationship of the engineering CRS with any geodetic or projected CRS can only be described by means of a coordinate operation.

C.4.7 Datum ensemble

Modern geodetic reference frames may be updated from time to time. The differences between successive realizations may be at the sub-decimetre level. For some geographic information applications this is insignificant and dealing with multiple coordinate reference systems that are insignificantly different is an unwanted overhead that in such applications offers no benefit. To address this issue, this document describes the artificial concept of a *datum ensemble*. Implementations that merge coordinate sets may evaluate the members of a datum ensemble and omit executing coordinate transformations between CRSs within that datum ensemble.

A datum ensemble is a group of closely related realizations. They must be realizations of the same terrestrial reference system or same vertical reference system. In the UML model the attribute name 'conventionalRS' is used to permit grouping of any subtype of reference frame or datum.

EXAMPLE

	Reference frame ID	value of 'conventionalRS' attribute
1	WGS 84 (G1674)	WGS 84
2	NAD83(CSRS)v3	NAD83(CSRS)
3	NAD83(2007)	NAD83(NSRS)
4	NAD83(CSRS)v6	NAD83(CSRS)
5	WGS 84 (G1762)	WGS 84
6	NAD27	(null)
7	NAD83(CSRS)v7	NAD83(CSRS)

— Reference frames 1 and 5 share the same conventionalRS and therefore may be combined into a datum ensemble.

— Reference frames 2, 4 and 7 share the same conventionalRS and therefore may be combined into one or more datum ensembles; permissible permutations are (2 and 4), (2 and 7), (4 and 7) and (2, 4 and 7).

— Reference frames 1 and 2 have different conventionalRSs and therefore may not be both included in the same datum ensemble.

— Reference frames 3 and 6 may not be included in a datum ensemble as no other frames [in these examples] share their conventionalRS. (Reference frame 6 has no associated conventionalRS so its value is not populated)

A datum ensemble acts as a surrogate datum in that it may be associated with a coordinate system to define a CRS.

A datum ensemble is comprised of multiple component refernce frames. These will all have identical ellipsoid and prime meridian attributes. An implementation reporting the description of the datum ensemble need not repeat these attributes but select them from any one of the component members: see example E.2.5.

The datum ensemble construct comes with the following warning. Data referenced directly to a CRS having a datum ensemble is approximate, to the stated *ensembleAccuracy*. If data is associated with a CRS having a datum ensemble, it will not be possible to identify which of the datum ensemble members the data might more accurately be referenced to. In geodesy or other high accuracy applications, datum ensembles should not be used; individual reference frames should be identified.

C.5 Coordinate operation

C.5.1 General characteristics of coordinate operations

Coordinate operations are defined in Clause 12.

If the relationship between any two coordinate reference systems is known, coordinate tuples can be *transformed* or *converted* to another coordinate reference system. The UML model therefore specifies a source and a target coordinate reference system for such coordinate operations.

A coordinate operation is often popularly said to *transform coordinate reference system A into coordinate reference system B*. Although this wording may be good enough for conversation, it should be realized that coordinate operations do **not** operate on coordinate reference systems, but on coordinates. This is important for the design of implementation specifications because it implies that a coordinate reference system cannot be created from another coordinate reference system by a coordinate operation. Neither can a coordinate operation be used to modify the definition of a coordinate reference system, for example by converting the units of measure of the coordinates. In all these cases, the source and target coordinate reference systems involved have to exist before the coordinate operation can exist.

The UML model also specifies an Interpolation CRS. This is the identifier of the CRS to be used for grid interpolation for coordinate operations in which it is neither source CRS or target CRS. An example is a transformation involving vertical offsets interpolated from a grid. The source and target CRSs will both be vertical CRSs (for example NGVD29 and NAVD88 in the USA), the interpolation CRS is a geographic CRS (for example NAD83). When the grid is referenced to the source CRS, as in the case of a geoid or height correction model, use of Interpolation CRS is not required. The source CRS takes that role.

In this document, three subtypes of single coordinate operation are recognized:

a) *Coordinate conversion* (CC) – mathematical operation on coordinates in which there are no parameters or in which the parameter values are defined mathematical constants rather than empirically derived. Application of the coordinate conversion introduces no error into the output coordinates. The application of a coordinate conversion does not involve any change of datum. Coordinate conversions are most frequently encountered as part of a derived CRS definition. The most frequently encountered type of coordinate conversion is a map projection;

b) *Coordinate transformation* (CT) – mathematical operation on coordinates in which the parameter values are empirically derived. This means that they contain observational error, and when the coordinate transformation is applied to a coordinate set presumed to be error-free, the output coordinate set will no longer be error free. The magnitude of the error is indicated by the *coordinateOperationAccuracy.* The stochastic nature of the parameters may result in several different versions of the same coordinate transformation. Multiple coordinate transformations may then exist for a given pair of coordinate reference systems, differing in their method, parameter values and accuracy characteristics;

c) *Point motion operation* (PMO) – mathematical operation within one coordinate reference system to account for the motion of a point through the coordinate space. This subtype is classed as a coordinate operation for modelling convenience. It has a constraint that requires the target CRS to be the same as the source CRS. The parameter values for a point motion operation are usually empirically derived through modelling. This means that they contain observational error.

In application the distinction between coordinate transformation and coordinate conversion usually manifest itself through the way in which they are described. A coordinate transformation description has the structure:

— Source CRS ID

— Target CRS ID

— Single operation method and parameters

A coordinate conversion typically will be part of the description of a derived coordinate reference system, which would have the structure:

— Base CRS datum component

— Single operation method and parameters

— Derived CRS coordinate system component

In this structure the coordinate conversion's source and target CRSs are implied: the base CRS acts as the source CRS for the coordinate conversion, and the derived CRS takes the role of the target CRS. The best-known example of this source-derived relationship is a projected coordinate reference system, which is always related to a base geodetic coordinate reference system. The associated map projection effectively **defines** the projected coordinate reference system from the geodetic coordinate reference system. This concept is modelled as an aggregation between (derived) coordinate reference system and coordinate conversion.

Once the parameter values are obtained, coordinate conversion, coordinate transformation and point motion operation use similar mathematical processes. In all three cases the coordinate operation method and parameters are as described in the coordinate operations UML diagram part 2.

C.5.2 Coordinate operation method and parameters

The algorithm used to execute a coordinate operation is defined in the coordinate operation method. Each coordinate operation method uses a number of parameters (although some coordinate conversions use none), and each coordinate operation assigns a value to these parameters. It is critical that the parameters and their values are consistent with the method's formula. Several superficially similar methods are in detail distinctly different. Different parameter values may then be required.

Although parameter values are usually numbers, for some coordinate operation methods, notably those implementing a grid interpolation algorithm, the parameter value could be a file name and location (this may be a URI). An example is the NADCON coordinate transformation from NAD 27 to NAD 83 in the USA in which one set of a series of sets of grid files is used.

It is recommended to make extensive use of identifiers, referencing well-known registers wherever possible. There is as yet no standard way of spelling or even naming the various coordinate operation methods. Client software requesting a coordinate operation to be executed by a coordinate transformation server implementation may therefore ask for a coordinate operation method which this server does not recognize, although a perfectly valid method using a different name may be available. The same holds for coordinate operation parameters used by any coordinate operation method.

To facilitate recognition and validation, it is recommended that the coordinate operation method formulae be included or referenced in the relevant object, if possible with a worked example.

NOTE   Concatenated coordinate operations and pass-through coordinate operations list single coordinate operations and themselves do not require a coordinate operation method to be specified.

C.5.3 Parameter groups

Some coordinate operation methods require that groups of coordinate operation parameters be repeatable as a group. Also, some coordinate operation methods may utilize a large number of coordinate operation parameters. In such cases, it is helpful to group related parameters. Each coordinate operation parameter group consists of a collection of coordinate operation parameters or nested coordinate operation parameter groups. Two or more coordinate operation parameter groups are then associated with a particular coordinate operation method.

This way of modelling is not mandatory. All coordinate operation parameters may be assigned directly to the coordinate operation method.

C.5.4 Concatenated coordinate operation

A concatenated coordinate operation is a non-repeating sequence of coordinate operations. This sequence of coordinate operations is constrained by the requirement that the target coordinate reference system of each step is required to be the same as the source coordinate reference system of the next step. The source coordinate reference system of the first step and the target coordinate reference system of the last step are the source and target coordinate reference systems specified for the concatenated coordinate operation.

The concatenated coordinate operation class is primarily intended to provide a mechanism that forces application software to use a preferred path to change coordinates from source to target coordinate reference system when a direct transformation between the two is not available.

C.5.5 Pass-through coordinate operation

Coordinate operations require input coordinate tuples of certain dimensions and produce output tuples of certain dimensions. The dimension of the source coordinate reference system need not be the same as that of the target source coordinate reference system.

The pass-through coordinate operation specifies what subset of a coordinate tuple is subject to a requested coordinate operation. It takes the form of referencing another coordinate operation and specifying a sequence of numbers defining the positions in the coordinate tuple of the coordinates affected by that coordinate operation.

NOTE   The ability to define compound coordinate reference systems combining two or more other coordinate reference systems introduces a difficulty. For example, it may be required to transform only the horizontal or only the vertical component of a compound coordinate reference system, which will put them at odds with coordinate operations specified for either horizontal or vertical coordinates only. To the human mind, this is a trivial problem, but not so for coordinate transformation software that ought to be capable of automatic operation, without human intervention; the software logic would be confronted with the problem of having to apply a coordinate operation expecting two-dimensional CRSs to (2 + 1) = three-dimensional coordinate tuples.

C.5.6 RegisterOperations

Two functions
```
findCoordinateReferenceSystem(CharacterSequence) : CoordinateReferenceSystem
```
and
```
findCoordinateOperation(CharacterSequence) : CoordinateOperation
```

are for retrieving the definition of a CRS or a coordinate operation from a geodetic registry. The registry is identified through the authority association to CI_Citation.

The `CharacterSequence` arguments are codes in the namespace of this registry. If these codes are numeric, implementations should parse the character sequences as numbers before using them as the primary key for registry lookup.

EXAMPLE 1   If the `authority` is EPSG, then an example of valid function call is:
```
CoordinateReferenceSystem crs = findCoordinateReferenceSystem("4326");
```

EXAMPLE 2:   If the authority is OGC, then an example of valid function call is:
```
CoordinateReferenceSystem crs = findCoordinateReferenceSystem("CRS84");
```

C.5.7 Implementation considerations

This explanation of coordinate operations is not complete without giving some thought to their implementations. Coordinate transformation services should be able to automatically derive coordinate operations that are not stored explicitly in any permanent data store, in other words determine their own concatenated or inverse operations. The reason is that it is practically impossible to store all possible pairs of coordinate reference systems in explicitly defined coordinate operations. The key to a successful software implementation is the ability to apply meaningful constraints and validations to this process. For example, it may be mathematically possible to derive a concatenated coordinate operation that will transform North American Datum of 1927 coordinates to Australian Geodetic Datum of 1966 coordinates but, in a practical sense, that operation would be meaningless. The key validation that would flag such a coordinate operation as invalid would be a comparison of the two domains of validity and the conclusion that there is no overlap between them.

Coordinate transformation services should also be able to derive or infer from a forward coordinate operation ("A" to "B") the inverse or complementary coordinate operation (from "B" to "A"). Most permanent data stores for coordinate reference parameter data will record only one of these two coordinate operations. The logic to derive the inverse coordinate operation should be built into the application software that performs the coordinate operation, be it server or client.

In some cases, the algorithm for the inverse coordinate operation is the same as the forward algorithm, and for the inverse operation to be fully defined only the signs of the parameter values need to be reversed. An example is the 7-parameter Helmert transformation (both position vector and coordinate frame rotation convention).

Some polynomial coordinate operation methods require the signs of only most, but not all, parameter values to be reversed. Other coordinate operation methods imply two algorithms, one for the forward and one for the inverse coordinate operation. The parameters and their values are generally the same in that case. The latter situation generally applies to map projections.

Finally, the same algorithm may be used for the inverse coordinate operation, with entirely different parameter values. This is the case with some polynomial and affine coordinate operation methods. In those cases, the inverse coordinate operation cannot be inferred from the forward coordinate operation but has to be explicitly defined.

Annex D: Temporal referencing by coordinates – Context for modelling
(informative)

D.1 General

ISO 19108[4] describes three forms of temporal reference system:

— calendars;

— ordinal scales;

— temporal coordinate systems.

Calendars have a variety of complex internal structures defined through a set of rules for composing the calendar date and time. Ordinal scales provide a basis for measuring only the relative position of temporal objects, for example geological eras. Calendars and ordinal scales are out of scope of this document.

NOTE   This document and ISO 19108:2002 use the term *temporal coordinate system* to describe different concepts. A 19108:2002 temporal coordinate system maps to this document's temporal coordinate reference system.

In this document a temporal coordinate reference system is a temporal coordinate system which is related to the Earth through a temporal datum. The datum defines the origin of the temporal coordinate system with respect to a calendar. In this document the only calendar supported is the Gregorian calendar with its Proleptic extension, as defined in ISO 8601. However the Calendar class codelist allows implementations to extend to alternatives not supported by this document, for example providing the name of a calendar for Mars.

This document describes three options for temporal coordinate systems:

a) dateTime;

    1) a value expressed as a DateTime in conformance with ISO 8601.

b) temporal count;

    1) discrete temporal quantity, expressed as an integer. A unit of measure is included;

    2) the time axis is with respect to the calendar defined in the temporal datum.

c) temporal measure;

    1) a continuous temporal quantity, expressed as a real value. A unit of measure is included;

    2) the time axis is with respect to the calendar defined in the temporal datum.

This document recognises both temporal count and temporal measure because for temporal count unambiguous conversion to dateTime is usually (but not always) possible whereas for temporal measure unambiguous conversion to dateTime generally is not possible; see the following sections of this Annex. Knowledge of this distinction *a priori* may be useful for implementation.

See E.4 for examples of TemporalCRS instances.

D.2 Temporal Units of Measure

Axis unit uses the datatype of UnitOfMeasure. This is defined in ISO 19103. The class includes a note "conversion ToISOstandardUnit is not null only if the conversion is a simple scale". For many temporal cases, the unit is not a simple scale, as the size of a month, a day or an hour vary at different locations in the calendar due to corrections factors and alterations such as leap seconds, leap years, and seasonal time zone changes. Conversion of a temporal quantity (unit of measure) to the SI base unit for time, the second, therefore may or may not be ambiguous when compared to a calendar definition of that quantity. Consequently, UnitOfMeasure instances for temporal counts and temporal measures may be defined with no relation to the second.

NOTE   In ISO 8601 the terms 'calendar day', 'calendar month' and 'calendar year' are used, with the note: often referred to as 'day', 'month' and 'year' respectively.

In the DateTimeTemporalCS case only, axis unit is prohibited. The dateTime syntax is a representation of a compound string including multiple units. Its components are defined in ISO 8601. This requirement prohibition is modelled through the DateTimeCoordinateSystemAxis class.

POSIX time is commonly used in software. It is dimensioned in seconds, but leap seconds are ignored (not applied)[13]. A unit of measure "second" may be used to represent this, but it is required to be defined independent of the SI second, not as a specific number of SI seconds. It may be thought of as a "calendar second".

D.3 Reduced Precision

ISO 8601 defines syntax for dateTime instances with reduced precision. A temporal datum may use reduced precision to define its origin. For example a datum used in a temporal CRS for decimal years in the common era may choose to define its datum origin as 0000, a representation with precision reduced to only years.

Individual dateTime coordinate values may also use reduced precision.

D.4 Calendar Arithmetic

D.4.1 General

Calendar Arithmetic is the process of adding or subtracting a temporal quantity, an offset from a DateTime, to calculate a new DateTime, whilst taking account of all of the calendar corrections.

Calendars define time through periodic and quasi-periodic quantities, together with corrections to particular instances of those quantities at specified points in the calendar. Leap seconds, leap years and seasonal time adjustments are all examples of corrections.

The definition of calendar arithmetic and the standardisation of results is not part of this document. Implementations of this standard may implement calendar arithmetic to enable the conversion of dimensioned offsets to dateTimes, but the results from different implementations in a variety of cases may differ (examples below).

Conversions between dateTimes and temporal quantities that are real values (temporal measures) are complicated, as context specific calculations and approximations for decimal remainders are generally required. This document does not support conversions to recalculate coordinate values using a different temporal measure unit or convert temporal measures to an ISO 8601 dateTime string.

Conversions between dateTimes and integer temporal counts can make use of integer arithmetic, reducing the complication significantly. In general integer calendar arithmetic returns consistent results and implementations may reasonably expect to agree on results. However there remain areas where some calculations are ambiguous or not well defined and caution is required. In such cases implementations may choose to return null results or raise exceptions.

To assist with implementation, some examples of definitive calendar arithmetic with integers are provided in D.4.2. Examples of ambiguous arithmetic, with integers and real values, are provided in D.4.3. These examples are illustrative, other interpretations of the ambiguous cases may also be plausible.

D.4.2 Definitive Calendar Arithmetic

Examples of unambiguous calendar arithmetic are:

a)   A DateTime offset by 25 months from a datum origin of 2012-01

—   2014-02

b)   A DateTime offset by 25 days from a datum origin of 2000-12-01T00:00Z

—   2000-12-25T00:00Z

c)   A DateTime offset by 31536000 hours from a datum origin of 1900-01-01T00Z

—   2006-04-01T00Z

d)   A DateTime offset by 1483228815 SI seconds from a datum origin of 1970-01-01T00:00:00Z

—   2016-12-31T23:59:48Z

e)   A DateTime offset by 1483228815 calendar seconds from a datum origin of 1970-01-01T00:00:00Z

—   2017-01-01T00:00:15Z (derived from POSIX formula[13])

f)   A DateTime offset by 5351236450450 SI microseconds from a datum origin of 2016-12-01T00:00:00.000000Z

—   2017-01-31T22:27:15.450450Z

g)   A DateTime offset by 5351236450450 calendar microseconds from a datum origin of 2016-12-01T00:00:00.000000Z

—   2017-01-31T22:27:16.450450Z (derived from POSIX formula[13])

Note the comparison between each of the last two pairs of examples.

D.4.3 Ambiguous Calendar Arithmetic

Examples of ambiguous calendar arithmetic are:

a) A DateTime offset by 1 month from a datum origin of 2016-01-30

 — The accuracy of the datum origin is day, whilst the unit of the offset is month. The day number is greater than the minimum. It is not clear how to interpret 1 month on from the 30th of January in any year, and a further complication is that the year may be a leap year. Plausible interpretations include:

 — 2016-02-29 (the largest allowed value but still less than 30)

 — 2016-02-28 (the largest allowed value -1, the last but one day of the month)

b) An offset of 25.1 months as a dateTime from a datum origin of 2012-01-01

 — The interpretation of 0.1 of a month within the calendar is open to interpretation, as is the level of accuracy to which the result should be given. Plausible interpretations include:

 — 2014-02-02 (25 months + the floor of 0.1 of a month of 28 days in days)

 — 2014-02-03 (25 months + the round of 0.1 of a month of 28 days in days)

 — 2014-02-02T19:12 (25 months + a time estimation based on the remainder)

c) An offset of 31536000.146 hours from a datum origin of 1900-01-01T00:00:0Z

 — The interpretation of 0.146 hours within the calendar is open to interpretation, as is the resolution to which the result should be given. Plausible interpretations include:

 — 2006-04-01T00:08:45.6 (all hours the same size, conversion to decimal seconds)

 — 2006-04-01T00:08:45 (all hours the same size, floor conversion to integer seconds)

 — 2006-04-01T00:08:46 (all hours the same size, round conversion to integer seconds)

 — 2006-04-01T00:09:45.6 (northern seasonal time adjustment 1 hour forward)

d) A temporal duration in hours from 0 hours to 24 hours from a datum origin of 2017-11-05T12:00:00

 — The time zone is not quoted, so the assumption in ISO 8601 is that this is local time, but the locale is not provided. In New York, USA, there was a seasonal clock change, adding an hour, so the elapsed time is 25 hours. In Wellington, New Zealand, there was also a seasonal clock change skipping an hour, so the elapsed time is 23 hours. In London, UK, there was no seasonal clock change on this date, so the elapsed time is 24 hours.

e) A temporal duration from 2011.163 years to 2012.163 years, from a datum origin of 0000-01-01T00:00, to be expressed as a dateTime

— 0.163 years within the calendar is open to interpretation. Plausible interpretations include:

— (2011-02-28T12:00, 2012-02-28T12:00)

— (2011-03-01T00:00, 2012-02-29T00:00)

f) 367 days [nameStandardUnit=second, scaleToStandardUnit=86400.0] from a datum origin of 2016-01-01T00:00:00Z, as a dateTime

— It is unclear whether to use the integer days as a count, or whether to convert the days to seconds, as per the definition of the unit of measure, and count using these. 2016 was a leap year, and there was a leap second at the beginning of 2017 so using days as a count will result in a different day from using days converted to seconds, which will not count the leap second.

Annex E: Examples (informative)

Several examples are given below to illustrate how this document can be applied when defining a coordinate reference system or coordinate transformation. The examples give both UML identifier and attribute name. For digital data processing purposes, the UML identifier should be used. When presenting coordinate reference system metadata to human beings, the attribute name should be given. The following examples are given:

Examples of identification of a coordinate reference system
E.1.1   CRS identification through a uniform resource identifier (URI)
E.1.2   CRS identification with all required attribute values referenced through a citation

Examples of definition of spatial coordinate reference systems
E.2.1   Geodetic CRS with dynamic reference frame ('Dynamic CRS')
E.2.2   Geodetic CRS with static reference frame ('Static CRS')
E.2.3   Derived geographic 3D CRS
E.2.4   Geographic 2D CRS
E.2.5   Geographic CRS with datum ensemble (and multiple usage entries)
E.2.6   Projected CRS (2D)
E.2.7   Projected CRS (3D)
E.2.8   Derived Projected CRS
E.2.9   Vertical CRS
E.2.10   Geoid-based vertical CRS
E.2.11   Compound CRS (projected + vertical)
E.2.12   Engineering CRS applied to a construction site
E.2.13   Engineering CRS applied to a moving object
E.2.14   Engineering CRS applied to an image

Examples of definition of parametric coordinate reference systems
E.3.1   Parametric CRS using a parameter (pressure).
E.3.2   Parametric CRS using a function (potential vorticity)
E.3.3   Spatio-parametric compound CRS

Examples of definition of temporal coordinate reference systems
E.4.1   Temporal CRS in which axis quantity is a string
E.4.2   Temporal CRS in which axis quantity is a count
E.4.3   Temporal CRS in which axis quantity is a measure
E.4.4 and E.4.5 Compound CRS including two temporal CRSs

Examples of definition of coordinate operations
E.5.1   Coordinate transformation
E.5.2   Geoid height model
E.5.3   Concatenated operation

Examples of description of change of coordinate through point motion operations
E.6.1   Change of coordinate using station velocities
E.6.2   Change of coordinate using velocity model

E.1 Identification of a coordinate reference system

These examples describe how a coordinate set may be associated with a CRS definition indirectly through reference to a full description held in a geodetic parameter registry. The two examples given here are defined in full in example E.2.6.

Example E.1.1: Coordinate reference system with all required attribute values identified through a reference to a geodetic registry web address (URI or URN).

The CRS may be identified through a web URI:
http://www.opengis.net/def/crs/EPSG/0/26734

Example E.1.2: Coordinate reference system with all required attribute values referenced through a citation to a geodetic registry which defines all of the coordinate reference system, datum, coordinate system and coordinate conversion information for this projected coordinate reference system. Citations are described in ISO 19115-1.

Attribute	UML identifier	Data Entry	Comment
	CRS		
	CI_Citation		Citation is documented in ISO 19115-1.
Citation title:	title:	EPSG v6.6	
Citation date type:	dateType:	003	This is a revision date.
Citation date:	date:	20041023	
Citation identifier:	identifier:	26734	This is the unique identifier (code) for the CRS as given within the citation.
Online resource linkage:	Online resource linkage:	https://www.epsg-registry.org	

E.2 Definition of spatial coordinate reference systems

Example E.2.1:   Definition of a Geodetic CRS with dynamic reference frame ('Dynamic CRS').

Attribute	UML identifier	Data Entry	Comment
	GeodeticCRS		
Geodetic CRS name:	name:	ITRF2008 - XYZ	
CRS scope:	scope:	Spatial referencing	
CRS validity:	domainOfValidity:	World	
CRS remarks:	remarks:	Replaces ITRF2005, replaced by ITRF2014	
	CartesianCS		
Cartesian coordinate system name:	name:	ECEF right-handed	
	CoordinateSystemAxis		The order of the axes is significant.
Coordinate system axis name:	name:	Geocentric X	
Coordinate system axis abbreviation:	axisAbbrev:	X	
Coordinate system axis direction:	axisDirection:	In the equatorial plane from the centre of the Earth towards the intersection of the equator with the prime meridian.	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	Geocentric Y	
Coordinate system axis abbreviation:	axisAbbrev:	Y	

Attribute	UML identifier	Data Entry	Comment
Coordinate system axis direction:	axisDirection:	In the equatorial plane from the centre of the Earth towards the intersection of the equator and the meridian π/2 radians eastwards from the prime meridian.	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	Geocentric Z	
Coordinate system axis abbreviation:	axisAbbrev:	Z	
Coordinate system axis direction:	axisDirection:	From the centre of the Earth parallel to its rotation axis and towards its north pole.	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	DynamicGeodetic ReferenceFrame		
Dynamic geodetic reference frame name:	name:	International Terrestrial Reference Frame 2008	
Terrestrial reference system:	conventionalRS:	ITRS	
Dynamic geodetic reference frame anchor definition:	anchorDefinition:	Origin is defined in such a way that it has zero translations and translation rates with respect to the mean Earth center of mass, averaged by the SLR station positions time series. Scale is defined by nullifying the scale factor and its rate with respect to the mean of VLBI and SLR long-term solutions as obtained by stacking their respective time series. Orientation (at epoch 2005.0) and its rate are aligned to ITRF2005 using 179 stations of high geodetic quality. Datum defined by a set of 3 dimensional Cartesian station coordinates and velocities.	
Frame reference epoch:	referenceEpoch:	2005.0	
Dynamic geodetic reference frame publication date:	publicationDate:	2010-05-31	

This defines the dynamic CRS. However this is not the complete definition of coordinate metadata required for a coordinate set referenced to a dynamic CRS. It is also necessary to give the coordinate epoch for which coordinates in the coordinate set are referenced. For example:

Station	Geocentric-X (m)	Geocentric-Y (m)	Geocentric-Z (m)
10001S006 Paris	4202777.214	171368.223	4778660.334
10002M006 Grasse	4581690.734	556115.067	4389360.944
10003M004 Toulouse	4627845.886	119629.575	4372999.970

Coordinate metadata:
    Coordinate reference system: ITRF2008
    Coordinate epoch: 2017.56

Example E.2.2: **Definition of a Geodetic CRS with static reference frame ('Static CRS').**

Attribute	UML identifier	Data Entry	Comment
	GeodeticCRS		
Geodetic CRS name:	name:	GDA2020 - XYZ	
CRS scope:	scope:	Spatial referencing	This attribute is optional but recommended.
CRS validity:	domainOfValidity:	Australia	This attribute is optional but recommended. This example shows a character string: refer to ISO 19115.
CRS remarks:	remarks:	Supersedes GDA94.	This attribute is optional.
	CartesianCS		A Cartesian CS may be 2- or 3-dimensional. The axes descriptions will be given 2 or 3 times, as appropriate. In this example, the system is 3-dimensional.
Cartesian coordinate system name:	name	ECEF right-handed	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	Geocentric X	
Coordinate system axis abbreviation:	axisAbbrev:	X	
Coordinate system axis direction:	axisDirection:	In the equatorial plane from the centre of the Earth towards the intersection of the equator with the prime meridian.	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	Geocentric Y	
Coordinate system axis abbreviation:	axisAbbrev:	Y	
Coordinate system axis direction:	axisDirection:	In the equatorial plane from the centre of the Earth towards the intersection of the equator and the meridian $\pi/2$ radians eastwards from the prime meridian.	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	Geocentric Z	
Coordinate system axis abbreviation:	axisAbbrev:	Z	
Coordinate system axis direction:	axisDirection:	From the centre of the Earth parallel to its rotation axis and towards its north pole.	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	GeodeticReferenceFrame		Because the datum type is GeodeticReferenceFrame the frame is not dynamic and therefore the CRS is static.
Geodetic reference frame name	name:	Geocentric Datum of Australia 2020	
Dynamic geodetic reference frame anchor definition:	anchorDefinition:	ITRF2014 at epoch 2020.0	This is an optional attribute. GDA2020 is coincident with ITRF2014 only at epoch 2020.0.
	Ellipsoid		For a geodetic CRS this attribute is optional but recommended.
Ellipsoid name:	name:	GRS 1980	
Length of semi-major axis:	semiMajorAxis:	6378137.0 m	
Inverse flattening:	inverseFlattening:	298.257222101	
Because the PrimeMeridian class is absent, the attributes name and Greenwich longitude take their default values of "Greenwich" and "0 degrees" respectively.			

Example E.2.3:   Definition of a Derived Geographic 3D CRS.

This geographic 3D CRS is derived from a geodetic CRS. This is only possible because the geodetic CRS description included an ellipsoid. This example uses the CRS from E.2.2 as the base CRS.

NOTE   This particular CRS could also be defined as a principal CRS: this is not true of all derived CRSs.

Attribute	**UML identifier**	**Data Entry**	**Comment**
	DerivedGeographicCRS		
Derived Geographic CRS name:	name:	GDA2020 - LatLonEht	
CRS scope:	scope:	Spatial referencing	
CRS validity:	domainOfValidity:	Australia	
	baseCRS		
Base CRS name:	name:	GDA2020 - XYZ	Example E.2.2
	GeodeticReferenceFrame		This is inherited from the base CRS.
Geodetic reference frame name:	name:	Geocentric Datum of Australia 2020	
	Ellipsoid		
Ellipsoid name:	name:	GRS 1980	
Length of semi-major axis:	semiMajorAxis:	6378137.0 m	
Inverse flattening:	inverseFlattening:	298.257222101	
	DerivingConversion		The source and target CRSs are implied because this is a component of the derivedCRS.
Conversion name:	name:	geocentric to geographic3D	
	OperationMethod		
Coordinate operation method name:	name:	Geocentric-Geographic conversions	
Coordinate operation method formula:	formula:	[A citation (CI_Citation) for the formula or the formula itself should be given here but is not detailed in this example]	The arguments used in this conversion method are the ellipsoid defining parameters. It is considered to be a parameterless conversion and no coordinate operation parameters are associated with the method..
	EllipsoidalCS		An ellipsoidal CS may be 2- or 3-dimensional. The axes descriptions will be given 2 or 3 times, as appropriate. In this example, the system is 3-dimensional.
Ellipsoidal coordinate system name:	name:	Ellipsoidal 3D CS. Axes: latitude, longitude, ellipsoidal height. Orientations: north, east, up. UoM: degree, degree, metre.	
	CoordinateSystemAxis		These are the attributes for the first axis, used by the first coordinate in a coordinate tuple.
Coordinate system axis name:	name:	geodetic latitude	
Coordinate system axis abbreviation:	axisAbbrev:	φ	
Coordinate system axis direction:	axisDirection:	north	
Coordinate system axis unit identifier:	axisUnitID:	degree	

Attribute	UML identifier	Data Entry	Comment
	CoordinateSystemAxis		These are the attributes for the second axis, used by the second coordinate in a coordinate tuple.
Coordinate system axis name:	name:	geodetic longitude	
Coordinate system axis abbreviation:	axisAbbrev:	λ	
Coordinate system axis direction:	axisDirection:	east	
Coordinate system axis unit identifier:	axisUnitID:	degree	
	CoordinateSystemAxis		These are the attributes for the third axis, used by the third coordinate in a coordinate tuple.
Coordinate system axis name:	name:	ellipsoidal height	
Coordinate system axis abbreviation:	axisAbbrev:	h	
Coordinate system axis direction:	axisDirection:	up	
Coordinate system axis unit identifier:	axisUnitID:	metre	

Example E.2.4:   **Definition of a Geographic 2D CRS**.

Attribute	UML identifier	Data Entry	Comment
	GeodeticCRS		
Geodetic CRS name:	name:	NAD83(CSRS) v6 - LatLon	
CRS scope:	scope:	Spatial referencing	

CRS validity:	domainOfValidity:	EX_GeographicBoundingBox
		    westBL: -120
		    eastBL: -57.1
		    southBL: 43.46
		    northBL: 62.56
			This attribute is optional but recommended. This example shows geographic bounding box entries: refer to ISO 19115-1.

	EllipsoidalCS		An ellipsoidal CS may be 2- or 3-dimensional. The axes descriptions will be given 2 or 3 times, as appropriate. In this example, although the CRS is 3-dimensional it is assumed that the coordinate tuple contains only latitude and longitude, and therefore, no description of a third, vertical CS axis is required.
Ellipsoidal coordinate system name:	name:	Latitude/longitude in degrees	Finding a suitable entry for the mandatory CS name is often a challenge as there is no established practice for naming coordinate systems.
	CoordinateSystemAxis		
Coordinate system axis name:	name:	geodetic latitude	
Coordinate system axis abbreviation:	axisAbbrev:	φ	
Coordinate system axis direction:	axisDirection:	north	
Coordinate system axis unit identifier:	axisUnitID:	degree	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	geodetic longitude	
Coordinate system axis abbreviation:	axisAbbrev:	λ	
Coordinate system axis direction:	axisDirection:	east	
Coordinate system axis unit identifier:	axisUnitID:	degree	
	GeodeticReferenceFrame		
Geodetic reference frame name:	name:	North American Datum 1983 (CSRS) version 6	
Geodetic reference frame remarks:	remarks:	Adopted by the Canadian federal government for Canada, and by provincial governments in Alberta, British Columbia, Manitoba, Newfoundland and Labrador, Nova Scotia, Ontario, Prince Edward Island. Replaces NAD83(CSRS) v5. Replaced by NAD83(CSRS) v7.	An optional entry.

Attribute	UML identifier	Data Entry	Comment
Geodetic reference frame anchor definition:	anchorDefinition:	Realization of the North American Datum of 1983 for the Canadian Spatial Reference System, referred to as CSRS98 or CSRS. The frame is defined by a time-dependent seven parameter transformation of ITRF2008 3D geocentric Cartesian coordinates and velocities for Canadian and bordering US and Greenland areas at reference epoch 2010.0. The frame is kept aligned to North America at other epochs using the NNR-NUVEL-1A estimate of three Cartesian rotation rates of change representing the tectonic plate motion of North America. The origin, scale and orientation of the frame are nominally defined to be that for the BIH Terrestrial System 1984 (BTS84).	An optional entry.
Geodetic reference frame publication date:	publicationDate:	2010-01-01	An optional entry.
	PrimeMeridian		Because the datum type is GeodeticReferenceFrame, if this PrimeMeridian class had been absent, the attributes 'prime meridian name' and 'Greenwich longitude' would have taken their default values.
Prime meridian name	name:	Greenwich	Because the value for this attribute is "Greenwich", it is not essential to provide this attribute information.
Prime meridian Greenwich longitude	GreenwichLongitude:	0 degrees	Because the value for the prime meridian name is "Greenwich", it is not essential to provide the prime meridian Greenwich longitude information.
	Ellipsoid		
Ellipsoid name:	name:	GRS 1980	
Length of semi-major axis:	semiMajorAxis:	6378137.0 m	
Inverse flattening:	inverseFlattening:	298.2572221	

Example E.2.5:   Definition of a Geographic CRS with datum ensemble and multiple usage entries.

Attribute	UML identifier	Data Entry	Comment
	GeographicCRS		
CRS name:	name:	WGS 84 ensemble	
CRS alias:	alias:	WGS 84	
CRS usage:			The scope and domain of validity pairing may be repeated.
CRS scope:	scope:	GIS.	

CRS validity:	domainOfValidity:	EX_GeographicBoundingBox
		    westBL: -180
		    eastBL: 180
		    southBL: -90
		    northBL: 90
			This example shows geographic bounding box entries: refer to ISO 19115-1.

Attribute	UML identifier	Data Entry	Comment
CRS usage:			
CRS scope:	scope:	Low and medium accuracy applications.	
CRS validity:	domainOfValidity:	World.	This example shows geographic description entry: refer to ISO 19115-1.
CRS remarks:	remarks:	"WGS 84" is used to mean either the original (Transit) realization or this datum ensemble. For high accuracy applications use one of the specific realizations.	This attribute is optional.
	EllipsoidalCS		
Ellipsoidal coordinate system name	name	Latitude/longitude in degrees	Finding a suitable entry for the mandatory CS name is often a challenge as there is no established practice for naming coordinate systems.
	CoordinateSystemAxis		
Coordinate system axis name:	name:	geodetic latitude	
Coordinate system axis abbreviation:	axisAbbrev:	φ	
Coordinate system axis direction:	axisDirection:	north	
Coordinate system axis unit identifier:	axisUnitID:	degree	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	geodetic longitude	
Coordinate system axis abbreviation:	axisAbbrev:	λ	
Coordinate system axis direction:	axisDirection:	east	
Coordinate system axis unit identifier:	axisUnitID:	degree	
	DatumEnsemble		Datum ensemble example uses abbreviation for each component as the component identifier.
Datum ensemble name	name:	WGS 84 ensemble	
Conventional reference system:	conventionalRS:	WGS 84	
Geodetic reference frame ID:	datumID:	WGS 84 (Transit)	
Geodetic reference frame ID:	datumID:	WGS 84 (G730)	
Geodetic reference frame ID:	datumID:	WGS 84 (G873)	
Geodetic reference frame ID:	datumID:	WGS 84 (G1150)	
Geodetic reference frame ID:	datumID:	WGS 84 (G1674)	
Geodetic reference frame ID:	datumID:	WGS 84 (G1762)	
	Ellipsoid		Note: this ellipsoid information is repeated through every reference frame ID. As all members of the datum ensemble should be using the same ellipsoid, for reporting the information only needs to be given once. Here it is shown after the last datum entry.
Ellipsoid name:	name:	WGS 1984	
Length of semi-major axis:	semiMajorAxis:	6378137.0 m	
Inverse flattening:	inverseFlattening:	298.257223563	
Ensemble accuracy:	ensembleAccuracy:	1 m	
DatumEnsemble remarks:	remarks:	Realizations differ by 0.7m between the Transit and G730 realizations, a further 0.2m between G730 and G873, 0.06m between G873 and G1150, 0.2m between G1150 and G1674 and 0.02m between G1674 and G1762.	An optional attribute.

Example E.2.6: Definition of a Projected 2D CRS.

This example shows the full definition of the CRS identified in the example in E.1.

Attribute	UML identifier	Data Entry	Comment
	ProjectedCRS		
Projected CRS name:	name:	NAD27 / Alaska zone 4	
CRS scope:	scope:	Topographic mapping.	This attribute is optional but recommended.
CRS validity:	domainOfValidity:	Alaska between 148 and 152 degrees west	This attribute is optional but recommended.
	baseCRS		
Base CRS name	name:	NAD27	
	GeodeticReferenceFrame		This is inherited from the base CRS. Because the datum type is GeodeticReferenceFrame, and because the PrimeMeridian class is absent, the attributes prime meridian name and Greenwich longitude take their default values of "Greenwich" and "0 degrees" respectively.
Geodetic reference frame name	name:	North American Datum of 1927	
Geodetic reference frame alias	alias:	NAD27	This is an optional attribute.
	Ellipsoid		This example uses the semi-minor axis as the second defining parameter.
Ellipsoid name:	Name:	Clarke 1866	
Length of semi-major axis:	semiMajorAxis:	6378206.4 m	
Length of semi-minor axis:	semiMinorAxis:	6356583.8 m	
Ellipsoid remarks:	remarks:	Inverse flattening derived from semi-major and semi-minor axes is 294.9786982. Semi-major axis in US survey feet = 20925832.164 ftUS.	Remarks is an optional attribute.
	CartesianCS		A Cartesian CS may be 2- or 3-dimensional. The axes descriptions will be given 2 or 3 times, as appropriate. In this example, the system is 2-dimensional.
Cartesian coordinate system name:	name	State Plane Coordinate System (ftUS)	
Cartesian coordinate system remarks:	remarks:	1 US survey foot = 12/39.37m	
	CoordinateSystemAxis		These are the attributes for the first axis, used by the first coordinate in a coordinate tuple.
Coordinate system axis name:	name:	easting	
Coordinate system axis abbreviation:	axisAbbrev:	X	
Coordinate system axis direction:	axisDirection:	east	
Coordinate system axis unit identifier:	axisUnitID:	US survey foot	
	CoordinateSystemAxis		These are the attributes for the second axis, used by the second coordinate in a coordinate tuple.
Coordinate system axis name:	name:	northing	

Attribute	UML identifier	Data Entry	Comment
Coordinate system axis abbreviation:	axisAbbrev:	Y	
Coordinate system axis direction:	axisDirection:	north	
Coordinate system axis unit identifier:	axisUnitID:	US survey foot	
	DerivingConversion		
Coordinate operation name:	name:	Alaska SPCS27 zone 4	
	OperationMethod		
Coordinate operation method name:	name:	Transverse Mercator ellipsoidal formula	
Coordinate operation method formula citation:	formula citation:	John P. Snyder[14] Map Projections – A Working Manual US Geological Survey Professional Paper 1395.	CI_Citation is described in ISO 19115.
	OperationParameter		The number of parameters (n) is dictated by the formula of the operation method. Parameter names, values (and, if required, optional attributes) will be given n times, as appropriate.
Operation parameter name:	name:	latitude of origin	
	ParameterValue:		
Operation parameter numeric value:	value:	54 degrees	
	OperationParameter		
Operation parameter name:	name:	longitude of origin	
	ParameterValue:		
Operation parameter numeric value:	value:	−150 degrees	
	OperationParameter		
Operation parameter name:	name:	scale factor	
	ParameterValue:		
Operation parameter numeric value:	value:	0.9999	This is a ratio and is unitless.
	OperationParameter		
Operation parameter name:	name:	false easting	
	ParameterValue:		
Operation parameter numeric value:	value:	500000 US survey foot	
	OperationParameter		
Operation parameter name:	name:	false northing	
	ParameterValue:		
Operation parameter numeric value:	value:	0 US survey foot	

Example E.2.7:   **Definition of a Projected 3D CRS**.

Attribute	UML identifier	Data Entry	Comment
	ProjectedCRS		
Projected CRS name:	name:	WGS 84 (G1762) / UTM zone 31N 3D	
CRS scope:	scope:	3D image georeferencing.	
CRS validity:	domainOfValidity:	Between 0°E and 6°E, northern hemisphere between equator and 84°N, onshore and offshore.	
	baseCRS		
Base CRS name:	name:	WGS 84 (G1762) – LatLonEht	
	GeodeticReferenceFrame		The reference frame is inherited from the base CRS.
Geodetic reference frame name:	name:	World Geodetic System of 1984 (G1762)	
	Ellipsoid		
Ellipsoid name:	name:	WGS 1984	
Length of semi-major axis:	semiMajorAxis:	6378137.0 m	
Inverse flattening:	inverse flatenning:	298.257223563	
	CartesianCS		
Cartesian coordinate system name:	name:	Cartesian 3D CS. Axes: easting, northing, ellipsoidal height (E,N,h). Orientations east, north, up. UoM m.	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	easting	
Coordinate system axis abbreviation:	axisAbbrev:	E	
Coordinate system axis direction:	axisDirection:	east	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	northing	
Coordinate system axis abbreviation:	axisAbbrev:	N	
Coordinate system axis direction:	axisDirection:	north	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	ellipsoidal height	
Coordinate system axis abbreviation:	axisAbbrev:	h	
Coordinate system axis direction:	axisDirection:	up	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	DerivingConversion		
Coordinate operation name:	name:	UTM zone 31N	
Coordinate operation scope:	scope:	Topographic mapping	
Coordinate operation validity:	domainOfValidity:	Between 0°E and 6°E, northern hemisphere between equator and 84°N, onshore and offshore.	.
	OperationMethod		
Coordinate operation method name:	name:	Transverse Mercator 3D	

Attribute	UML identifier	Data Entry	Comment
Coordinate operation method remarks:	remarks:	This map projection method is two-dimensional. The ellipsoidal height of the base CRS is passed through as the Cartesian CS vertical axis.	
Coordinate operation method formula:	formula:	(CI_Citation)	[Citation (CI_Citation) for the formula or the formula itself should be given here but is not detailed in this example]
	OperationParameter		The number of parameters (n) is dictated by the formula of the operation method. Parameter names, values (and, if required, optional attributes) will be given n times, as appropriate.
Operation parameter name:	name:	latitude of origin	
	ParameterValue		
Operation parameter numeric value:	value:	0 degrees	
	OperationParameter		
Operation parameter name:	name:	longitude of origin	
	ParameterValue		
Operation parameter numeric value:	value:	3 degrees	
	OperationParameter		
Operation parameter name:	name:	scale factor	
	ParameterValue		
Operation parameter numeric value:	value:	0.9996	This is a ratio and is unitless.
	OperationParameter		
Operation parameter name:	name:	false easting	
	ParameterValue		
Operation parameter numeric value:	value:	500000 metre	
	OperationParameter		
Operation parameter name:	name:	false northing	
	ParameterValue		
Operation parameter numeric value:	value:	0 metre	

Example E.2.8: Definition of a Derived Projected CRS.

Attribute	UML identifier	Data Entry	Comment
	DerivedProjectedCRS		
Derived Projected CRS name:	name:	NAD27 / Gulf of Mexico speculative seismic survey bin grid	
CRS scope:	scope:	Geophysical exploration.	This attribute is optional but recommended.
CRS validity:	domainOfValidity:	US - Gulf of Mexico	This attribute is optional but recommended.
	baseCRS		
Base CRS name:	name:	NAD27 / Texas South Central	
	GeodeticReferenceFrame		This is inherited from the base CRS. Because the datum type is GeodeticReferenceFrame, and because the PrimeMeridian class is absent, the attributes prime meridian name and Greenwich longitude take their default values of "Greenwich" and "0 degrees" respectively.
Geodetic reference frame name:	name	North American Datum of 1927	
Datum alias:	alias	NAD27	This is an optional attribute.
	Ellipsoid		
Ellipsoid name:	name:	Clarke 1866	
Length of semi-major axis:	semiMajorAxis:	6378206.4 m	
Length of semi-minor axis:	semiMinorAxis:	6356583.8 m	
Ellipsoid remarks:	remarks:	Inverse flattening derived from semi-major and semi-minor axes is 294.9786982	Remarks is an optional attribute.
	DerivingConversion		Definition of the map projection of the base projected CRS.
Coordinate operation name:	name:	Texas South Central SPCS27	
	OperationMethod		
Coordinate operation method name:	name:	Lambert Conic Conformal (2SP) ellipsoidal formula	
Coordinate operation method formula citation:	formula citation:	John P. Snyder Map Projections – A Working Manual US Geological Survey Professional Paper 1395.	CI_Citation is described in ISO 19115.
	OperationParameter		The number of parameters (n) is dictated by the formula of the operation method. Parameter names, values (and, if required, optional attributes) will be given n times, as appropriate.
Operation parameter name:	name:	latitude of origin	
	ParameterValue		
Operation parameter numeric value:	value:	27.83333333333 degrees	27°50'N
	OperationParameter		
Operation parameter name:	name:	longitude of origin	
	ParameterValue		
Operation parameter numeric value:	value:	−99 degrees	

Attribute	UML identifier	Data Entry	Comment
	OperationParameter		
Operation parameter name:	name:	Latitude of 1st standard parallel	
	ParameterValue		
Operation parameter numeric value:	value:	28.383333333333 degrees	28°23'N
	OperationParameter		
Operation parameter name:	name:	Latitude of 2nd standard parallel	
	ParameterValue		
Operation parameter numeric value:	value:	30.283333333333 degrees	30°17'N
	OperationParameter		
Operation parameter name:	name:	false easting	
	ParameterValue		
Operation parameter numeric value:	value:	2000000 US survey foot	
	OperationParameter		
Operation parameter name:	name:	false northing	
	ParameterValue		
Operation parameter numeric value:	value:	0 US survey foot	
	DerivingConversion		Definition of the deriving conversion for the derived projected CRS
Coordinate operation name:	name:	Gulf of Mexico speculative survey bin grid	
	OperationMethod		
Coordinate operation method name	name:	IOGP P6 (I = J-90°) seismic bin grid transformation	
Coordinate operation method formula citation	formula citation:	EPSG Guidance note 7-2 "Coordinate Conversions and Transformations including Formulas" IOGP Geomatics Publication 373-7-2.	CI_Citation is described in ISO 19115.
	OperationParameter		The number of parameters (n) is dictated by the formula of the operation method. Parameter names, values (and, if required, optional attributes) will be given n times, as appropriate.
Operation parameter name:	name:	Bin grid origin I	
	ParameterValue		
Operation parameter numeric value:	value:	5000 I-bin	
	OperationParameter		
Operation parameter name:	name:	Bin grid origin J	
	ParameterValue		
Operation parameter numeric value:	value:	0 J-bin	
	OperationParameter		
Operation parameter name:	name:	Bin grid origin easting	
	ParameterValue		
Operation parameter numeric value:	value:	871200 ftUS	
	OperationParameter		

Attribute	UML identifier	Data Entry	Comment
Operation parameter name:	name:	Bin grid origin Northing	
	ParameterValue		
Operation parameter numeric value:	value:	10280160 ftUS	
	OperationParameter		
Operation parameter name:	name:	Scale factor of bin grid	
	ParameterValue		
Operation parameter numeric value:	value:	1.0	
	OperationParameter		
Operation parameter name:	name:	Bin width on I-axis	
	ParameterValue		
Operation parameter numeric value:	value:	82.5 ftUS	
	OperationParameter		
Operation parameter name:	name:	Bin width on J-axis	
	ParameterValue		
Operation parameter numeric value:	value:	41.25 ftUS	
	OperationParameter		
Operation parameter name:	name:	Map grid bearing of bin grid J-axis	
	ParameterValue		
Operation parameter numeric value:	value:	340°	
	OperationParameter		
Operation parameter name:	name:	Bin node increment on I-axis	
	ParameterValue		
Operation parameter numeric value:	value:	1.0	
	OperationParameter		
Operation parameter name:	name:	Bin node increment on J-axis	
	ParameterValue		
Operation parameter numeric value:	value:	1.0	
	OrdinalCS		For each axis in an ordinal CS the attribute axisUnitID and its UnitOfMeasure is not given.
Ordinal coordinate system name:	name:	Gulf of Mexico speculative seismic survey bin grid	
	CoordinateSystemAxis		These are the attributes for the first axis, used by the first coordinate in a coordinate tuple.
Coordinate system axis name:	name:	Bin grid I	
Coordinate system axis abbreviation:	axisAbbrev:	I	
Coordinate system axis direction:	axisDirection:	northNorthWest	
	CoordinateSystemAxis		These are the attributes for the second axis, used by the second coordinate in a coordinate tuple.
Coordinate system axis name:	name:	Bin grid J	
Coordinate system axis abbreviation:	axisAbbrev:	J	
Coordinate system axis direction:	axisDirection:	westSouthWest	

Example E.2.9: **Definition of a Vertical CRS**.

Attribute	UML identifier	Data Entry	Comment
	VerticalCRS		
Vertical CRS name:	name:	NGVD29 height	
CRS scope:	scope:	National height system	
CRS validity:	domainOfValidity:	Conterminous US (lower 48 states)	
CRS remarks:	remarks:	Superseded by NAVD88	
	VerticalCS		
Vertical coordinate system name:	name:	Gravity-related height	
Vertical coordinate system remarks:	remarks:	1 US survey foot = 12/39.37 meter.	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	height	
Coordinate system axis abbreviation:	axisAbbrev:	H	
Coordinate system axis direction:	axisDirection:	up	
Coordinate system axis unit identifier:	axisUnitID:	US survey foot	
	VerticalReferenceFrame		
Vertical reference frame name:	name:	National Geodetic Vertical Datum of 1929	
Vertical reference frame alias:	alias:	NGVD29	This is an optional attribute.
Vertical reference frame anchor definition:	anchorDefinition:	26 tide gauges in the US and Canada	This is an optional attribute.
Vertical reference frame realization method:	realization:	levelling	This is an optional attribute, shown here only as an example.

Example E.2.10:   Definition of a geoid-based Vertical CRS.

A geoid-based vertical CRS is realized through the application of a geoid height model to a geodetic CRS. However it is not a derived CRS because the vertical CRS does not inherit the geodetic reference frame of the geodetic CRS but has its own vertical reference frame. The geoid height model is described as a coordinate transformation. The same geoid-based vertical CRS may also be realized from a different geodetic CRS (in this case ITRF2008) through application of a second geoid height model. Each of the geoid height models is applied to a specified CRS. The geoid height model and vertical CRS are related to each other through the *HeightTransfomation* association. See example E.5.2 for the description of a geoid height model.

Attribute	UML identifier	Data Entry	Comment
	VerticalCRS		
Vertical CRS name:	name:	CGVD2013 - OHt	
CRS scope:	scope:	Spatial referencing	
CRS validity:	domainOfValidity:	Canada - onshore and offshore: Alberta; British Columbia; Manitoba; New Brunswick; Newfoundland and Labrador; Northwest Territories; Nova Scotia; Nunavut; Ontario; Prince Edward Island; Quebec; Saskatchewan; Yukon.	
CRS remarks:	remarks:	Information source: M. Veronneau and J. Huang. The Canadian Geodetic Vertical Datum of 2013 (CGVD2013). Geomatica, Vol. 70, No. 1, 2016, pp. 9-19. Replaces CGVD28.	
	VerticalCS		
Vertical coordinate system name:	name:	Gravity-related height	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	height	
Coordinate system axis abbreviation:	axisAbbrev:	H	
Coordinate system axis direction:	axisDirection:	up	
Coordinate system axis unit identifier:	axisUnitID:	metre	

Attribute	UML identifier	Data Entry	Comment
	VerticalReferenceFrame		
Vertical reference frame name:	name:	Canadian Geodetic Vertical Datum of 2013	
Vertical reference frame alias:	alias:	CGVD2013	This is an optional attribute.
Vertical reference frame anchor definition:	anchorDefinition:	CGVD2013 is a gravimetric datum defined by the Canadian Gravimetric Geoid of 2013 (CGG2013), referenced to the NAD83(CSRS) v6 geodetic datum. The geoid-based datum is defined by the equipotential surface Wo=62,636,856.0 m^2s^-2, representing by convention the coastal mean sea level for North America. The definition and geopotential value comes from an agreement between Canada and the USA. The Canadian Gravimetric Geoid of 2013 (CGG2013) is the first realization of the vertical datum. CGG2013 was defined at epoch 2011.0 and is considered static. It is available in both the NAD83(CSRS) and ITRF2008 geometric reference frames using the GRS80 ellipsoid, making it compatible with space-based positioning techniques. Heights in CGVD2013 are orthometric and can be obtained from NAD83(CSRS) v6 or ITRF2008 ellipsoidal heights by subtracting the CGG2013 geoid height in either NAD83(CSRS) v6 or ITRF2008, respectively.	This is an optional attribute.
Vertical reference frame publication date:	publicationDate:	2013-11-28	
Vertical reference frame realization method:	realization:	geoid	
Geoid model:	heightTransformation:	CGG2013	This is a reference to the geoid model through which this geoid-based vertical CRS is defined. It is described in example E.5.2 through which it may be established that the source geodetic CRS is NAD83(CSRS) v6.

Example E.2.11:   Definition of a Compound CRS

This example demonstrates a spatial compound CRS formed from a projected CRS with a vertical CRS.

Attribute	UML identifier	Data Entry	Comment
	CompoundCRS		This example supports a coordinate tuple of easting, northing, gravity-related height.
Compound CRS name:	name:	British National Grid + ODN	
Compound CRS scope:	scope:	National mapping including heighting related to mean sea level.	
Compound CRS validity	domainOfValidity:	Great Britain mainland.	
			The individual CRSs forming the compound CRS are next described. The sequence is significant as it implies the order of coordinates in the coordinate tuple.
	ProjectedCRS		The projected 2D CRS is then described in a similar manner to that in Example E.2.6.
Projected CRS name	name:	British National Grid	
Projected CRS scope	scope:	Large, medium and small scale topographic mapping, engineering survey and GIS.	
Projected CRS validity	domainOfValidity:	England, Wales, Scotland, Isle of Man.	
	CartesianCS		
Cartesian coordinate system name	name:	National grid	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	easting	
Coordinate system axis abbreviation:	axisAbbrev:	E	
Coordinate system axis direction:	axisDirection:	east	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	northing	
Coordinate system axis abbreviation:	axisAbbrev:	N	
Coordinate system axis direction:	axisDirection:	north	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	GeodeticReferenceFrame		
Geodetic reference frame name	name:	Ordnance Survey of Great Britain 1936	
	Ellipsoid		
Ellipsoid name	name:	Airy 1830	
Length of semi-major axis	semiMajorAxis:	6377563.396 m	
Inverse flattening	inverseFlattening:	299.3249646	
	Conversion		
Coordinate operation name:	name:	British National Grid	
Coordinate operation scope:	scope:	Topographic mapping	
	OperationMethod		
Coordinate operation method name	name:	Transverse Mercator	

Attribute	UML identifier	Data Entry	Comment
Coordinate operation method formula	formula:	[Citation (CI_Citation) describing the formula or the formula itself should be given here and is not detailed in this example].	
	OperationParameter		
Operation parameter name:	name:	latitude of origin	
	ParameterValue		
Operation parameter numeric value:	value:	49 degrees	
	OperationParameter		
Operation parameter name:	name:	longitude of origin	
	ParameterValue		
Operation parameter numeric value:	value:	−2 degrees	
	OperationParameter		
Operation parameter name:	name:	scale factor	
	ParameterValue		
Operation parameter numeric value:	value:	0.9996012717	
	OperationParameter		
Operation parameter name:	name:	false easting	
	ParameterValue		
Operation parameter numeric value:	value:	400000 m	
	OperationParameter		
Operation parameter name:	name:	false northing	
	ParameterValue		
Operation parameter numeric value:	value:	−100000 m	
	VerticalCRS		The vertical CRS is then described in a similar manner to that in Example E.2.9 or E.2.10.
Vertical CRS name:	name:	ODN	
Vertical CRS validity:	domainOfValidity:	British mainland	
Vertical CRS scope:	scope:	National height system	
	VerticalCS		
Vertical coordinate system name:	name:	ODN heights	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	height	
Coordinate system axis abbreviation:	axisAbbrev:	H	
Coordinate system axis direction:	axisDirection:	up	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	VerticalReferenceFrame		
Vertical reference frame name:	name:	Ordnance Datum Newlyn	
			The order of coordinates in a coordinate tuple referenced to the compound CRS is then implied as E,N,H.

Example E.2.12   Definition of an Engineering CRS applied to a construction site.

Attribute	UML identifier	Data Entry	Comment
	EngineeringCRS		
Engineering CRS name:	name:	Best building CRS	
CRS scope:	scope:	Building construction and maintenance.	
CRS validity:	domainOfValidity:	The Best building, Gondwanaland.	
	CartesianCS		Because there are three instances of CS axis, this is a 3D system.
Cartesian coordinate system name:	name:	Right-handed 3D CS	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	site east	
Coordinate system axis abbreviation:	axisAbbrev:	E	
Coordinate system axis direction:	axisDirection:	northeast	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	site north	
Coordinate system axis abbreviation:	axisAbbrev:	N	
Coordinate system axis direction:	axisDirection:	northwest	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	height	
Coordinate system axis abbreviation:	axisAbbrev:	H	
Coordinate system axis direction:	axisDirection:	up	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	Engineering Datum		
Engineering datum name:	name:	Best building site	
Engineering datum anchor definition:	anchorDefinition:	Peg A in the southwest corner of the site. Direction to peg B in the northwest corner of the site defined as grid north. Horizontal coordinates of peg A are (0,0). Horizontal coordinates of peg B are (0, 273.46). Height of peg A assigned the value of +50.0m.	

Copyright © 2021 Open Geospatial Consortium

Example E.2.13   Definition of an Engineering CRS applied to a moving object.

Attribute	UML identifier	Data Entry	Comment
	EngineeringCRS		
Engineering CRS name:	name:	Cessna K-1234 CRS	
CRS scope:	scope:	Aerial survey project ABC-123.	
CRS validity:	domainOfValidity:	Cessna K-1234	
CRS remarks:	remarks:	Project ABC-123.	
	CartesianCS		Because there are three instances of CS axis, this is a 3D system.
Cartesian coordinate system name:	name:	Right-handed 3D CS	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	forward	
Coordinate system axis abbreviation:	axisAbbrev:	X	
Coordinate system axis direction:	axisDirection:	forward	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	starboard	
Coordinate system axis abbreviation:	axisAbbrev:	Y	
Coordinate system axis direction:	axisDirection:	starboard	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	down	
Coordinate system axis abbreviation:	axisAbbrev:	Z	
Coordinate system axis direction:	axisDirection:	down	
Coordinate system axis unit identifier:	axisUnitID:	metre	
	Engineering Datum		
Engineering datum name:	name:	Cessna K-1234	
Engineering datum anchor definition:	anchorDefinition:	Aircraft centre of gravity / inertial reference system reference point.	

Example E.2.14   Definition of an Engineering CRS applied to an image.

Attribute	UML identifier	Data Entry	Comment
	EngineeringCRS		
Engineering CRS name:	name:	Photo 123456	
CRS scope:	scope:	Building construction and maintenance.	
CRS validity:	domainOfValidity:	The Best building, Gondwanaland.	
	OrdinalCS		Because there are two instances of CS axis, this is a 2D system.
Cartesian coordinate system name:	name:		
	CoordinateSystemAxis		
Coordinate system axis name:	name:	row	
Coordinate system axis abbreviation:	axisAbbrev:	R	
Coordinate system axis direction:	axisDirection:	rowPositive	
Coordinate system axis unit identifier:	axisUnitID:	micrometre	
	CoordinateSystemAxis		
Coordinate system axis name:	name:	column	
Coordinate system axis abbreviation:	axisAbbrev:	C	
Coordinate system axis direction:	axisDirection:	columnPositive	
Coordinate system axis unit identifier:	axisUnitID:	micrometre	
	Engineering Datum		
Engineering datum name:	name:	Photo 123456	
Engineering datum anchor definition:	anchorDefinition:	Top left corner	

E.3   Definition of parametric coordinate reference systems

Example E.3.1:   Definition of a Parametric CRS using a parameter (pressure)

Barometric pressure is the basic measure of height used in aviation and meteorology, but the exact translation to a height depends on current conditions in the local atmospheric profile. In 1951, the International Civil Aviation Organisation (ICAO) incorporated the international standard atmosphere (ISA) into international law[1]. There have been a number of extensions since, up to 80 km. With the publication of ISO 2533 in 1975, a standard atmosphere in the range 2 000 m to 5 000 m was established[2]. See References [15] and [16].

Height in the atmosphere is measured by barometric pressure which is monotonically decreasing in height. Although the ISA is calibrated in both thousands of feet and in metres, it does not measure true height, but approximate geopotential height. This is because the datum ignores the variation of the atmospheric temperature and pressure near the bottom of the atmosphere. Heights are named as flight levels (e.g. FL320 is nominally 32 000 ft). Even if a true height measure is available in an aircraft, for example, through radar or GPS (Global Positioning System), the readings must be converted to ISA flight levels — unless the pilot is flying under visual flying rules (VFR) near the ground.

The datum is set at mean sea level pressure in the standard atmosphere at 1 013.25 hectopascals (hPA) [also expressed in the non-SI unit of millibars (mb)][3].

NOTE   When aircraft fly at low level over topography, air traffic control (ATC) regulations set a transitional flight level or altitude below which the ISA does not apply, but at which the reference atmosphere does. This involves the pilot resetting the datum to ensure the aircraft is above all topography. The new datum (known as QNH) is transmitted by radio from ATC, and is the lowest pressure (reduced to mean sea level) forecast for the next 3 h in the low-flying zone, or the current aerodrome pressure (QFE) if the aircraft is about to land.

---

[1]   Convention on International Civil Aviation (the Chicago Convention 1947), Annex 8.

[2]   The US, ICAO and WMO (World Meteorological Organization) standard atmospheres are the same as the ISO standard atmosphere for altitudes up to 32 km.

[3]   For aviation in North America, by practice and by law, the datum is expressed as 29.92 in and hundredths of mercury.

Attribute	UML identifier	Entry	Comment
	ParametricCRS		
Parametric CRS name:	name:	ICAO international standard atmosphere (ISA)	
CRS alias:	alias:	WMO standard atmosphere	This is an optional attribute.
CRS scope:	scope:	Aviation, meteorology	
CRS validity:	domainOfValidity	2 km to 80 km in the free atmosphere (above topography)	
CRS remarks:	remarks:	From 2 km to 32 km, equivalent to ISO 2533:1975	This is an optional attribute.
	ParametricCS		
Parametric coordinate system name:	name:	Aviation flight levels	
Parametric CS remarks:	remarks:	Flight level FL320 is 32000 ft (as geopotential height)	This is an optional attribute.
	CoordinateSystemAxis		
Coordinate system axis name:	name:	Flight level	
Coordinate system axis abbreviation:	axisAbbrev:	FL	
Coordinate system axis direction:	axisDirection:	Up	
Coordinate system axis unit identifier:	axisUnitID:	Geopotential metres	
Coordinate system axis minimum value:	minimumValue:	2000	This is an optional attribute.
Coordinate system axis maximum value:	maximumValue:	80000	This is an optional attribute.
Coordinate system axis range meaning:	rangeMeaning:	Exact	This is a conditional attribute.
CS axis remarks	remarks:	Used only above legal transitional altitude in regions with topography	This is an optional attribute.
	ParametricDatum		
Parametric datum name:	name:	Standard atmospheric pressure	
Parametric datum alias:	alias:	Mean sea level pressure (MSLP)	This is an optional attribute.
Parametric datum scope:	scope:	Aviation, meteorology	
Parametric datum anchor definition:	anchorDefinition:	Mean sea level	This is an optional attribute.
Parametric datum remarks:	remarks:	1 013.25 hPa	This is an optional attribute.

Example E.3.2:   Definition of a Parametric CRS using a function (potential vorticity)

Potential vorticity (PV) is a function which varies strongly with height. One common application of PV levels is to show values of fields on a single level of 2 PV units (1 PV unit = 10^{-6} m^{-2} s^{-1} K kg^{-1}), as this is a PV value often taken to denote the mid-latitude tropopause.

The PV is the absolute circulation of an air parcel that is enclosed between two isentropic surfaces. In the following equation, the PV is the product of absolute vorticity on an isentropic surface and the static stability. Thus, PV consists of two factors: a dynamical element and a thermodynamical element.

$$\text{PV} = -g\left(\zeta_{\Theta} + f\right)\frac{\partial \Theta}{\partial p}$$

where

f is the Coriolis parameter;

g is the gravitational acceleration;

p is the pressure;

Θ is the potential temperature;

ζ_{Θ} is the relative isentropic vorticity.

Within the troposphere, the values of PV are usually low. However, the potential vorticity increases rapidly from the troposphere to the stratosphere due to the significant change in the static stability. Typical changes of the potential vorticity within the area of the tropopause are from 1 (tropospheric air) to 4 (stratospheric air) PV units. Typically, the 2 PV unit anomaly, which separates tropospheric from stratospheric air, is referred to as *dynamical tropopause*. The traditional way of describing the tropopause is to use the potential temperature or static stability. This is only a thermodynamical way of characterizing the tropopause. The benefit of using PV is that the tropopause can be understood in both thermodynamic and dynamic terms. An abrupt folding or lowering of the dynamical tropopause can also be called an upper PV anomaly. When this occurs, stratospheric air penetrates into the troposphere, resulting in high values of PV with respect to the surroundings, creating a positive PV anomaly. In the lower levels of the troposphere, strong baroclinic zones often occur which can be regarded as low level PV anomalies. Due to the conservation of PV, significant features that are related to synoptic-scale weather systems can be identified and followed in space as well as in time.

Attribute	UML identifier	Entry	Comment
	ParametricCRS		
Parametric CRS name:	name:	Potential vorticity functional CRS	
CRS scope:	scope:	Meteorology	
CRS validity:	domainOfValidity:	The whole atmosphere	
	ParametricCS		
Parametric coordinate system name:	name:	Potential vorticity functional CS	
	CoordinateSystemAxis		
Parametric system axis name:	name:	Potential vorticity	
Coordinate system axis abbreviation:	axisAbbrev	PV	
Coordinate system axis direction:	axisDirection:	Upward	
Coordinate system axis unit identifier:	axisUnitID:	PVU	

Attribute	UML identifier	Entry	Comment
CS axis remarks:	remarks:	The potential vorticity unit is scaled to give values in the order of $(10^{-6}\,\text{K}\,\text{kg}^{-1}\,\text{m}^{-2}\,\text{s}^{-1})$	This is an optional attribute.
	ParametricDatum		
Parametric datum name:	name:	Zero of the computed PV function	

Example E.3.3: Definition of a spatio-parametric CRS

Presented here is the construction of a spatio-parametric coordinate reference system using Example E.1.6, for the horizontal component and an oceanographic example where the parameter is density for the vertical component.

The Miami isopycnal coordinate model (MICOM)[17] is an oceanographic numerical integration model which has horizontal latitude/longitude coordinates and a vertical coordinate which uses a temporal form based on potential density. One model version has MICOM configured for the Atlantic Ocean at 1/12th degree resolution providing fields of temperature and salinity for the MICOM domain for periods within a 20 year MICOM integration.

The MICOM grid in the deep oceans is in steps of potential density (density corrected for compressibility effects), rather than depth. The density of water varies with salinity and temperature as well as depth, and the isopycnal surfaces (constant potential density) are not level under the actions of wind and currents. Numerical ocean or weather forecasting models require complex grids in the vertical (and often the horizontal) to properly represent the physical processes involved. Using natural physical scales helps interpretation and most importantly keeps the model numerically stable. Computing the grid on density coordinates greatly reduces the numerically induced diabatic dispersion of water mass properties and preserves conservation laws, particularly on long model runs.

Different oceanographic models can have grids which differ greatly in detail. Many have hybrid coordinates which can be specified according to location. For example, the grid can be modified at the ocean bottom, in shallow seas and in unstratified water to allow a better representation of the specific physical processes involved there. For this example, all such complexity is ignored.

When the sea surface is used as a datum, the ocean model is subject to diurnal heating. For some ocean models, the datum is taken as 10 m to remove these fast variations; otherwise, a relevant mean sea level is used.

Attribute	UML identifier	Entry	Comment
	CompoundCRS		
Compound CRS name:	name:	WGS 84(G1762) + MICOM grid	
Compound CRS scope:	scope:	Oceanography	
Compound CRS validity:	domainOfValidity:	Surface to ocean bottom, worldwide	
	GeodeticCRS		The individual CRS forming the compound CRS are next described. The sequence is significant, implying the order in which coordinates are given. In this example it is latitude, longitude, potential density.
Geodetic CRS name:	name:	WGS 84(G1762)	

Attribute	UML identifier	Entry	Comment
Geodetic CRS scope:	scope:	Navigation	
Geodetic CRS validity:	domainOfValidity:	World	
EllipsoidalCS			
Ellipsoidal coordinate system name:	name:	Latitude/longitude in degrees	
CoordinateSystemAxis			
Coordinate system axis name:	name:	Geodetic latitude	
Coordinate system axis abbreviation:	axisAbbrev:	φ	
Coordinate system axis direction:	axisDirection:	North	
Coordinate system axis unit identifier:	axisUnitID:	Degree	
CoordinateSystemAxis			
Coordinate system axis name:	name:	Geodetic longitude	
Coordinate system axis abbreviation:	axisAbbrev:	λ	
Coordinate system axis direction:	axisDirection:	East	
Coordinate system axis unit identifier:	axisUnitID:	Degree	
GeodeticReferenceFrame			
Geodetic reference frame name:	name:	World Geodetic System of 1984(G1762)	
Ellipsoid			
Ellipsoid name:	name:	WGS 84	
Length of semi-major axis:	semiMajorAxis:	6 378 137.0 m	
Second defining parameter:	secondDefiningParameter:	inverseFlattening	
Inverse flattening:	inverseFlattening:	298.257 223 563	
ParametricCRS			The second component CRS is then described.
Parametric CRS name	name:	MICOM potential density CRS	
CRS scope	scope:	Oceanography	
CRS validity	domainOfValidity:	Global, oceans and seas	
ParametricCS			
Parametric coordinate system name	name:	Potential density in kg m^{-3}	
CoordinateSystemAxis			
Coordinate system axis name:	name:	Potential density	
Coordinate system axis abbreviation:	axisAbbrev:	PD	
Coordinate system axis direction	axisDirection:	Down	
Coordinate system axis unit identifier:	axisUnitID:	kg m^{-3}	
ParametricDatum			
Parametric datum name	name:	Sea surface	
Datum alias	alias:	Mean sea level	This is an optional attribute.
Datum anchor	anchorDefinition:	Mean sea level	This is an optional attribute.

E.4   Definition of temporal coordinate reference systems

Example E.4.1   Definition of a Temporal CRS in which axis quantity is DateTime

Presented here is a temporal coordinate reference system defined with respect to the ISO Gregorian calendar with coordinate values using dateTime representation conforming to ISO 8601.

Attribute	UML identifier	Data Entry	Comment
	TemporalCRS		
Temporal CRS name:	name:	DateTime in Gregorian calendar	The time zone is given or implied in the coordinate value, as defined in ISO 8601.
Temporal CRS scope:	scope:	Date/time as defined in ISO 8601.	This attribute is optional but recommended.
Temporal CRS temporal validity:	domainOfValidity:	1582-10-15 /	This attribute is optional but recommended. This example shows a character string entry: refer to ISO 19115-1 and ISO 8601.
	DateTimeTemporalCS		
DateTime temporal coordinate system name:	name:	DateTime	
Coordinate data type:	coordinateDataType:	dateTime	As defined in ISO 8601.
	Temporal CoordinateSystemAxis		For a DateTimeTemporalCS axisUnitID and its UnitOfMeasure are not required.
Coordinate system axis name:	axisName:	dateTime	
Coordinate system axis abbreviation:	axisAbbrev	T	
Coordinate system axis direction:	axisDirection:	Future	
	TemporalDatum		
Temporal datum name:	name:	Gregorian calendar	
Calendar:	calendar:	prolepticGregorian	This is the default value.
Temporal datum origin:	origin:	1875-05-20	
Temporal datum remarks:	remarks:	The origin is by convention the calendar day that the "Convention du Mètre" was signed in Paris.	This is an optional attribute.

Example E.4.2: **Definition of a Temporal CRS in which axis quantity is integer count**

Presented here is a temporal coordinate reference system defined with respect to an instance in time in the ISO Gregorian calendar. Coordinates defined with respect to this CRS are required to have integer values, interpreted as calendar hours.

Attribute	UML identifier	Data Entry	Comment
	TemporalCRS		
Temporal CRS name	name:	Calendar hours from 1979-12-29	
	TemporalCountCS		
Temporal count coordinate system name:	name:	calendar hours	
Coordinate data type:	coordinateDataType:	integer	
	Temporal CoordinateSystemAxis		
Coordinate system axis name:	name:	Hour	
Coordinate system axis abbreviation:	axisAbbrev	T	
Coordinate system axis direction:	axisDirection:	future	
Coordinate system axis temporal quantity:	axisUnitID:	hour	For temporal CRSs, the label 'temporal quantity' may be used instead of 'axis unit ID'. This 'hour' unit of measure is defined in isolation. It is explicitly not defined as a number of seconds.
	TemporalDatum		
Temporal datum name:	name:	1979-12-29	
Temporal datum origin:	origin:	1979-12-29T00:00:00.0Z	This is the temporal origin expressed in the Gregorian calendar following ISO 8601 syntax.
Calendar:	calendar:	prolepticGregorian	

If two coordinates defined with respect to this CRS have values of 36 and 96; then the corresponding dateTime strings for these coordinates are 1979-12-30T12:00Z and 1980-01-02T00:00Z respectively.

There is a leap second in the 96 calendar hours between 1979-12-29T00:00:00.0Z and 1980-01-02T00:00:00.0Z. The result accounts for this leap second, the arithmetic operates as an integer number of hours within the calendar. One of these 'hour' quantities within the calendar is 1 second longer than the others. This demonstrates that the behaviour is a calendar calculation.

Example E.4.3:   **Definition of a Temporal CRS in which axis quantity is measure**

Presented here is a temporal coordinate reference system defined with respect to an instance in time in the ISO Proleptic Gregorian calendar. Coordinates defined with respect to this CRS is required to have real (floating point) values, interpreted as years on a continuous year scale.

Attribute	UML identifier	Data Entry	Comment
	TemporalCRS		
Temporal CRS name:	name:	Decimal Years CE	
	TemporalMeasureCS		
Temporal coordinate system name:	name:	Decimal years	
Coordinate data type:	coordinateDataType:	real	
	Temporal CoordinateSystemAxis		
Coordinate system axis name:	name:	year	
Coordinate system axis abbreviation:	axisAbbrev	a	
Coordinate system axis direction:	axisDirection:	future	
Coordinate system axis temporal quantity:	axisUnitID:	year	
	TemporalDatum		
Temporal datum name:	name:	Common Era	
Temporal datum alias:	alias:	CE	
Temporal datum origin:	origin:	0000	This is the temporal origin expressed in the proleptic Gregorian calendar for date 0000-01-01 using reduced precision (following ISO 8601).
Calendar:	calendar:	prolepticGregorian	

Example E.4.4: **Definition of a Compound CRS containing independant Temporal CRSs (dateTime and count)**

Presented here is a compound CRS containing two independent TemporalCRS instances, one using dateTime and the other integer count. Each coordinate tuple contains two different time values, representing two different pieces of information. The first value in the tuple is the reference dateTime (time when the run was initialised) for a meteorological forecasting model run; the second value in the tuple is the forecast time, the time at which the forecast quantity is forecasted, encoded as a count, an integer offset. Such data representations are used in weather forecasting to provide trend analysis across multiple forecast runs.

2017-12-06T00:00:00Z, 24,
2017-12-06T00:00:00Z, 48,
2017-12-06T12:00:00Z, 24,
2017-12-06T12:00:00Z, 48

This CoordinateSet is referenced to the CompoundCRS defined below.

Attribute	UML identifier	Data Entry	Comment
	CompoundCRS		
Compound CRS name:	name:	Meteorological run E44 calendar	
Compound CRS scope:	scope:	Weather forecasting	This attribute is optional but recommended.
Compound CRS validity	domainOfValidity:	1875-05-20 /	This attribute is optional but recommended. This example shows a character string entry: refer to ISO 19115-1 and ISO 8601.
			The individual CRSs forming the compound CRS are next described. The sequence is significant as it implies the order of coordinates in the coordinate tuple.
	TemporalCRS 1		This is example E.4.1 above
Temporal CRS name:	name:	DateTime in Gregorian calendar	
:	:	:	
[...*full CRS definition for Temporal CRS 1, as in E.4.1 above, required here but omitted in this example for brevity*...]:			
:	**TemporalCRS 2**		This is example E.4.2 above.
	name:	Calendar hours from 1979-12-29	
Temporal CRS name:			
[...*full CRS definition for Temporal CRS 2, as in E.4.2 above, required here but omitted in this example for brevity*...]			

Example E.4.5:   **Definition of a Compound CRS containing independent Temporal CRSs** (dateTime and Measure)

Presented here is a compound CRS containing two independent TemporalCRS instances, one using dateTime and the other a temporal measure.

Annex D of this document suggests that converting a dateTime from a measure, a real number offset from a temporal origin, may not produce standardised results; implementations may differ. If a measure and a dateTime are required to be accurately defined for an entity, as one is not reliably able to be inferred from the other then a coordinate tuple of the dateTime and the real value measure may be provided. For example:

2016-02-28T12:00:00Z, 2016.1626,
2016-02-29T12:00:00Z, 2016.1653,
2017-02-28T12:00:00Z, 2017.1630,
2017-03-01T12:00:00Z, 2017.1658

This Coordinate Set is referenced to the CompoundCRS defined below.

Attribute	UML identifier	Data Entry	Comment
	CompoundCRS		
Compound CRS name:	name:	Meteorological run E45 calendar	
Compound CRS scope:	scope:	Weather forecasting for June 2017	This attribute is optional but recommended.
Compound CRS validity	domainOfValidity:	1875-05-20 /	This attribute is optional but recommended. This example shows a character string entry: refer to ISO 19115-1 and ISO 8601.
	TemporalCRS 1		This is example E.4.1 above
Temporal CRS name:	name:	DateTime in Gregorian calendar	
:	:	:	
[...*full CRS definition for Temporal CRS 1, as in E.4.1 above, required here but omitted in this example for brevity*...]			
:	**TemporalCRS 2**		This is example E.4.3 above.
	name:	Decimal Years CE	
Temporal CRS name:			
[...*full CRS definition for Temporal CRS 2, as in E.4.3 above, required here but omitted in this example for brevity*...]			

E.5 Definition of coordinate operations

Example E.5.1: **Definition of a coordinate transformation**.

This example shows a coordinate transformation from WGS 84 to ED50.

Attribute	UML identifier	Data Entry	Comment
	Transformation		
Coordinate operation name:	name:	WGS 84 to ED50 NIMA 1993 mean Europe	
Coordinate operation version:	operationVersion:	NIMA mean for Europe	
Coordinate operation scope:	scope:	Military operations	
Coordinate operation validity:	domainOfValidity:	Austria; Belgium; Denmark; Finland; France; Germany (west); Gibraltar; Greece; Italy; Luxembourg; Netherlands; Norway; Portugal; Spain; Sweden; Switzerland	
Coordinate operation remarks:	remarks:	For civilian applications consult EuroGeographics or national mapping authorities.	This field is optional.
Coordinate operation accuracy	coordinateOperationAccuracy:	3 m, 8 m and 5 m in X, Y and Z axes.	This field is optional but for transformations is recommended.
Source CRS:	**sourceCRS:**	WGS 84	(Additional data defining the source CRS should be given here but is not detailed in this example.)
Target CRS:	**targetCRS:**	ED50	(Additional data defining the target CRS should be given here but is not detailed in this example.)
	OperationMethod:		
Coordinate operation method name:	name:	geocentric translations	
Coordinate operation method formula:	formula	$X_t = X_s + dX$ $Y_t = Y_s + dY$ $Z_t = Z_s + dZ$ where dX, dY and dZ are translations along X, Y and Z axes, respectively. (The subscripts "t" and "s" refer to target and source.)	
	OperationParameter		The number of parameters (n) is dictated by the formula of the operation method. Parameter names, values (and, if required, optional attributes) will be given n times. In this example, $n = 3$.
Operation parameter name:	name:	X-axis translation	
	ParameterValue:		
Operation parameter numeric value:	value:	87 m	
	OperationParameter		
Operation parameter name:	name:	Y-axis translation	
	ParameterValue:		
Operation parameter numeric value:	value:	98 m	
	OperationParameter		

Attribute	UML identifier	Data Entry	Comment
Operation parameter name:	name:	*Z*-axis translation	
	ParameterValue:		
Operation parameter numeric value:	value:	121 m	

Example E.5.2: Definition of a geoid height model (coordinate transformation).

This example describes a model which transforms ellipsoid heights which are referenced to NAD83(CSRS) to gravity-related heights. By convention it realizes CGVD2013, a geoid-based vertical CRS described in example E.1.11.

Attribute	UML identifier	Data Entry	Comment
	Transformation		
Coordinate operation name:	name:	Canadian Gravimetric Geoid 2013	
Coordinate operation alias:	alias:	CGG2013	
Source CRS:	**sourceCRS**:	NAD83(CSRS)v6 LatLonEht	(Additional data defining the source CRS should be given here but is not detailed in this example.)
Target CRS:	**targetCRS**:	CGVD2013-OHt	(Additional data defining the target CRS should be given here but is not detailed in this example.)
Coordinate operation version:	operationVersion:	v1	
Coordinate operation scope:	scope:	Derivation of orthometric heights from GNSS-determined ellipsoidal heights..	
Coordinate operation validity:	domainOfValidity:	Canada	
Coordinate operation remarks:	remarks:	Source: M. Veronneau and J. Huang. The Canadian Geodetic Vertical Datum of 2013 (CGVD2013). Geomatica, Vol. 70, No. 1, 2016, pp. 9-19.	This attribute is optional.
	OperationMethod		
Coordinate operation method name:	name:	CGG geoid height model	
Coordinate operation method formula:	formula:	$h_{CSRS} = H_{CGVD} + N_{CGG}$	
Coordinate operation method remarks:	remarks:	The method requires bilinear interpolation of geoid height (*N*) within the geoid height model, using latitude and longitude components of the source CRS as arguments for the interpolation.	This attribute is optional.
			The number of parameters (*n*) is dictated by the formula of the operation method. Parameter names, values (and, if required, optional attributes) will be given *n* times, as appropriate. In this example, *n* is the value of each grid node in the geoid height model and is too large to be conveniently described directly. It is therefore given indirectly through a file reference. The format of the file will be dictated by the operation method.
	OperationParameter		
Operation parameter name:	name:	geoid height model file name	

Attribute	UML identifier	Data Entry	Comment
	ParameterValue		
Operation parameter file reference:	valueFile:	CGG2013n83.byn	Filename might be a URI.
	OperationParameter		
Operation parameter name:	name:		
	ParameterValue		
Interpolation CRS:	stringValue	NAD83(CSRS) v6 LatLonEht	This is the CRS used for interpolation in the gridded data set. In this example it is the source CRS so implicit in the operation method but for other methods (such as vertical offsets, e.g. from CGVD1928 to CGVD2013) needs to be stated.

Example E.5.3:   Definition of a Concatenated Operation.

This example demonstrates the concatenation of a transformation between Egypt 1907 to WGS 72 with one between WGS 72 and WGS 84 to form a concatenated operation between Egypt 1907 and WGS 84.

Attribute	UML identifier	Data Entry	Comment
	ConcatenatedOperation		
Concatenated coordinate operation name:	name	ED50 to WGS 84 Egypt	
Source CRS:	**sourceCRS:**	ED50	This is the source CRS for the concatenated coordinate operation. (Additional data defining the source CRS should be given here but is not detailed in this example.)
Target CRS:	**targetCRS:**	WGS 84	This is the target CRS for the concatenated coordinate operation. (Additional data defining the target CRS should be given here but is not detailed in this example.)
Coordinate operation version:	operationVersion:	MCE and DMA concatenation	
Coordinate operation scope:	scope:	Oil exploration.	
Coordinate operation validity:	domainOfValidity:	Egypt - Western Desert.	
			Then each of the single coordinate operations forming the concatenated operation are given in turn. The order is that in which the operations are to be made. In this example, only selected attributes are shown – see Example E.5.1 for a full example of a single coordinate operation.
	Transformation		The first step of the concatenated transformation is described next.
Coordinate operation name:	name:	ED50 to WGS 72 Egypt	
Source CRS:	**sourceCRS:**	ED50	This is the source CRS for the first step of the concatenated operation, in this example ED50.

Attribute	UML identifier	Data Entry	Comment
Target CRS:	**targetCRS:**	WGS 72	This is the target CRS for the first step of the concatenated operation, in this example WGS 84.
Coordinate operation version:	operationVersion:	MCE 1974	
Coordinate operation validity:	domainOfValidity:	Egypt.	
Coordinate operation scope:	scope:	Geodetic survey.	
	OperationMethod		
Coordinate operation method name	name:	geocentric translations	
Coordinate operation method formula	formula:	$X_t = X_s + \mathrm{d}X$ $Y_t = Y_s + \mathrm{d}Y$ $Z_t = Z_s + \mathrm{d}Z$ where dX, dY and dZ are translations along X, Y and Z axes respectively. (The subscripts "t" and "s" refer to target and source.)	
	OperationParameter		The number of parameters (n) is dictated by the formula of the operation method. Parameter names, values (and, if required, optional attributes) will be given n times, as appropriate. In this example, first step, $n = 3$.
Operation parameter name:	name:	X-axis translation	
	ParameterValue		
Operation parameter numeric value:	value:	−121.8 m	
	OperationParameter		
Operation parameter name:	name:	Y-axis translation	
	ParameterValue		
Operation parameter numeric value:	value:	98.1 m	
	OperationParameter		
Operation parameter name:	name:	Z-axis translation	
	ParameterValue		
Operation parameter numeric value:	value:	−15.2 m	
			The next step of the concatenated transformation is described.
	Transformation		
Coordinate operation name	name:	WGS 72 to WGS 84 DMA	
Source CRS	**sourceCRS:**	WGS 72	This is the source CRS for the second step of the concatenated operation, in this example WGS 72.
Target CRS:	**targetCRS:**	WGS 84	This is the target CRS for the second step of the concatenated operation, in this example WGS 84.
Coordinate operation version:	operationVersion:	DMA 1987	
Coordinate operation scope:	scope:	Geodetic survey.	
Coordinate operation validity:	domainOfValidity:	World.	
	OperationMethod		

Attribute	UML identifier	Data Entry	Comment
Coordinate operation method name:	name:	Helmert similarity transform (position vector rotation convention)	
Coordinate operation method formula	formula:	(The method formula or a citation for it should be given here and is not detailed in this example.)	
	OperationParameter		The number of parameters (n) is dictated by the formula of the operation method. Parameter names, values (and, if required, optional attributes) will be given n times, as appropriate. In the second step of this example, $n = 7$.
Operation parameter name:	name:	*X*-axis translation	
	ParameterValue		
Operation parameter numeric value:	value:	0 m	
	OperationParameter		
Operation parameter name:	name:	*Y*-axis translation	
	ParameterValue		
Operation parameter numeric value:	value:	0 m	
	OperationParameter		
Operation parameter name:	name:	*Z*-axis translation	
	ParameterValue		
Operation parameter numeric value:	value:	4.5 m	
	OperationParameter		
Operation parameter name:	name:	*X*-axis rotation	
	ParameterValue		
Operation parameter numeric value:	value:	0 s	
	OperationParameter		
Operation parameter name:	name:	*Y*-axis rotation	
	ParameterValue		
Operation parameter numeric value:	value:	0 s	
	OperationParameter		
Operation parameter name:	name:	*Z*-axis rotation	
	ParameterValue		
Operation parameter numeric value:	value:	0.554 s	
	OperationParameter		
Operation parameter name:	name:	Scale difference	
	ParameterValue		
Operation parameter numeric value:	value:	0.2263 parts per million	

E.6   Change of coordinate epoch in a dynamic CRS

Coordinate epoch is an attribute of data; change of coordinate epoch does not modify the definition of a CRS having a dynamic reference frame. An audit trail of transactions on coordinate data should include documentation of the CRS and, when the CRS is dynamic, the source and target coordinate epochs.

Example E.6.1:   Change of coordinate epoch using station velocities

Coordinates of the coordinate tuple are:

Station	Geocentric-X (m)	Geocentric-Y (m)	Geocentric-Z (m)
Alice Springs ITRF station ALIC	-4052052.148	4212836.068	-2545105.400

Coordinate metadata:
　　　Coordinate reference system: ITRF2008
　　　Coordinate epoch: 2005.0

Problem: To find coordinates of the station at epoch 2017.56 (target epoch). The point motion operation to be employed is:

Attribute	UML identifier	Data Entry	Comment
	PointMotionOperation		
Coordinate operation name:	name:	Change of coordinate epoch	
Source CRS:	**sourceCRS**:	ITRF2008 - XYZ	(Additional data defining the source CRS should be given here but is not detailed in this example.)
Target CRS:	**targetCRS**:	ITRF2008 - XYZ	(Additional data defining the target CRS should be given here but is not detailed in this example.)
Coordinate operation version:	operationVersion:	v1	
Coordinate operation scope:	scope:	Change of coordinate epoch	
Coordinate operation validity:	domainOfValidity:	World	
	OperationMethod		
Coordinate operation method name:	name:	Change of coordinate epoch using station velocities	
Coordinate operation method formula:	formula	$X_{t2} = X_{t1} + VX \bullet (t_2 - t_1)$ $Y_{t2} = Y_{t1} + VY \bullet (t_2 - t_1)$ $Z_{t2} = Z_{t1} + VZ \bullet (t_2 - t_1)$	
	OperationParameter		
Operation parameter name:	name:	VX	
	ParameterValue		
Operation parameter numeric value:	value:	-0.0396 m/yr	
	OperationParameter		
Operation parameter name:	name:	VY	
	ParameterValue		
Operation parameter numeric value:	value:	-0.0050 m/yr	
	OperationParameter		
Operation parameter name:	name:	VZ	
	ParameterValue		
Operation parameter numeric value:	value:	0.0541 m/yr	

After application of the point motion coordinate operation, coordinates of the coordinate tuple are:

Station	Geocentric-X (m)	Geocentric-Y (m)	Geocentric-Z (m)
Alice Springs ITRF station ALIC	-4052052.645	4212836.005	-2545104.721

Coordinate metadata:

Coordinate reference system: ITRF2008
Coordinate epoch: 2017.56

Note that the CRS is unchanged, but the coordinate epoch has changed (as have the coordinate values).

Example E.6.2:   Change of coordinate epoch using deformation or velocity model.

Coordinates of the coordinate tuple are:

Station	Geodetic Latitude	Geodetic Longitude	Ellipsoid Height
Centennial Monument NCC100 (Canadian Geodetic Survey station number 6530100)	45°25'45.714920"N	75°42'05.960075"W	39.524 m

Coordinate metadata:

Coordinate reference system: NAD83(CSRS) v6
Coordinate epoch: 2010.0

Problem: To find coordinates of the station at coordinate epoch 2002.00 (target epoch). The point motion operation to be employed is:

Attribute	UML identifier	Data Entry	Comment
	PointMotionOperation		
Coordinate operation name:	name:	Canadian Velocity Grid v6.0	
Coordinate operation alias:	alias:	CVG v6.0	
Source CRS:	**sourceCRS:**	NAD83(CSRS) v6 - LatLonEht	(Additional data defining the source CRS should be given here but is not detailed in this example.)
Target CRS:	**targetCRS:**	NAD83(CSRS) v6 - LatLonEht	(Additional data defining the target CRS should be given here but is not detailed in this example.)
Coordinate operation version:	operationVersion:	v6.0	
Coordinate operation scope:	scope:	Change of coordinate epoch for a coordinate set referenced to a NAD83(CSRS) v6	
Coordinate operation validity:	domainOfValidity:	Canada	
Coordinate operation accuracy:	coordinateOperationAccuracy:	0.02m	
Coordinate operation remarks:	remarks:	Accuracy ... in horizontal, ... in vertical.	
	OperationMethod		
Coordinate operation method name:	name:	NTv2_Vel	

Attribute	UML identifier	Data Entry	Comment
Coordinate operation method formula:	formula:	The method first requires bilinear interpolation of velocities within the velocity grid, using latitude and longitude components of the source CRS as arguments for the interpolation. Then: $V\varphi = VN / (\rho+h)$ $V\lambda = VE / [(\nu+h) \cos \varphi]$ $\varphi_{t2} = \varphi_{t1} + V\varphi \bullet (t_2 - t_1)$ $\lambda_{t2} = \lambda_{t1} + V\lambda \bullet (t_2 - t_1)$ $h_{t2} = h_{t1} + Vh \bullet (t_2 - t_1)$	ρ = radius of curvature of the reference ellipsoid in the meridian ν = radius of curvature of the reference ellipsoid in the prime vertical
	OperationParameter		
Operation parameter name:	name:	velocity model	
	ParameterValue		
Operation parameter file reference:	valueFile:	cvg60.cvb	Filename might be a URI.

Interpolation of the velocity grid for the station velocities at Centennial Monument gives

      VN = north velocity = -0.00156 m/yr

      VE = east velocity = 0.00177 m/yr

      Vh = vertical velocity = 0.00202 m/yr

from which

      $V\varphi$ = latitude velocity = -5.05E-5 arc-seconds / yr

      $V\lambda$ = longitude velocity = 8.14E-5 arc-seconds / yr

may be calculated.

After application of the point motion operation for δt = -8.0 years, coordinates of the coordinate tuple are:

Station	Geodetic Latitude	Geodetic Longitude	Ellipsoidal Height
Centennial Monument NCC100 (Canadian Geodetic Survey station number 6530100)	45°25'45.715324"N	75°42'05.960726"W	39.508 m

Coordinate metadata:

      Coordinate reference system: NAD83(CSRS) v6

      Coordinate epoch: 2002.0

Note that the CRS is unchanged, but the coordinate epoch has changed (as have the coordinate values).

Annex F: Recommended best practice for interfacing to ISO 19111
(informative)

Standards which reference the ISO 19111 UML model can minimize dependencies by only referencing to the following:

a) An interface to the ISO 19111 model which requires coordinate reference system definition should be only to the CRS class or any one of its concrete subclasses GeodeticCRS, GeographicCRS, ProjectedCRS, VerticalCRS, EngineeringCRS, ParametricCRS, TemporalCRS and their derived equivalents, or to CompoundCRS.

   Interfaces to Datum (including datum subclasses) or to CoordinateSystem will not provide for a complete coordinate reference system definition and should not be made.

b) If a coordinate operation is required, interfacing should be made through the CoordinateOperation class.

c) For interfacing coordinate sets to an instance of a coordinate reference system, refer to Clause 7. For dynamic CRSs, interfacing to the DataEpoch class will also be required.

Annex G: Backward compatibility with ISO 19111:2007 (informative)

Overview

The following is a summary of changes made to the data modelling in the previous edition of this document. These changes have been introduced to support:

− the definition of dynamic geodetic and vertical reference frames;
− the definition of Earth-centred Cartesian geodetic reference frames without ellipsoid being a mandatory attribute;
− the definition of geoid-based vertical CRSs;
− the addition of requirements to describe coordinate metadata;
− the definition of 3D projected CRSs (map projection easting and northing with ellipsoidal height);
− the addition of datum ensembles to permit the merging without transformation of coordinate sets referenced to different realizations of a terrestrial reference system (TRS) which, for lower accuracy applications, may be considered insignificantly different;
− clarification in the definitions and data modelling of derived CRSs;
− the addition of temporal datum and temporal coordinate system classes to support the definition of temporal coordinate reference systems suitable for spatio-temporal referencing without reference to ISO 19108;
− the absorption of ISO 19111-2:2009, extension for parametric values, into this document to have the full data model for CRS definition consolidated into one document;
− the upgrade of UML from version 1 to version 2, in accordance with changes made in ISO 19103: this includes the removal of package identifiers;
− the correction of minor errors in the supporting text.

The ImageCRS class of 19111:2007 has been removed from this document. It was an incomplete modelling of a grid, which is fully documented in ISO 19123[5]. However this document includes an additional coordinate system type, ordinal CS, to describe CSs with continuous axes having discrete coordinate values.

Because of the inclusion of the coordinate reference system subtypes of parametric and temporal, the title of the document has been changed from'Spatial referencing by coordinates' to 'Referencing by coordinates'.

The majority of changes are extensions to the data model or modifications to the modelling that have no impact on implementation. Several constraints on associations in the previous version of the data model have been moved to be on classes. The provisions of ISO 19111-2:2009, extension for parametric values, that have been brought into this document are unchanged other than correcting a minor error in the UML diagram (19111-2:2009 figure 1) for the name of the association from Compound CRS to SingleCRS.

To accord with the latest version of the ISO Directives part 2, there has been some rearrangement of the clauses in this document compared to the previous edition:
• 19111:2007 clause 2 (Conformance) becomes clause 4 in this document;
• 19111:2007 clause 3 (Normative references) becomes clause 2 in this document;
• 19111:2007 clause 4 (Terms and definitions) becomes clause 3.1 in this document;
• 19111:2007 clause 5.1 (Symbols) becomes clause 3.2 in this document.
• 19111:2007 clause 5.2 (Abbreviated terms) becomes clause 3.3 in this document;
• 19111:2007 clause 5.3 (UML notation) becomes clause 5.1 in this document; and
• 19111:2007 clause 5.4 (Attribute status) becomes clause 5.2 in this document.

Clause 5, conventions

Clause 5.1 and data model descriptions.
UML updated from v1 to v2. The data type <<Type>> has been changed to <<Interface>>. The package prefixes (IO, SC, CS, CD and CC) have been dropped.

Errata: throughout the 19111:2007 text, the package names were singular: in this version they have been made plural to accord with the UML model.

Clause 7, Coordinates package

19111:2007 clause 6 has been split into two. These revised clauses separate the overview of the UML model (this document clause 6) from the description of the relationship between coordinates and the CRS to which they are referenced (this document clause 7). A new Coordinates package with three new classes (DataEpoch, CoordinateSet and CoordinateMetadata) has been introduced to facilitate the description of referencing by coordinates.

As a consequence of the revision of ISO 19107:2003 and to reflect changes made to this in ISO 19107:2018, the associations of the CRS class from DirectPosition and GM_Object have been replaced by one from Geometry, with DirectPosition used as a data type for coordinate tuple. The description of the UML model describing this been moved from the CRS package to this Clause 7; the requirements for describing a CRS remain in the CRS package (now Clause 9).

Clause 8, Common Classes package (renamed from Identified Objects in 19111:2007)

Following the removal of RS_ReferenceSystem from 19115-1:2014, the 19111:2007 IdentifiedObjectBase class has been removed. Its attributes (identifier, alias and remarks) have been moved to the IdentifiedObject class. Inheritance of these attributes by classes in other packages is unaffected.

The RS_Identifier data type was removed from 19115-1:2014 and replaced with MD_Identifier. This replacement has been mirrored in this revision to 19111. As MD_Identifier is a superset of RS_Identifier there are no backward compatibility issues.

Scope and DomainOfValidity attributes
In 19111:2007 these attributes were separately included in each of the CRS, Datum and CoordinateOperation classes, but have now been moved from those classes to the Common Classes package from where they are inherited by those classes through subtyping.

To facilitate descriptions of usage such as 'purpose 1 in area A, purpose 2 in area B', two new classes (ObjectUsage and ObjectDomain) have been added. This creates a pairing of the Scope and DomainOfValidity attributes, previously used individually in ISO 19111:2007. The cardinality of Scope has been changed from mandatory to optional, that for domainOfValidity remains optional, but these two attributes now cannot be provided separately.

Clause 9, Coordinate Reference Systems package

CRS class. The subtyping of the CRS class off RS_ReferenceSystem has been removed. This follows the removal of that class from the 2014 revision of ISO 19115-1. The attributes previously inherited through this subtyping (Name and DomainOfValidity) have been included in the Common Classes package and are now inherited through this. The attribute *Scope* has been moved to the Common Classes package (see above) from which it is inherited through subtyping.

An additional association from CRS to CoordinateOperation, *InterpolationCRS*, has been added. This is to facilitate the identification of the CRS to be used for interpolation within a file of gridded geodetic data when it is neither the source CRS or the target CRS for the coordinate operation.

SingleCRS. A constraint has been added to enforce the association to either a datum or a datum ensemble (see Clause 11 below).

GeodeticCRS. A new subtype, GeographicCRS, has been added. Although the model is unchanged in that Geodetic CRS may associated with a Cartesian, spherical CS or ellipsoidal CS (see Clause 10 below), it is recommended that a geodetic CRS having an ellipsoidalCS be deprecated and replaced by a geographic CRS. A constraint has been added to the class to require an ellipsoid for a geographic CRS: an ellipsoid becomes optional for a geodetic CRS having Cartesian or spherical CS. (See Clause 11 below for further related information).

A new optional association to a new PointMotionOperation class (see Clause 12 below) has been added to facilitate the linking of a dynamic CRS to a velocity or deformation model.

VerticalCRS. A new optional association to Transformation has been added to facilitate the description of geoid-based vertical CRSs. A new optional association to a new PointMotionOperation class (see clause 12 below) has been added to facilitate the linking of a dynamic CRS to a velocity model.

ProjectedCRS. The possibility of being 3-dimensional (with ellipsoid height) has been added.

DerivedCRS. The UML modelling has been significantly changed to clarify implementation.
− Seven new classes (Derived*CRS) have been added.
− The SC_DerivedCRS class and derivedCRSType codelist have been removed, and the SC_GeneralDerivedCRS class renamed DerivedCRS. Constraints to enforce the inheritance of the baseCRS's datum have been added to the Derived CRS class.
− The simple association to the coordinate Conversion class has been changed to an aggregation.
− The erroneous statement prohibiting a projectedCRS being used as the baseCRS for a derived CRS in the description of the defining elements of the SC_DerivedCRS class (19111:2007, Table 8) has been excluded from this revision; the text is now harmonised with the UML model.

TemporalCRS. In 19111:2007 this was a hook to an anticipated revision to the 19108 model. In this revision a new class has been created in this package.

ParametricCRS. This class has been brought from 19111-2 into this package, to consolidate all information into a single document and model. It is otherwise unchanged.

ImageCRS. This class from 19111:2007 has been removed. It was an incomplete attempt to describe grids. These are fully described in ISO 19123[6].

Clause 10, CoordinateSystems package

CartesianCS. This is now subtyped off CoordinateSystem indirectly through the affineCS class.

OrdinalCS. This new class has been added. It is constrained to have integer coordinate values.

TemporalCS. This new class has been added, together with three subtypes and a new supporting class CoordinateDataType.

ParametricCS. This class has been brought from 19111-2 into this package. It is otherwise unchanged.

UserDefinedCS. This class from 19111:2007 has been removed. It did not assist interoperability.

DerivedProjectedCS. This new union class has been added

ImageCS. This union class from 19111:2007, used only by ImageCRS, has been removed.

EngineeringCS. userDefinedCS has been removed from this union, ordinalCS has been added.

CoordinateSystemAxis. The axisUnitID attribute in the CoordinateSystemAxis class has been made conditional, the condition enforced throughconstraints in the CoordinateSystem class (that it is mandatory except when this is overridden through the new DateTimeTemporalCS and OrdinalCS classes).. There is no practical impact on CS subtypes from 19111:2007.

AxisDirection code list: attributes 'Future' and 'Past' have been added.

Clause 11, Datums package

Datum. The realizationEpoch attribute (which had data type of Date) has been removed because of an ambiguous definition and replaced by two attributes:
   i)   publicationDate, with data type Date, retained in the Datum class.
   ii)   frameReferenceEpoch, with data type Measure (allowing a number rather than date), included within new DynamicGeodeticReferenceFrame and DynamicVerticalReferenceFrame classes and mandatory for these subtypes (see below).
Backward compatibility may be maintained by mapping realizationEpoch to publicationDate. However any data that was describing a dynamic reference frame's frame reference epoch should be moved to frameReferenceEpoch and have its data type changed from Date to Measure.

A new optional attribute *conventionalRS* has been added to support datum ensembles (see below).

GeodeticDatum and VerticalDatum. These classes have been renamed GeodeticReferenceFrame and VerticalReferenceFrame respectively. Each has a new subtyped class for dynamic reference frames (containing the frameReferenceEpoch attribute mentioned above). A new optional attribute to describe the realization method has been added to the VerticalReferenceFrame class, implemented through a new code list *RealizationMethod*.

The association from GeodeticReferenceFrame to Ellipsoid has been changed from mandatory to conditional. The condition is that it remains mandatory if the CRS subtype has an ellipsoidal CS, i.e. is geographic. For geodetic CRSs with other CS types (notably Cartesian) the association is now optional.

Ellipsoid. An additional optional attribute, semiMedianAxis, has been added to support triaxial ellipsoids.

ImageDatum. This class from 19111:2007 has been removed, together with the PixelInCell codelist used only by the ImageDatum class..

ParametricDatum. This class has been brought from 19111-2 into this package. A new optional attribute *datumDefiningParameter* has been added, implemented through a new *definingParameter* class.

TemporalDatum. A new class and Calendar codelist have been added.

DatumEnsemble. This new class has been added, with association to Datum. A constraint has been added to the SingleCRS class to enforce its association to either a datum or a datum ensemble.

Errata. Errors in the description of associations given in 19111:2007, Tables 38 and 41 (Engineering and Vertical datums) have been corrected in Tables 61 and 56 respectively in this document (to bring these descriptions in line with the UML data model).

Clause 12, Coordinate Operation package

CoordinateOperation. An additional association from CoordinateOperation to CRS, *InterpolationCRS*, has been added. Two new associations to the new Coordinates::DataEpoch class have been added.

Transformation. An additional association from VerticalCRS, *HeightTransfomation*, has been added. Constraints on the class use of the associations to CRS (inherited through CoordinateOperation) have been moved from being on the association to on the class (this has no practical consequence) and

constraints on the class use of associations to the new Coordinates::DataEpoch class have been added.

Conversion. The association from derivedCRS to Conversion has been changed from a simple association to an aggregation. Constraints on the class use of the associations to CRS (inherited through CoordinateOperation) have been moved from being on the association to being on the class (this has no practical consequence) and constraints on the class use of associations to the new Coordinates::DataEpoch class have been added.

PointMotionOperation. This new class has been added.

RegisterOperations. This new class has been added.

OperationMethod. The two attributes *sourceDimension* and *targetDimension* have been removed. This information is part of a CRS definition. These attributes were redundant and potentially could cause data conflicts.

GeneralOperationParameter. The attribute *minimumOccurs* has been moved to the OperationParameterGroup class. (It therefore now cannot be utilised by the OperationParameter class, but no reason for it to do so could be demonstrated and no known implementation of this could be found).

parameterValue. An additional attribute, *geographicObject*, has been added, implemented through a new GeographicObject class.

Errata:
1. In the description of the Defining elements of PassThroughOperation class (19111:2007, Table 47 and Table 66 in this document) the inheritance has been corrected from SingleOperation to CoordinateOperation (brings the description in line with the UML model).

2. In the description of the Defining elements of Formula class (19111:2007,Table 49 and Table 73 in this document) the stereotype has been corrected from CodeList to Union (brings the description in line with the UML model).

Bibliography

[1] HOOIJBERG, M. *Practical Geodesy*, Springer, 1997 [4]

[2] *IERS Conventions (2010)* — International Earth Rotation and Reference Systems Service (IERS) Technical Note No. 36ISO 19107:2003, *Geographic information — Spatial schema*

[3] ISO 19107:2*003, Geographic information — Spatial schema*

[4] ISO 19108*, Geographic information — Temporal schema*

[5] ISO 19112, *Geographic information — Spatial referencing by geographic identifier*s

[6] ISO 19123, *Geographic information — Schema for coverage geometry and functions*

[7] ISO 19148, *Geographic information — Linear referencing*

[8] ISO 19157, *Geographic information — Data Quality*

[9] ISO 19161-1, *Geographic information — The International Terrestrial Reference System (ITRS)*

[10] ISO 19162, *Geographic information — Well-known text representation of coordinate reference systems*

[11] ISO/IEC 19501, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*

[12] ISO/IEC/IEEE 9945:2009, *Information technology — Portable Operating System Interface (POSIX®) Base Specifications, Issue 7.* The POSIX formula is in 4.16, Seconds Since the Epoch.

[13] ISO 80000-3, *Quantities and units — Part 3: Space and time*

[14] SNYDER, JOHN P. *MAP PROJECTIONS: A WORKING MANUAL*, USGS PROFESSIONAL PAPER 1395, 1987, 383 PP

Parametric bibliography:

[15] ISO 2533:1975, *Standard Atmosphere*, amended by ISO 2533:1975/Add 1:1985, *Addendum 1: Hypsometrical tables* and ISO 2533:1975/Add 2:1997, Addendum 2: *Extension to − 5000 m and standard atmosphere as a function of altitude in feet*

[16] Doc 7488, *Manual of the ICAO Standard Atmosphere: extended to 80 kilometres (262 500 feet).* International Civil Aviation Organisation (ICAO), Third Edition, 1993

[17] *The Miami Isopycnal Coordinate Model*, 2000, available at http://oceanmodeling.rsmas.miami.edu/micom/

---

[4] This and similar literature describing the geodetic concepts incorporated within this document may use different terminology to that defined herein. Some terms may be common to both documents but have different meanings.