Copyright notice

This ISO document is an International Standard and is copyright-protected by ISO. Except as permitted under the applicable laws of the user’s country, neither this ISO draft nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured.

Requests for permission to reproduce should be addressed to either ISO at the address below or ISO’s member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Reproduction may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.
# ISO 19123-1:20xx(E)

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>vii</td>
<td>Foreword</td>
<td>vii</td>
</tr>
<tr>
<td>viii</td>
<td>Introduction</td>
<td>viii</td>
</tr>
<tr>
<td>1</td>
<td>Scope</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Normative references</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Terms, definitions, abbreviated terms and notation</td>
<td>1</td>
</tr>
<tr>
<td>3.1</td>
<td>Terms and definitions</td>
<td>1</td>
</tr>
<tr>
<td>3.2</td>
<td>Abbreviated terms</td>
<td>9</td>
</tr>
<tr>
<td>3.3</td>
<td>Notation</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Conformance</td>
<td>10</td>
</tr>
<tr>
<td>4.1</td>
<td>Interoperability and Conformance Testing</td>
<td>10</td>
</tr>
<tr>
<td>4.2</td>
<td>Organization</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Coverages</td>
<td>12</td>
</tr>
<tr>
<td>5.1</td>
<td>Overview</td>
<td>12</td>
</tr>
<tr>
<td>5.2</td>
<td>Coverage Schema</td>
<td>13</td>
</tr>
<tr>
<td>5.3</td>
<td>Probing Coverages: evaluate() Function</td>
<td>13</td>
</tr>
<tr>
<td>5.4</td>
<td>Domain of a coverage</td>
<td>14</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Concept</td>
<td>14</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Coordinates</td>
<td>14</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Coordinate Reference Systems for Coverage Domains</td>
<td>15</td>
</tr>
<tr>
<td>5.4.4</td>
<td>Multi-Dimensional Coverages</td>
<td>17</td>
</tr>
<tr>
<td>5.4.5</td>
<td>Mathematical vs Physical Coordinates</td>
<td>17</td>
</tr>
<tr>
<td>5.5</td>
<td>Range of a coverage</td>
<td>18</td>
</tr>
<tr>
<td>5.6</td>
<td>Interpolation</td>
<td>18</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Concept</td>
<td>18</td>
</tr>
<tr>
<td>5.6.2</td>
<td>Discrete and Continuous Coverages</td>
<td>19</td>
</tr>
<tr>
<td>5.6.3</td>
<td>Interpolation Set</td>
<td>20</td>
</tr>
<tr>
<td>5.7</td>
<td>Common Point Rule</td>
<td>20</td>
</tr>
<tr>
<td>5.8</td>
<td>Realization Variants</td>
<td>21</td>
</tr>
<tr>
<td>5.8.1</td>
<td>Geometry/Value Pair View</td>
<td>21</td>
</tr>
<tr>
<td>5.8.2</td>
<td>Domain/Range View</td>
<td>21</td>
</tr>
<tr>
<td>5.8.3</td>
<td>Partitioned View</td>
<td>22</td>
</tr>
<tr>
<td>5.8.4</td>
<td>Functional View</td>
<td>23</td>
</tr>
<tr>
<td>5.9</td>
<td>Envelope</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>Multi-Point Coverages</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>Grid Coverages</td>
<td>24</td>
</tr>
<tr>
<td>7.1</td>
<td>Overview</td>
<td>24</td>
</tr>
<tr>
<td>7.2</td>
<td>Grids</td>
<td>24</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Grid Definition</td>
<td>24</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Grid Axis Types</td>
<td>26</td>
</tr>
<tr>
<td>7.2.3</td>
<td>Grid Offsets</td>
<td>29</td>
</tr>
<tr>
<td>7.3</td>
<td>Grid Cells</td>
<td>29</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Grid cells</td>
<td>29</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Pixel-in-center, Pixel-in-corner</td>
<td>29</td>
</tr>
<tr>
<td>7.4</td>
<td>Grid Coverage</td>
<td>29</td>
</tr>
<tr>
<td>7.5</td>
<td>Rectified and Referenceable Grid Coverages</td>
<td>30</td>
</tr>
<tr>
<td>7.6</td>
<td>Further Grid Coverages</td>
<td>32</td>
</tr>
<tr>
<td>8</td>
<td>Multi-Curve Coverages</td>
<td>33</td>
</tr>
<tr>
<td>8.1</td>
<td>Overview</td>
<td>33</td>
</tr>
<tr>
<td>8.2</td>
<td>General Multi-Curve Coverages</td>
<td>33</td>
</tr>
<tr>
<td>8.3</td>
<td>Segmented curve coverages</td>
<td>34</td>
</tr>
</tbody>
</table>
ISO 19123-1:20xx(E)

9 Multi-Surface Coverages ........................................................................................................... 34
  9.1 Overview .................................................................................................................................. 34
  9.2 General Multi-Surface Coverages ............................................................................................... 34
  9.3 Further Surface Coverages ......................................................................................................... 35
    9.3.1 General ............................................................................................................................... 35
    9.3.2 Thiessen Polygon Coverages ............................................................................................... 35
    9.3.3 Triangulated Irregular Networks (TINs) ............................................................................. 36
10 Multi-Solid Coverages .................................................................................................................. 36
Annex A (normative) Conformance Tests ....................................................................................... 38
  A.1 Conformance Class .................................................................................................................... 38
  A.2 Conformance Class Coverage Core ........................................................................................... 38
    A.2.1 Requirement 1 ....................................................................................................................... 38
    A.2.2 Requirement 2 ....................................................................................................................... 38
    A.2.3 Requirement 3 ....................................................................................................................... 38
    A.2.4 Requirement 4 ....................................................................................................................... 38
    A.2.5 Requirement 5 ....................................................................................................................... 39
    A.2.6 Requirement 6 ....................................................................................................................... 39
    A.2.7 Requirement 7 ....................................................................................................................... 39
    A.2.8 Requirement 8 ....................................................................................................................... 39
    A.2.9 Requirement 9 ....................................................................................................................... 39
    A.2.10 Requirement 10 ...................................................................................................................... 39
    A.2.11 Requirement 11 .................................................................................................................... 39
    A.2.12 Requirement 12 .................................................................................................................... 39
    A.2.13 Requirement 13 .................................................................................................................... 40
    A.2.14 Requirement 14 .................................................................................................................... 40
    A.2.15 Requirement 15 .................................................................................................................... 40
    A.2.16 Requirement 16 .................................................................................................................... 40
    A.2.17 Requirement 17 .................................................................................................................... 40
    A.2.18 Requirement 18 .................................................................................................................... 40
    A.2.19 Requirement 19 .................................................................................................................... 40
  A.3 Conformance Class Multi-Point Coverage .................................................................................. 41
    A.3.1 Requirement 20 ...................................................................................................................... 41
    A.3.2 Requirement 21 ...................................................................................................................... 41
    A.3.3 Requirement 22 ...................................................................................................................... 41
  A.4 Conformance Class Grid Coverage ............................................................................................ 41
    A.4.1 Requirement 23 ...................................................................................................................... 41
    A.4.2 Requirement 24 ...................................................................................................................... 41
    A.4.3 Requirement 25 ...................................................................................................................... 41
    A.4.4 Requirement 26 ...................................................................................................................... 41
    A.4.5 Requirement 27 ...................................................................................................................... 42
    A.4.6 Requirement 28 ...................................................................................................................... 42
    A.4.7 Requirement 29 ...................................................................................................................... 42
    A.4.8 Requirement 30 ...................................................................................................................... 42
    A.4.9 Requirement 31 ...................................................................................................................... 42
  A.5 Conformance Class Multi-Curve Coverage .................................................................................. 42
    A.5.1 Requirement 32 ...................................................................................................................... 42
    A.5.2 Requirement 33 ...................................................................................................................... 43
  A.6 Conformance Class Multi-Surface Coverage .............................................................................. 43
    A.6.1 Requirement 34 ...................................................................................................................... 43
    A.6.2 Requirement 35 ...................................................................................................................... 43
  A.7 Conformance Class Multi-Solid Coverage ................................................................................... 43
    A.7.1 Requirement 36 ...................................................................................................................... 43
    A.7.2 Requirement 37 ...................................................................................................................... 43
Annex B (informative) Interpolation methods ................................................................. 44
B.1 General .................................................................................................................. 44
B.2 Nearest neighbour interpolation ................................................................. 45
B.3 Linear interpolation .......................................................................................... 45
B.4 Quadratic interpolation .................................................................................... 45
B.5 Cubic interpolation ............................................................................................ 46
B.6 Lost area interpolation ....................................................................................... 46
B.7 Barycentric interpolation ................................................................................... 47
B.8 Other .................................................................................................................... 47
Annex C (informative) Sequential enumeration .......................................................... 48
C.1 General ................................................................................................................ 48
C.2 Linear scanning .................................................................................................. 49
C.3 Boustrophedonic scanning .............................................................................. 49
C.4 Cantor-diagonal scanning ............................................................................... 49
C.5 Spiral scanning .................................................................................................. 50
C.6 Morton order ...................................................................................................... 50
C.7 Hilbert order ...................................................................................................... 51
C.8 Interleaving of feature attribute values ......................................................... 51
Annex D (normative) Data-Centric Coverage Specification ...................................... 54
D.1 Overview .............................................................................................................. 54
D.2 Generic Data Structure Coverage Schema ...................................................... 54
D.2.1 Generic Data Structure Package .................................................................. 54
D.2.2 Class CV_Coverage as a Data Structure Template ...................................... 55
D.2.3 CV_DomainObject ......................................................................................... 57
D.2.4 CV_AttributeValues ...................................................................................... 57
D.2.5 CV_FunctionalElement ................................................................................. 58
D.2.6 CV_ValueObject ............................................................................................ 58
D.2.7 CV_GeometryValuePair .............................................................................. 59
D.2.8 CV_CommonPointRule .................................................................................. 59
D.2.9 CV_CommonPointRule .................................................................................. 60
D.3 Coverage types .................................................................................................. 61
D.3.1 Discrete and Continuous Coverages ............................................................. 61
D.3.2 Multi-Point Coverages .................................................................................. 61
D.3.3 Multi-Curve Coverage .................................................................................. 61
D.3.4 Multi-Surface Coverage ................................................................................ 62
D.3.5 Multi-Solid Coverage .................................................................................... 62
D.3.6 Grid Coverage ............................................................................................... 62
D.4 Subclassing CV_GeometryValueObject .......................................................... 63
D.4.1 General ............................................................................................................ 63
D.4.2 CV_ValueCurve ............................................................................................. 63
D.4.3 CV_CurveValuePair ...................................................................................... 64
D.4.4 CV_ValueSegment .......................................................................................... 65
D.4.5 CV_PointValuePair ...................................................................................... 65
D.4.6 CV_ThiessenValuePolygon .......................................................................... 65
D.4.7 CV_ValueTriangle .......................................................................................... 66
D.4.8 CV_ValueHexagon .......................................................................................... 66
© ISO 2020– All rights reserved
D.4.9  8.12 CV_GridValueCell ................................................................. 67
D.4.10 CV_GridPointValuePair ............................................................. 67
D.4.11 CV_GridCell ................................................................. 67
D.4.12 CV_CurveValuePair .............................................................. 67
D.4.13 CV_SurfaceValuePair .............................................................. 68
D.4.14 CV_SolidValuePair ............................................................... 68
D.4.15 CV_Grid ................................................................. 68
D.4.16 GridEnvelope ................................................................. 69
D.4.17 CV_GridPoint ............................................................... 69
D.4.18 CV_GridCoordinate ............................................................. 69
D.4.19 CV_GridValuesMatrix ............................................................ 69
D.4.20 CV_GridCell ............................................................... 70
D.4.21 CV_SequenceRule ............................................................... 70
D.4.22 CV_SequenceType ............................................................... 70
Annex E (informative) Backward Compatibility .............................................. 71
E.1 General ..................................................................................... 71
E.2 Changes .................................................................................. 71
Bibliography .................................................................................. 73
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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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.

ISO 19123-1 was prepared by Technical Committee ISO/TC 211, Geographic information/Geomatics.
Introduction

This standard defines, at a high, implementation-independent level, the notion of coverages as digital representations of space-time varying geographic phenomena, corresponding to a field in physics: a physical quantity that has a value for each point in space-time. Such coverages can be discrete or continuous.

Historically, geographic information has been treated in terms of two fundamental types called vector data and raster data.

“Vector data” deals with discrete phenomena, each of which is conceived of as a feature. The spatial characteristics of a discrete real-world phenomenon are represented by a set of one or more geometric primitives (points, curves, surfaces or solids). Other characteristics of the phenomenon are recorded as feature attributes. Usually, a single feature is associated with a single set of attribute values. ISO 19107:2019 provides a schema for describing features in terms of geometric and topological primitives.

“Raster data”, on the other hand, deals with real-world phenomena that vary over space and time. It contains a set of values, each associated with one of the elements in an array of points or cells. It is usually associated with a method for interpolating values at spatial positions between the points or within the cells. Since this data structure is not the only one that can be used to represent phenomena that vary continuously over space, this document uses the term “coverage,” adopted from the Abstract Specification of the Open GIS Consortium[1], to refer to any data representation that assigns values directly to spatio-temporal position. A coverage is a function from a spatial, temporal or spatio-temporal domain to an attribute range. A coverage associates a position within its domain to a record of values of defined data types.

A coverage function has as its domain, an area or space defined by any combination of the three physical spatial dimensions plus the physical dimension time. Mathematics also uses the word dimension to represent an axis in a numeric space. The mathematical meanings of dimension and space are broader than those used in the physical world. The three physical spatial dimensions plus the physical dimension time may be mapped to mathematical dimensions. The range of a coverage function is a set of attribute values for each of the attribute types. These range values may also be represented as mathematical dimensions. That is, we have two complementary ways of viewing a coverage function, as a domain and range or as a mathematical space based on axes.

In this document, coverage is modelled as a subtype of feature as defined in ISO 19101. A coverage is a feature that has multiple values for each attribute type, where each direct position within the geometric representation of the feature has a single value for each attribute type.

A coverage consists of spatio-temporally extended objects where information content depends on (and varies with) the particular coordinate where it is probed. Standardization in this area is a cornerstone for other geographic information design, specification and standardization.

Such space-time varying objects are described as sets of geographic objects (“features”), called coverages. The feature objects collected in a coverage define the positions where values are available (called Direct Positions), and the individual values associated with each feature.

Note 1 Direct Positions can be of different dimension. For example, in a raster image modelled as a coverage the Direct Positions will be the grid points; in a Multi-Solid Coverage a Direct Position is given by the interior of a 3D solid.

In practice, coverages encompass regular and irregular grids, point clouds, and general meshes. Examples include raster data, triangulated irregular networks, point sets and polygon coverages. Coverages are multi-dimensional, including examples like 1D sensor timeseries, 2D satellite images, 3D x/y/t image timeseries and x/y/z geophysical voxel data, and 4D x/y/z/t climate and ocean data. Axes of such coverages can have spatial, temporal, or any other dimension, and they can be combined freely.

1 “Raster” is a widely used but imprecise colloquial term that encompasses imagery, gridded and other types of coverage data.
EXAMPLE  The electromagnetic spectrum is an example for an axis with neither spatial nor temporal semantics. As such a spectral axis can be defined following the rules of ISO 19111, so it qualifies as a coverage axis.

A coverage which provides values only at the Direct Positions is called “a discrete coverage” (discrete in its domain); if interpolation information is added so that values can be obtained also beyond the coverage’s Direct Positions such a coverage is called “a continuous coverage”.

Just as the concepts of discrete and continuous phenomena are not mutually exclusive, their representations as discrete features or coverages are not mutually exclusive. The same phenomenon may be represented as either a discrete feature or a coverage. A city may be viewed as a discrete feature that returns a single value for each attribute, such as its name, area and total population. The city feature may also be represented as a coverage that returns values such as population density, land value or air quality index for each position in the city.

A coverage, moreover, can be derived from a collection of discrete features with common attributes, the values of the coverage at each position being the values of the attributes of the feature located at that position. Conversely, a collection of discrete features can be derived from a coverage; each discrete feature being composed of a set of positions associated with specified attribute values.

The previous version of this standard ISO 19123:2005 addressed coverage modelling on both conceptual and (to some extent) implementation level. For this edition of the document, coverage modelling has been split into two separate, but connected documents: ISO 19123-1 (this document) establishes an abstract, high-level coverage model while ISO 19123-2 establishes an implementation-level model ensuring interoperability, based on the concepts of ISO 19123-1.
Geographic information — Schema for coverage geometry and functions -- Part 1: Fundamentals

1 Scope

This document defines a conceptual schema for coverages. A coverage is a mapping from a spatial, temporal or spatio-temporal domain to attribute values sharing the same type within the domain. A coverage domain consists of a collection of direct positions in a coordinate space that may be defined in terms of spatial and/or temporal dimensions. Examples of coverages include meshes/grids, triangulated irregular networks, point coverages and polygon coverages. Coverages are the prevailing data structures in a number of application areas, such as remote sensing, meteorology and mapping of bathymetry, elevation, soil and vegetation. This document defines the relationship between the domain of a coverage and an associated attribute range. The characteristics of the domain are defined whereas the characteristics of the attribute range are not part of this standard.

2 Normative references

The following referenced documents are indispensable for the application 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 19103:2015 Geographic information — Conceptual schema language
ISO 19107:2019 Geographic information — Spatial schema
ISO 19108:2002 Geographic information — Temporal schema
ISO 19109:2015 Geographic information — Rules for application schema
ISO 19111:2019 Geographic information — Referencing by coordinates

3 Terms, definitions, abbreviated terms and notation

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:

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

3.1.1 analytical coverage
type of continuous coverage which is a spatially bounded, but transfinite set of direct positions, and a mathematical function that relates direct position to feature attribute value
3.1.2
axis
(coordinate geometry) linear feature from which a one-dimensional coordinate system is constructed

Note 1 to entry: This definition is established in accordance with ISO 19111:2019 clause 10.4

3.1.3
continuous coverage
coverage that returns different values for the same feature attribute at different direct positions within a single spatial object, temporal object or spatio-temporal object in its domain

Note 1 to entry: Although the domain of a continuous coverage is ordinarily bounded in terms of its spatial and/or temporal extent, it can be subdivided into an infinite number of direct positions.

3.1.4
coordinate
one of a sequence of numbers designating the position of a point

Note 1 to entry: A direct position is described by an ordered sequence of coordinates. The number of elements in a direct position is established by the number of axes of the coverage.

[source: iso 19111:2019, 3.1.5, modified — original note 1 to entry has been replaced with a new note to entry.]

3.1.5
coordinate dimension
<coordinate geometry> number of measurements separate decisions needed to describe a position in a coordinate system

Note 1 to entry: The number of separate decisions corresponds to the number of axes.

[source: iso 19107:2019, 3.17, modified — original note 1 to entry has been replaced with a new note to entry.]

3.1.6
coordinate system
set of mathematical rules for specifying how coordinates are to be assigned to points

[source: iso 19111:2019, 3.1.11]

3.1.7
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.

[source: iso 19111:2019, 3.1.9]
3.1.8 coverage
feature that acts as a function to return values from its range for any direct position within its domain

3.1.9 coverage CRS
the common CRS in which all coordinates of a coverage are expressed

Note 1 to entry: Sometimes a coverage’s CRS is also referred to as the coverage’s Native CRS.

3.1.10 coverage dimension
coordinate dimension
<coordinate geometry> number of separate decisions needed to describe a position in a coordinate system

Note 1 to entry: This is equivalent to “the number of axes in the coordinate reference system of the coverage domain set.

[SOURCE: ISO 19107:2019, 3.17, modified — Original Note 1 to entry has been replaced with a new note to entry.]

3.1.11 coverage geometry
configuration of the domain of a coverage described in terms of coordinates

3.1.12 Delaunay triangulation
network of triangles such that the circle passing through the vertices of any triangle does not contain, in its interior, the vertex of any other triangle

3.1.13 direct position
<geographic information> position described by a single set of coordinates within a coordinate reference system

Note 1 to entry: Cells in a grid coverage are identified by their direct position in the domain set of this coverage.

[SOURCE: ISO 19136-1:2020, 3.1.20, modified — Note 1 to entry has been added.]

3.1.14 cell
(coverage) neighbourhood around a direct position in a coverage grid

Note 1 to entry: Coverage cells are also known as grid cells.

3.1.15 discrete coverage
coverage that returns the same feature attribute values for every direct position within any object in its domain

Note 1 to entry: The domain of a discrete coverage consists of a finite set of spatial, temporal, or spatio-temporal objects.

Note 2 to entry: Discrete coverages have values only where they are defined, whereas continuous coverages can be interpolated thereby providing intermediate values.
3.1.16  
**domain**  
well-defined set

Note 1 to entry: All elements within a domain (set) are of a given type

[SOURCE: ISO 19109:2015, 4.8, modified — Original Note 1 to entry has been replaced with a new note to entry.]

3.1.17  
**external coordinate reference system**  
coordinate reference system whose datum is independent of the object that is located by it


3.1.18  
**evaluation**  
〈coverage〉 determination of the values of a coverage at a direct position within the domain of the coverage

3.1.19  
**feature**  
abstraction of real world phenomena

[SOURCE: ISO 19101-1:2014, 4.1.11, modified — Note 1 to entry has been removed.]

3.1.20  
**feature attribute**  
characteristic of a feature

Note 1 to entry: Also known as “feature property” and may support potential attribute, quality, or characteristic of a feature.

[SOURCE: ISO 19101-1:2014, 4.1.12, modified — Original Notes to entry have been deleted and a new Note 1 to entry added.]

3.1.21  
**function**  
<mathematics, programming> rule that associates each element from a domain (“source domain”, or “domain” of the function) to a unique element in another domain (“target domain”, “co-domain” or “range” of the function)

[SOURCE: ISO 19107:2019, 3.41]

3.1.22  
**geometric dimension**  
〈geometry, topology〉 largest number n such that each point in a set of points can be associated with a subset that has that point in its interior and is topologically isomorphic to $\mathbb{E}^n$, Euclidean n-space

[SOURCE: ISO 19107:2019, 3.48 modified — Original Notes to entry have been deleted.]

3.1.23  
**geometric object**  
〈geometry〉 spatial object representing a geometric set

Note 1 to entry: A geometric object consists of a geometric primitive, a collection of geometric primitive, or a geometric complex treated as a single entity. A geometric object may be the spatial representation of a feature object.

3.1.24
**geometric set**
<geometry> set of direct positions

[SOURCE: ISO 19136-1:2020, 3.1.32, modified — Original Note to entry has been deleted.]

3.1.25
**georectified**
corrected for positional displacement with respect to the surface of the earth.


3.1.26
**georeferenceable**
associated with a geopositioning information that can be used to convert grid coordinate values to values of coordinates referenced to an external coordinate reference system related to the Earth by a datum.

[SOURCE: ISO 19163-1:2016]

3.1.27
**georeferencing**
geopositioning an object using a Correspondence Model derived from a set of points for which both ground and image coordinates are known.

[SOURCE: ISO 19130-1:2018, 3.37]

3.1.28
**grid**
<coverage> nonempty, ordered set of axes with a set of positions along each axis acting as reference points for connected compact smooth hypersurfaces

Note 1 to entry: In 19123:2005 a grid consists of a network composed of one or more sets of curves in which the members of each set intersect the members of the other sets. This definition is intended to be applicable also to the 1-D case.

Note 2 to entry: The 19123:2005 definition is equivalent to the revised definition of this document.

3.1.29
**grid coordinate reference system**
coordinate reference system for the positions in a grid that uses a defined coordinate system congruent with the coordinate system described by the GridEnvelope and axisLabels of gml:GridType

Note 1 to entry: The grid’s CRS is identical to the CRS of the coverage defined by that grid.

[SOURCE: ISO 19136-2:2015, 4.2.1, modified — Original Note 1 to entry has been deleted and a new Note 1 to entry added.]

3.1.30
**grid coordinate system**
coordinate system in which a position is specified relative to the intersection of curves


3.1.31
**grid coordinates**
sequence of two or more numbers specifying a position with respect to its location on a grid

3.1.32
grid point
point located at the intersection of two or more curves in a grid

3.1.33
grided data
data whose attribute values are associated with positions on a grid coordinate system.

Note 1 to entry: Gridded data are a subtype of coverage data, which represent attribute values of geographic features in terms of a spatial grid.

[SOURCE: ISO 19115-2:2019, 3.16, modified — Note 1 to entry has been added.]

3.1.34
image coordinate system
Image CS
two-dimensional non-georeferenced Cartesian grid coordinate system

Note 1 to entry: An Image CS is a two-dimensional Index CS, hence a special case of an Index CS.

3.1.35
image coordinate reference system
Image CRS
Cartesian coordinate reference system applicable to an image

Note 1 to entry: The CRS of a raster image (without georeferencing) is a two-dimensional grid where the axes are Cartesian axes (i.e., based on an Index CRS); this is commonly referred to as Image CRS.

3.1.36
image coordinates
coordinates with respect to a Cartesian coordinate system of an image

Note 1 to entry: Gridded data are a subtype of coverage data, which represent attribute values of geographic features in terms of a spatial grid.

Note 2 to entry: The image coordinates can be in pixels or in a measure of length (linear measure).

[SOURCE: ISO 19130-2:2014, 4.33, modified — A new Note 2 to entry has been added.]

3.1.37
image datum
datum describing the relationship of an image coordinate system to the origin of a Cartesian coordinate system

3.1.38
index coordinate system
Index CS
multi-dimensional non-georeferenced Cartesian grid coordinate system

3.1.39
index coordinate reference system
Index CRS
multi-dimensional non-georeferenced Cartesian grid coordinates reference system

3.1.40
index datum
datum which describes the relationship of an index coordinate system to the origin of a Cartesian coordinate system
3.1.41 Interface

<UML> classifier that represents a declaration of a set of coherent public UML features and obligations

Note 1 to entry: An interface specifies a contract; any classifier that realizes the interface must fulfill that contract. The obligations that can be associated with an interface are in the form of various kinds of constraints (such as pre- and post-conditions) or protocol specifications, which can impose ordering restrictions on interactions through the interface.

[SOURCE: ISO 19103:2015, 4.21]

3.1.42 Mesh

geometry with associated topology of dimension greater than zero

Note 1 to entry: Geometry and topology are defined in ISO 19107. Mesh examples include curves, TINs, and solids. Points (and point clouds) resemble geometries with dimension zero.

3.1.43 Pixel

smallest element of a digital image to which attributes are assigned

Note 1 to entry: A pixel is the smallest unit of display for a visible image.

Note 2 to entry: This term originated as a contraction of "picture element"

[SOURCE: ISO 19101-2:2018, 3.28, modified — Note 1 to entry has been moved to Note 2 to entry and a new Note 1 to entry has been added.]

3.1.44 Point cloud

collection of data points in 3D space

Note 1 to entry: The distance between points is generally non-uniform and hence all three coordinates (Cartesian or spherical) for each point must be specifically encoded.


3.1.45 Point coverage

coverage that has a domain composed of points

3.1.46 Polygon coverage

coverage that has a domain composed of polygons

3.1.47 Range

(set of feature attribute values associated by a function, the coverage, with the elements of the domain of a coverage)

Note 1 to entry: This is consistent with the more generic definition of range in ISO 19107:2019.

3.1.48 Raster

rectangular pattern of parallel scanning lines forming a grid

Note 1 to entry: The term is also used as an imprecise generic term for all imagery and gridded coverage data.
Note 2 to entry: Historically, the term derives from the display pattern on a cathode ray tube.

3.1.48\textbf{rectified grid}
grid for which there is an affine transformation between the grid coordinates and the coordinates of an external coordinate reference system

Note 1 to entry: If the coordinate reference system is related to the earth by a datum, the grid is a georectified grid.

3.1.50\textbf{referenceable grid}
grid associated with a transformation that can be used to convert grid coordinate values to values of coordinates referenced to an external coordinate reference system

Note 1 to entry: If the coordinate reference system is related to the Earth by a datum, the grid is a georeferenceable grid.

3.1.51\textbf{solid}
three-dimensional geometric primitive, representing the continuous image of a region of Euclidean 3-space

Note 1 to entry: A solid is realizable locally as a three-parameter set of direct positions. The boundary of a solid is the set of oriented, closed surfaces that comprise the limits of the solid.

[SOURCE: ISO 19107:2019]

3.1.52\textbf{spatial object}
\langle\text{topology, geometry}\rangle\ object used for representing a spatial characteristic of a feature

[SOURCE: ISO 19107:2019, 3.87]

3.1.53\textbf{spatial reference system}
\text{system for identifying position in the real world}

[SOURCE: ISO 19155:2012, 4.20]

3.1.54\textbf{spatio-temporal domain}
\langle\text{coverage}\rangle\ domain composed of spatio-temporal objects

Note 1 to entry: The spatio-temporal domain of a continuous coverage consists of a set of direct positions defined in relation to a collection of spatio-temporal objects.

3.1.55\textbf{spatio-temporal object}
object representing a set of direct positions in space and time

3.1.56\textbf{tessellation}
partitioning of a space into a set of conterminous subspaces having the same dimension as the space being partitioned

3.1.57\textbf{Thiessen polygon}
polygon that encloses one of a set of points on a plane so as to include all direct positions that are closer to that point than to any other point in the set
3.1.58 triangulated irregular network
TIN
tessellation composed of triangles

3.1.59 vector
quantity having direction as well as magnitude

Note 1 to entry: A directed line segment represents a vector if the length and direction of the line segment are equal to the magnitude and direction of the vector. The term vector data refers to data that represents the spatial configuration of features as a set of directed line segments.

3.2 Abbreviated terms

1D one-dimensional
2D two-dimensional
3D three-dimensional
4D four-dimensional
CRS Coordinate Reference System
DGGS Discrete Global Grid System
GIS Geographic Information System
TIN Triangulated Irregular Network
UML Unified Modelling Language

3.3 Notation

In this document, conceptual schemas are presented in the Unified Modeling Language (UML). ISO 19103 Conceptual schema language presents the specific profile of UML used in this document.

Several model elements used in this schema are defined in other International Standards developed by ISO/TC 211. UML classes defined in this document have the two-letter prefix of CV. Table 1 lists the other standards and packages in which UML classes used in this document have been defined.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>International Standard</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX</td>
<td>ISO 19115-1</td>
<td>Extent</td>
</tr>
<tr>
<td>GF</td>
<td>ISO 19109</td>
<td>General Feature Model</td>
</tr>
<tr>
<td>GM</td>
<td>ISO 19107</td>
<td>Geometry</td>
</tr>
<tr>
<td>SC</td>
<td>ISO 19111</td>
<td>Spatial Coordinates</td>
</tr>
<tr>
<td>TM</td>
<td>ISO 19108</td>
<td>Temporal Schema</td>
</tr>
</tbody>
</table>

PROCEDURAL Note: There has been a proposal to change the reference to temporal coordinates from ISO 19108 to ISO 19111. This decision is left to the Working Group in consultation with the 19111 Project Leader.
4 Conformance

4.1 Interoperability and Conformance Testing

This document being an abstract standard allows for multiple different implementations and does not define a standardized interoperable implementation. The abstract concepts described herein can be implemented in a variety of ways which may not be directly interoperable, that is: the same abstract coverage represented through two different implementation models will not necessarily be identical in their structure, and services following two different implementation models will not necessarily deliver equivalent results on equivalent queries or other operations. The purpose of the abstract description standardized in this document is to provide an underlying consistency at the data model level that makes it possible to establish concretizing, interoperable standards.

Conformance testing is accomplished by manually validating a candidate concretization against all requirements by exercising the tests set out in Annex A.

In an implementation standard based on this abstract specification the semantics defined in this document will normally be cast into a concrete data model (describing data structures to be stored, transferred, ingested, or extracted) and a concrete service model (describing the functionality of a service operating on coverages); such derived models should be designed in an interoperable manner, i.e.: allow concise conformance testing.

This document has a companion standard, ISO 19123-2 Coverage Implementation Schema (CIS). Based on this ISO 19123-1 standard, ISO 19123-2 defines a concrete coverage model in the sense that interoperability can be guaranteed and interoperability tests (such as the OGC compliance tests on coverages [16]) can be established.

In addition to the main body, Annex D defines a Generic Coverage Data Structure. This structure represents one possible data structure that is compatible with the interface defined in this standard. This is the structure that was defined in the previous version of ISO 19123. The purpose of Annex D is to retain backward compatibility because there exist other standards both in ISO and in external organizations that make direct reference to the generic data classes that were defined in the previous version of ISO 19123 and which now may reference the same classes as defined in this Annex. The interface approach is more flexible, but the classes defined in Annex D are one valid structure that may be supported through the interface.

4.2 Organization

The coverage schema is organized into the packages shown in Figure 1, Figure 2 and Table 2. Each package establishes one requirements class. Grouping into these requirements classes has been done with a practical perspective in mind: Realizations of this document may be, focused on particular structures, such as grid coverages or point clouds, while ignoring all the other options.

<table>
<thead>
<tr>
<th>Conformance class</th>
<th>Clause</th>
<th>Identifying URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage (Core)</td>
<td>5</td>
<td><a href="https://standards.isotc211.org/19123/-1/1/conf/core">https://standards.isotc211.org/19123/-1/1/conf/core</a></td>
</tr>
<tr>
<td>Multi-Point Coverage</td>
<td>6</td>
<td><a href="https://standards.isotc211.org/19123/-1/1/conf/multi-point">https://standards.isotc211.org/19123/-1/1/conf/multi-point</a></td>
</tr>
<tr>
<td>Grid Coverage</td>
<td>7</td>
<td><a href="https://standards.isotc211.org/19123/-1/1/conf/grid">https://standards.isotc211.org/19123/-1/1/conf/grid</a></td>
</tr>
<tr>
<td>Multi-Curve Coverage</td>
<td>8</td>
<td><a href="https://standards.isotc211.org/19123/-1/1/conf/multi-curve">https://standards.isotc211.org/19123/-1/1/conf/multi-curve</a></td>
</tr>
<tr>
<td>Multi-Surface Coverage</td>
<td>9</td>
<td><a href="https://standards.isotc211.org/19123/-1/1/conf/multi-surface">https://standards.isotc211.org/19123/-1/1/conf/multi-surface</a></td>
</tr>
<tr>
<td>Multi-Solid Coverage</td>
<td>10</td>
<td><a href="https://standards.isotc211.org/19123/-1/1/conf/multi-solid">https://standards.isotc211.org/19123/-1/1/conf/multi-solid</a></td>
</tr>
</tbody>
</table>
ISO 19505 Unified Modeling Language (OMG UML)

**Figure 1 — Coverage Modelling Package Hierarchy**

**Figure 2 — Packages of the coverage schema**

ISO 19123-1:20xx(E)
5 Coverages

5.1 Overview

This Clause defines conformance class Core of package CV_Coverage.

This Document uses the term coverage, adopted from the Abstract Specification of the Open GIS Consortium [19], to refer to any data representation that assigns values directly to spatial and/or temporal positions. As such, a coverage conceptually can be viewed as a function which, for every value of its domain set, provides a particular value taken from its range set; actually, a coverage may provide a set of values for a particular position.

**EXAMPLE** Point clouds can be modelled as Multi-Point Coverages. Several observation values can be acquired for a particular location, and reading such a location will result in the set of all values observed.

Coverages are multi-dimensional by nature – such as 1-D sensor timeseries, 2-D x/y images, 3-D x/y/t image timeseries and x/y/z geophysical information, and 4-D x/y/z/t atmospheric and oceanographic information. The dimension axes spanning the coverage’s can be of spatial, temporal, or “abstract” nature (where abstract is understood in the sense of being neither spatial nor temporal), or any combination of these. As such, a coverage forms a digital representation of some space/time varying phenomenon.

**EXAMPLE** A satellite image timeseries has two spatial and one temporal axis. A geo-statistical datacube may have a temporal axis, spatial extents, and population density.

The set of locations (in a wide, spatio-temporal sense) where a coverage has objects sitting and, hence, has values to offer is called the coverage’s Domain Set, said locations are called Direct Positions. The set of all these values is the coverage’s Range Set, described by the Range Type.

Coordinates in a coverage are all expressed in one and the same Coordinate Reference System (CRS). Such a CRS may either be defined directly (such as in the EPSG collection of CRSs [6]) or it may be composed from CRS and axis definitions through the mechanisms defined in ISO 19111.

**EXAMPLE** The CRS of a satellite image timeseries consists of two spatial and one temporal axis, in some given order. Coordinates along each axis are expressed accordingly – horizontal spatial coordinates may be expressed in degrees Lat and Long, laid down in the EPSG:4326 CRS, while time is expressed in seconds since epoch or some calendar, like Gregorian-Proleptic.

In terms of common data structures coverages encompass regular and irregular grids (requirements class Grid Coverage), point clouds (requirements class Multi-Point Coverage), and general meshes (requirements classes Multi-Curve Coverage, Multi-Surface Coverage, and Multi-Solid Coverage).

Different views on this coverage concept exist. Therefore, several practically relevant views are explicitly supported:

- In a mathematical view a coverage is defined as a function $C: D \rightarrow R$ with Domain Set $D$ and Range Set $R$ which delivers some value for each element from $D$. This view is realized as one variant of coverage modelling in subclass CV_CoverageByDomainAndRange (see Subclause 5.8.2).

- In standardization terms, a coverage can be described as a set of features (as per ISO 19101-2). As each such feature has a location and attributes, the set of all these pairs defines the coverage’s mapping from location to values. “Geometry” here is understood in the widest possible sense, including all spatial and temporal dimensions. All features in such a collection forming a coverage must have locations expressed in the same CRS (the coverage’s CRS), and all attribute values must share the same data type (the coverage’s Range Type). See Subclause 5.8 for details.
5.2 Coverage Schema

This Clause describes package CV_Coverage in terms of interfaces (as per ISO 19103) of its central class, CV_Coverage. It defines a set of requirements for compliance with this interface description, based on the concepts of UML modelling from ISO 19103:2015, Coordinates and Collections from ISO 19107, time from ISO 19108, and Coordinate Systems from ISO 19111.

Requirement 1:
An instantiation of package CV_Coverage shall have all instances and properties specified for this package, its contents, and its dependencies for CV_Coverage as per Figure 2 and Figure 3.

![Figure 2: CV_Coverage central classes](image)

The UML interface modelling approach allows multiple different compatible data structures to be implemented which exhibit the same behaviour through the interface. Standardization targets are specifications concretizing the abstract concepts into implementation standards; one such example is 19123-2 which in turn as standardization target has concrete implementations.

Standardization targets are specifications concretizing the abstract concepts into implementation standards; one such example is 19123-2 which in turn as standardization target has concrete implementations.

Requirement 2:
An instantiation of package CV_Coverage shall make use of ISO 19107 package Coordinates.

5.3 Probing Coverages: evaluate() Function

One way to define the semantics of a coverage is via a probing function which, for some Direct Position coordinate expressed in the coverage’s CRS, returns the set of values associated with it. This function can be defined, for a coverage $C$ with Domain Set $D$ and value set $V$ of that coverage, as

$$evaluate_C: D \rightarrow V, \text{evaluate}_C(p) = \bigcup_{f \in C} f\text{.contains}(p)$$

based on the contains() probing predicate of ISO 19107.

Note 1 This probing function serves for definition purposes only, it is not required to be implemented. In practice, other retrieval functionality is desirable, such as bounding box subsetting in the OGC Web Coverage Service (WCS) Core [15].

While in general more than one value can be returned for a particular Direct Position, sometimes exactly one value will be delivered. This can occur in two cases:

— In case of a Grid Coverage, no two Direct Positions share the same coordinates by definition, and so there is always one value available via evaluate().

— If a Common Point Rule (see 5.7) is defined then this constitutes a selection mechanism ensuring that exactly one member of the set of values available for a given Direct Position is returned by evaluate().
EXAMPLE More than one values per Direct Position can occur in several cases, such as in point clouds with two points incidentally sharing the same coordinates, but bearing different values, or in MultiSolid Coverages that overlap.

5.4 Domain of a coverage

5.4.1 Concept

The coverage domain set describes for which positions in the coverage’s multi-dimensional space values are available. Within this multi-dimensional space defined by the Domain’s Coordinate Reference System (CRS) the coverage Domain contains a set of geometric objects which determines the Direct Positions, i.e.: the locations in this space where the coverage offers a value. This description can be given through direct enumeration of the Direct Positions (example: point clouds) or through containment descriptions (example: areas and volumes), or some other mechanism (example: Ground Control Points in sensor models). The coverage’s Extent gives a bounding box – i.e.: lower and upper bounds along every coordinate axis – within which all its Direct Positions are located.

EXAMPLE The space spanned in coverages representing 1-D temperature measurement timeseries, 2-D x/y images of the Earth surface, 3-D x/y/t image timeseries and x/y/z subsurface voxel data, and 4-D x/y/z/t atmospheric and ocean data, all are described through some appropriate CRS with the respective dimension.

The geometric objects in a coverage domain are not strictly confined through 3-dimensional physical space. They can be m-dimensional objects where m≤n for an n-dimensional coverage. Coverage subtypes are defined in terms of their domain types in Clause 6 onwards.

EXAMPLE A 3-dimensional x/y/t image timeseries datacube may be composed of 0-dimensional points, its pixels. A domain of coordinate dimension 3 may be composed of points, curves, surfaces, or solids, while a domain of coordinate dimension 2 may be composed only of points, curves or surfaces.

5.4.2 Coordinates

ISO 19111-2 defines CRSs which contain ordered lists of axes used for addressing points in space. For an n-dimensional coverage, n>0, the corresponding CRS contains an ordered set of n axes whose syntax and semantics (such as unit of measure, discrete or continuous) is part of the CRS definition. Points in the coverage domain’s n-dimensional space are addressed through n-tuples of coordinates based on the n-dimensional coverage’s CRS.

ISO 19107 defines geometric objects (subtypes of class GM_Object), ISO 19108 defines temporal objects (class TM_GeometricPrimitives) that may be used to define domains of coverages.

The Domain Set of a coverage, as described by its Coordinate Reference System (CRS) consists of a number of axes which together define some n-dimensional space, with n>0.

Often the unit of measure of an axis is numeric, such as degrees or meters or seconds. However, there may be several representations of coordinate measurements, such as degree/minute/seconds or Gregorian Proleptic time/date strings. Finally, axes without any spatial nor temporal semantics, might be added (such as geo statistics measures) which, for the purpose of this standard, are subsumed informally as “abstract”. Therefore, CV_DirectPosition defining coordinates for CV_Coverage takes a broader approach to coordinate values than ISO 19107 class DirectPosition, generalizing spatial, temporal, and other coordinate types into one class. Further, as the type now is Any instead of numeric any syntax becomes possible for expressing coordinates. In Figure 4, on the left the ISO 19107 style of separated coordinate types is shown while bottom right the integrated, simplified definition of CV_DirectPosition is shown.
EXAMPLE  The following is a non-exhaustive list of possible axes in coverage Domain Sets:

- A CRS consisting of a single time axis can model a timeseries of measurements. Time coordinates can be expressed, among others, in seconds since epoch (often: January 1, 1970) or as time/date strings of the Gregorian Proleptic Calendar defined in ISO 8601, such as “2021-05-12”.
- Pressure altitudes, measured in hekto-Pascal (hPa), as well as Flight Levels, expressed in 100 ft steps like “FL150” for 15000 ft above some reference point, represent “proxies” for altitude used in aviation.
- Integer coordinates can be used to define a coverage axis of spectral frequencies.

5.4.3 Coordinate Reference Systems for Coverage Domains

The coverage’s CRS defines the theoretically available space for Direct Positions, bounded additionally by the coverages extent in that space. Generally, though, not for every position in this space a Direct Position may be defined. The type of feature bundled in a coverage determines where Direct Positions, expressed in coordinates based on the coverage’s CRS, are available. As feature types commonly are sorted along their topological dimensions – thereby defining points, lines, surfaces, and solids in natural space – this is an appropriate criterion for classifying coverage types. Additionally, there may hold further constraints, such as with grid coverages which are based points having topological dimension 0, but additionally require the Direct Positions to sit on some grid. In the following Clauses, coverage types sorted along their topological dimension as defined in ISO 19107.

ISO 19111-2 describes ways to compose CRSs for building higher-dimensional CRSs from lower-dimensional ones.

Requirement 3:
The coverage CRS shall be described in accordance with ISO 19111 which describes the semantics of CRSs and their axes, as well as their composition into higher-dimensional CRSs.
Requirement 4:
A coverage CRS shall be identified by an ISO 19107 DirectPosition::RSID.

Requirement 5:
In a coverage CRS, coordinates with a spatial semantics shall be represented by an ISO 19107 DirectPosition.

Requirement 6:
In a coverage CRS, coordinates with a temporal semantics shall be represented by an ISO 19108 TM_Position.

Axes without any further semantics may make use of a Cartesian CRS (IndexCRS, Figure 5), as defined by OGC\(^2\), which is a family of CRSs where each axis is defined over unit-less integer numbers. IndexCRS1D defines a single-axis CRS, Index2D a 2D CRS, and so on. By combining such IndexCRSs with spatial and temporal CRSs a wide range of multi-dimensional domain sets can be described.

The concept of an Image Coordinate Reference System (Image CRS, Figure 5) has been introduced with the purpose of unifying the handling of non-georeferenced imagery and referenced objects. With this Image CRS there is always a CRS for coverages, even if they are not related to any position in space/time. Technically, an Image CRS can be defined as subclass of ISO 19111 Coordinate Reference System consisting of an Image CS and an Image Datum. It is characterized by being 2D and having Cartesian integer coordinates on both axes. As such, an Image CRS is identical to the OGC Index2D CRS.

\[\text{Figure 5 — IndexCRS and ImageCRS}\]

\(^2\) As served, for example, through http://secure.rasdaman.org/def/crs/OGC/0/Index2D which is under the control of OGC
EXAMPLE 1   A 4D domain set may be described through axes Lat and Long (given by EPSG:4326) combined with a time axis as per ISO 8601 combined with an IndexCRS2D for spectral bands.

EXAMPLE 2   The Direct Positions in a coverage’s Domain Set can be enumerated (example: point sets in a Multi-Point Coverage), they can be given implicitly (example: corner points indicating the set of all raster positions in a Rectified Grid Coverage).

The range of CRSs to be used is open-ended; CRSs not described nor mentioned in this standard might be used in a coverage. Recommendation, though, is to use CRSs from some standard repository, such as the OGC spatial and temporal CRS repository, or following some standardized syntax, such as OGC WKT, or some geo referenced grid system like DGGS which uses a compound tessellation by fitting a polyhedron such as a 20-sided icosahedron or 6-sided cube to a sphere or ellipsoid and then tessellating each facet.

5.4.4 Multi-Dimensional Coverages

A single axis, being component of a one- or higher-dimensional CRS, can be established in various ways. When composing axes, as per ISO 19111, both CRSs and single axes can be combined into some higher-dimensional CRS. Mechanisms to do so are out of scope of this document, they are defined in ISO 19111.

The Domain Set of a coverage, as described by its Coordinate Reference System (CRS) consists of a number of axes which together define some \( n \)-dimensional space, with \( n > 0 \). While this document does not define Domain Set semantics (this is done in ISO 19111) it can be stated that physical dimensions consist of two horizontal axes (such as \( x \) and \( y \)), a vertical axis, and a temporal axis. By combining one or more of these axes spatio-temporal objects can be built. Further, axes can represent alternate representations of coordinate measurements. Finally, other possible axes, without any spatial nor temporal semantics, might be called “abstract”.

Example The following is a non-exhaustive list of possible axes in coverage Domain Sets:

- A CRS consisting of a single time axis represents a timeseries of values.
- Pressure altitudes, measured in hekto-Pascal (hPa), as well as Flight Levels, expressed in 100 ft steps like “FL150” for 15000 ft above Mean Sea Level, represent “proxies” for altitude used in aviation.
- Spectral frequencies can define a coverage axis.

The CRS defines the theoretically available space for Direct Positions. Generally, though, not for every position in this space a Direct Position may be defined. The type of feature bundled in a coverage determines where Direct Positions, expressed in coordinates based on the coverage’s CRS, are available. As feature types commonly are sorted along their topological dimensions – thereby defining points, lines, surfaces, and solids in natural space – this is an appropriate criterion for classifying coverage types. Additionally, there may hold further constraints, such as with grid coverages which are based points having topological dimension 0, but additionally require the Direct Positions to sit on some grid. In the following Clauses, coverage types sorted along their topological dimension as defined in ISO 19107.

5.4.5 Mathematical vs Physical Coordinates

The Domain Set of a coverage, as described by its Coordinate Reference System (CRS) consists of a number of axes which together define some \( n \)-dimensional space, with \( n > 0 \). Geographic data typically have a subset of two horizontal axes, one height axis (expressing elevation or bathymetry), and time. Climate modeling adds a second time axis for differentiating model run time from time modelled. By combining one or more of these axes multi-dimensional spatio-temporal objects can be built. Further, axes can represent alternate representations of coordinate measurements. Additionally, “abstract” (in the sense of non-spatio-temporal)

3 http://www.opengis.net/def/crs
4 https://www.ogc.org/standards/wkt-crs
axes may occur as well, like in Online Analytical Processing (OLAP) where, e.g., time / product / subsidiary axes are common.

EXAMPLE Non-spatio-temporal axes occur in practice as well. For example, bands in hyperspectral imagery make sense as a numbered sequence once there are hundreds of such bands, such as the Hyperion instrument on board of EO-1 with its 220 bands\(^5\). With natural numbers addressing becomes less wieldy.

Note 2 Originally spatial dimensions might become non-spatial at some level of generalization. For example, cities in Europe might be expressed through coordinates originally, but at some higher level get abstracted to be in Bavaria, in the Alps region, etc. which is at a symbolic level. This may lead to dimension hierarchies like in OLAP.

5.5 Range of a coverage

At every direct position a coverage holds a single value or a nonempty set of values. All these values of a coverage together make up the coverage's range set.

EXAMPLE A coverage might assign to each direct position in a county the temperature, pressure, humidity, and wind velocity and direction vector, at a specific time, at that point. The coverage maps every direct position in the county to a record of these fields. In this case, the common type of the coverage range values is a record of components, each of its individual type.

The \textit{Range Type} component describes the type of the set of values available for the Range Set. Structures such described can be atomic or composite.

Requirement 7: The \textit{Range Type} component shall be of type \textit{Record Type} as defined in ISO 19103.

Requirement 8: The \textit{Range Type} of a coverage at minimum shall provide the information necessary to decode the Range Set values and process them in a computer system, consisting of (i) data type information in some available typing system, (ii) null values, and (iii) unit of measure.

EXAMPLE RGB images, when modelled as a coverage, have as their range type a record consisting of three components \textit{red}, \textit{green}, and \textit{blue} (in that order), each of them of type unsigned 8-bit integer.

Recommendation 1: Range Type definitions should rely on some commonly agreed system of measurement.

Note 3 One way of providing this information is via a HTTP-URI pointing to a location that provides this information in a standardized manner. Another way is to provide this information directly as part of the coverage.

5.6 Interpolation

5.6.1 Concept

Through interpolation, range values can be obtained for coordinates within the domain set of a coverage which are not Direct Positions. Basically, a coverage provides values only at its Direct Positions. Interpolation means applying some algorithm to obtain a range value for locations inside a coverage’s domain set which are not Direct Positions, usually by combining the values of several Direct Positions in the neighbourhood of the coordinate location under inspection.

Interpolation requires domain set coordinates to be in a CRS which allows for expressing “inbetween” values.

\[^5\text{https://archive.usgs.gov/archive/sites/\text{eo1.usgs.gov/hyperion.html}}\]
EXAMPLE Index coordinates, representing integer numbers, do not allow expressing any inbetween value as it would not be an integer any more. Latitude, Longitude, height, and time, conversely allow expressing values between any two given coordinates as they conceptually map to real numbers.

Depending on the interpolation method, i.e.: the algorithm applied, the range type may require providing “in between” values, too. In general, for interpolation it may be necessary that the range type allows for arbitrary such values to be generated, requiring a continuous range type. Therefore, applicability of a particular interpolation method in general also depends on the range type.

EXAMPLE “Nearest neighbour” interpolation does not require new values to be computed as the resulting value is chosen from the available ones taken from the surrounding Direct Positions. “Linear interpolation”, on the other hand, usually will generate values different from the surrounding ones, based on the arithmetic means computation.

Interpolation can be applied along a single axis or along several axes simultaneously. Therefore, interpolation methods on high, conceptual level are defined on axis level. A concretization of this standard may bundle interpolation methods to allow only one and the same interpolation method across all (or a subset of) axes.

EXAMPLE In horizontal geographic CRSs interpolation will normally be coupled so that, say, Latitude and Longitude always share the same interpolation. Time and height, on the other hand, will normally be interpolated individually.

### 5.6.2 Discrete and Continuous Coverages

The domain set characteristics decide in the first place whether interpolation is applicable. The CRS, which defines the set of possible coordinate values, may or may not allow addressing of coordinate values beyond the Direct Positions. Hence, coverages are possible which allow interpolation along one axis, but not along another.

EXAMPLE A grid coverage with Lat, Long, and an index axis can be interpolated in Lat and Long, but not along the index axis.

An axis is called **discrete** if every possible interval with finite bounds describes a finite set of values, otherwise such an axis is called **continuous**.

A coverage is called **discrete** if its axis list contains only discrete axes. A coverage is called **continuous** if its axis list contains at least one continuous axis.

Coverages can be discrete (or continuous) in their range, in their domain, or in both.

EXAMPLE 1 A map of postal code zones is a coverage which is discrete in its range. The postal code zones cover an entire country and at every location in the country one can evaluate the coverage function and get a value that represents the postal code for that location. Within a postal code zone the value is constant. One cannot interpolate such a discrete coverage.

EXAMPLE 2 A coverage that maps a set of polygons to the soil type found within each polygon is a coverage which is discrete in its range. More examples of this type of coverage are given in ISO 19144-2 Classification Systems – Part 2 – Land Cover Meta Language (LCML).

EXAMPLE 3 A point set representing a set of measurements that are only valid at the position of each point, and which cannot be interpolated, is a discrete coverage (discrete in domain).

EXAMPLE 4 An image, sampled by a sensor, may be represented as a grid coverage consisting of a set of pixels corresponding to grid cells in the domain of the coverage. A value is associated with each grid cell. However, since the coverage is continuous an interpolation function – such as linear, quadratic, or cubic – may be applied so that a continuously variable value may be determined at any location within the domain extent of the image.

EXAMPLE 5 In a coverage that maps direct positions in San Diego County to their temperature at noon on a specific day, both domain and range may take an infinite number of different values. This continuous coverage would be associated with a discrete coverage that holds the temperature values observed at a set of weather stations (discrete in domain). That is, the measured values correspond to a point set coverage. This point set coverage is discrete because
each point can only have one value. The associated continuous coverage uses the point values as driving values for the coverage function and allows interpolation between the points.

EXAMPLE 6 A set of bathymetric soundings is a discrete point set coverage with a single measured water depth value at each point location (discrete in domain). An associated continuous coverage allows one to interpolate between the measured depth soundings to determine the bottom surface of a body of water.

EXAMPLE 7 Evaluation of a triangulated irregular network involves interpolation of values within a triangle composed of three neighbouring point value pairs.

5.6.3 Interpolation Set

The Interpolation Set component contains a – possibly empty – set of interpolation methods which are applicable to the coverage on hand, thereby establishing ways to derive values for locations beyond the coverage’s direct position. If interpolation is allowed on a coverage then proper range values can be obtained between direct positions by applying one of the coverage’s interpolation methods.

An empty interpolation set indicates that no interpolation can be applied meaningfully. See Annex B for descriptions of specific interpolation methods. This list is open ended – additional interpolation methods, including interpolation methods for higher-dimensional coverages, may be defined in an Application Schema that makes use of this standard.

Note 4 As coverages are passive data structures this cannot be enforced. However, it is a valuable hint to any application as to what interpolation technique(s) can be applied meaningfully to the data set on hand.

ISO 19107 defines a set of interpolation methods, individually for the particular geometry types, upon which this Document relies.

Requirement 9:
Interpolation types allowed for the geometries contained in a coverage shall be given by the interpolation methods specified in ISO 19107 for the respective geometry type.

Recommendation 2:
An application may apply one interpolation method collectively along all axes of the coverage, or it may allow indicating individual interpolation methods applied to different axes of the coverage.

EXAMPLE In an image timeseries having Latitude, Longitude, and time axes several ways of interpolation are relevant:

— Linear interpolation along latitude and longitude (“bilinear interpolation”), with temporal resolution unchanged (in plain words: all existing timeslices get extracted) and, hence, no interpolation occurring along time.

— Interpolate each pixel’s history (i.e., along time) using linear interpolation, without any spatial interpolation.

— Linear interpolation along latitude, longitude, and time simultaneously (“trilinear interpolation”).

Note 5 As this standard is a data model and not a processing nor service model use of a particular interpolation method on a given coverage cannot be enforced. Rather, a coverage may provide information on the set of interpolation methods that should be applied whenever, in the course of performing some general processing, interpolation needs to be performed (such as in rescaling a coverage along one or more axes).

Note 6 Interpolation methods may require additional control parameters; these are not considered in this Document.

5.7 Common Point Rule

The optional CV_Coverage attribute commonPointRule of type CV_CommonPointRule identifies the procedure to be used for evaluating the CV_Coverage at a position for which more than one range values exist. Its behaviour is defined in Annex D.
5.8 Realization Variants

In a coverage, the domain and range sets may be organised in different ways, driven by practical considerations. Possible organisations include:

- Separate representation of domain and range sets in some serialization; if both use the same serialization scheme then corresponding location/value pairs can be identified through their position in the sequence. This is useful, for example, in image processing when the domain is not needed for the image operation on hand.

- Having an implicit description (rather than an explicit representation) of the domain. This reduces a coverage’s size and, again, allows convenient range set processing while avoiding the sometimes unwanted overhead of addressing each value one by one through its direct position.

- A set of position/value pairs. This is a natural representation, for example, for point clouds.

- Partitioning a coverage into smaller units. This is commonly known as tiling. In fact, this concept can be extended to recursively nested coverages.

- A functional, analytic description where domain and range are defined through mathematical expressions.

- Combinations of the above, where feasible.

Note 7 One interoperable concretization supporting various such organizations is given by ISO 19123-2 [17] which is identical to OGC CIS [16].

In the subclauses below some of the above coverage structures are detailed further. They are summarized in Figure 6.

5.8.1 Geometry/Value Pair View

The collection of feature objects constituting a coverage can be seen as establishing a mapping from the features’ space to the set of values associated with the features. In case these features and their values get enumerated this naturally leads to a set representation where each set element is a (geometry, value) pair. This is the view supported by the ISO 19107 Collection paradigm, captured in CV_CoverageByPartitioning containing a Partition Set consisting of geometry/value pairs (Figure 6). The constraints relevant for this coverage variant are established in 5.8.3.

However, other logical organisations of a coverage, with identical information content, can be constructed. Two of them are established below. Figure 6 gives a synoptic view of all variants discussed.

Requirement 10:
A CV_CoverageByGeometryValuePair shall be structured as described in Figure 6 — .

5.8.2 Domain/Range View

Another way of viewing a coverage is as a mapping from a set of direct positions (given by the geometry objects) to a set of values (given by the feature’s associated value payload). In this view, a coverage is defined as a function $C: D \rightarrow R$ with Domain Set $D$ and Range Set $R$ which delivers some value for each element from $D$. This variant is realized in subclass CV_CoverageByDomainAndRange (Figure 6).

Requirement 11:
A CV_CoverageByDomainAndRange shall be structured as described in Figure 6 — .
Note 8 Conceptually, the Domain Set of a coverage consists of all Direct Positions defined for this coverage. This set can be enumerated (Example: point sets in a MultiPointCoverage), they can be given implicitly (Example: corner points indicating the set of all raster positions in a RectifiedGridCoverage as per ISO 19123:2005). Hence, the domain/range view on a coverage is particularly relevant for grid coverages.

### 5.8.3 Partitioned View

The previously introduced modelling variants resemble two extreme ends of a modelling continuum: In the geometry/value pair representation, the finest possible granularity is adopted by having each single feature object with its value represented together. In the domain/range variant, conversely, the coarsest possible granularity is adopted by having one set of positions which get mapped to one set of values. In the partitioning approach, organizations in between these extremes are gathered. To this end, a coverage gets split into sub-coverages forming partitions of the original coverage. In the coverage model, this is captured by a PartitionSet containing Partitions.

For a coverage set to be aggregated into a larger coverage, some homogeneity constraints must hold:

**Requirement 12:**
A CV_CoverageByPartitioning shall be structured as described in Figure 6 — .

**Requirement 13:**
All sub-coverages in a partition shall share the same CRS and range type; their Domain Sets shall fulfill all constraints imposed on the particular coverage type of the super-coverage.

**Requirement 14:**
A coverage shall not recursively contain itself in a partitioning hierarchy.
Note 9 Having only one partition realizes the domain/range variant. Having partitions which contain only one feature object each represents the geometry/value pair variant.

Note 10 Partitioned storage organization can massively increase performance of access and processing. In technology such partitioning is also known as tiling and chunking.

5.8.4 Functional View

A coverage can be given by some mathematical descriptions. These descriptions must support the `evaluate()` function. This standard does not explicitly describe any particular mechanism for expressing such functions.

Requirement 15:
A CV_CoverageByFunction shall be structured as described in Figure 6 — .

EXAMPLE In Constructive Solid Geometry (CSG) objects are built by recursive composition of analytical primitives (such as box, sphere, cylinder, torus, etc.) through (regularized) set operations.

5.9 Envelope

For practical purposes it may be convenient for an application reading a coverage that it can quickly determine where approximately Direct Positions can be expected. Therefore, Coverage contains an optional component envelope of type Envelope of type Extent (as per ISO 19115-1) giving a simplified summary description of the coverage’s Domain Set. This envelope consists of an enclosing bounding shape which should be close to the actual coverage extent but does not have to be minimal. Further, the envelope can be expressed in any CRS which can be transformed to and from the coverage’s CRS.

EXAMPLE A satellite swath image may contain a bounding box expressed in WGS 84 (Figure 7).

Recommendation: Coverages should contain an Envelope.

Requirement 16:
The Envelope of a Coverage shall contain all or a subset of the axes of the corresponding Domain Set.

EXAMPLE In a 3D x/y/t image timeseries datacube the extent may only represent the approximated 2D footprint on the Earth surface, say, in WGS84, thus ignoring the time axis.

Requirement 17:
The Envelope of a Coverage should approximate its Domain Set as closely as possible, for all axes of the Domain Set present in Extent.

Figure 7 — Satellite image embedded in bounding box
Requirement 18:
If a Coverage contains an Envelope then this Envelope shall contain the Domain Set of this Coverage.

Requirement 19:
The CRS component of the Coverage shall reference a CRS definition conformant with ISO 19111.

The association Coordinate Reference System shall link the Coverage to the coordinate reference system to which the direct positions in the domain set are referenced. Class SC_CRS is specified in ISO 19111.

6 Multi-Point Coverages
This Clause defines conformance class package CV_MultiPointCoverage.

A MultiPointCoverage is a coverage consisting of a collection of points. To maintain a unique value per Direct Position it is mandatory that the point coordinates be disjoint.

Requirement 20:
A MultiPointCoverage shall contain only elements of the same type, which is ISO 19107 data type PointData or a subtype thereof, as described by Figure 8.

Requirement 21:
Function \texttt{evaluate}() shall be defined, for some coverage c and position p, as
\[ \texttt{evaluate}(p) = \{ v \mid \exists \text{ point feature } f \in c: f.\text{contains}(p) \} \]
where contains() is defined in ISO 19107.

Note 1  An alternative (and different) realization of MultiPointCoverage is given by the ISO 19107 data type PointCloud. MultiPointCoverage is included here for achieving a complete, coherent framework across all topological and geometric dimensions.

7 Grid Coverages
7.1 Overview
This Clause defines conformance class package CV_GridCoverage.

A Grid Coverage is a special case of a Multi-Point Coverage in that all direct positions must sit on a grid. The concept of a multi-dimensional grid is defined in Subclause 7.2.

Requirement 22:
A Grid Coverage shall be a subtype of Multi-Point Coverage.

Note 1 Although abstractly Grid Coverage is a subtype of Multi-Point Coverage, in practice implementation of both types will differ substantially. The regularity of a grid generally allows determining Direct Positions easier than in a point cloud, and often – depending on the degree of regularity of the grid – it is not even required to materialize the coordinates. This entails particularly efficient storage and processing methods.

7.2 Grids
7.2.1 Grid Definition
In a Grid Coverage, a Grid serves to determine the locations of the Direct Positions of the Domain. The Direct Positions carrying range values are aligned to specific points given by the grid definition. The grid’s CRS is identical to the CRS of the coverage defined by that grid.
Grids generally can be constructed based on triangles, rectangles, or hexagons. In the context of coverages, rectangular grids are modelled through Grid Coverages, hexagonal grids can be mapped to Grid Coverages (see Subclause 7.6), and triangular grids are modelled through meshes, i.e., Multi-Surface (Clause 9) or Multi-Solid Coverages (Clause 10). Therefore, in this standard the term “grid” is always understood as a rectangular grid.

Note 1
In 2-D, rectangular grids form tessellations based on quadrilaterals; in the n-D case these become n-gons.

In a rectangular grid, every direct position not sitting on the domain set boundary (“inner position”) has exactly two distinct neighbouring direct positions; those direct positions sitting on the domain set boundary (“boundary position”) have exactly one such neighbouring direct position. Rectangular grids in general do not have equidistant spacing between the Direct Positions. Figure 9 illustrates some cases of regular and irregular grids.

Note 2
As opposed to a mesh (in Computational Fluid Dynamics also called unstructured grid) where a vertex (i.e.: cell) can be connected with any number of neighbouring vertices a Grid has a regular structure based on some given number of neighbourhood vertices; therefore such a Grid sometimes is referred to as a structured grid, for clarity.

Mathematically, an n-dimensional Grid is a tessellation of the Grid Coverage’s Domain Set defining a set of Direct Positions through geometric rules as follows. For some n>0 let \( A = (a_1, \ldots, a_n) \) be a finite ordered set of axes where each axis \( a_i = (v_{i,1}, \ldots, v_{i,m_i}) \) is a totally ordered set of \( m_i > 0 \) values. This induces a Grid \( G = a_1 \times \cdots \times a_n \) as the cross product. \( G \) can be interpreted as a set of coordinates yielding Direct Positions, \( G = \{ (x_1, \ldots, x_n) | x_i \in a_i \text{ for } 1 \leq i \leq n \} \). In a coverage context, \( A \) is described by an n-dimensional CRS together with axis boundaries, typically given as lower and upper bounds per axis. The resulting regular and irregular grids are described in Subclause 7.2.2.2.
Obviously, when walking from one Direct Position to the next neighbour by incrementing or decrementing just one single axis coordinate $x_i$ to the next allowed value of this axis we obtain another Direct Position, except when starting from the lower or upper bound of the axis. We can therefore replace the actual axis coordinates by counts and establish a bijective mapping between an axis $a_i = \{ v_{i,1}, \ldots, v_{i,m_i} \}$ and a Cartesian axis $c_i = \{ 0, \ldots, m_i \}$. The CRS corresponding to this Cartesian grid is an Index CRS.

Note 3 Often rectangular grids get described through curve bundles, one per axis, whose intersections establish the Direct Positions; within each such bundle the curves pairwise must not touch nor intersect). In case of CV_IndexAxis, CV_REGULARAxis and CV_IRREGULARAxis a curve bundle will be a set of straight lines. Obviously this construction method can only explain grids in 2-D or higher.

Cartesian grids, also known as square grids, can be represented efficiently as arrays in programming languages, which leads to a preferred storage technique for coverages where the range set is modelled as an array based on some implementation-dependent mapping (such as row-major or column-major arrangement) of the CRS to the array’s Cartesian CRS. This allows tools like image processing to ignore the real-world coordinates and operate on the (Cartesian) array.

This grid notion can be generalized so that Direct Positions do not sit at the coordinate positions induced by the grid, but with some offset in arbitrary directions (although not beyond the neighbouring positions). Further, n-D grids can be embedded in some (n+m)-D space for some m>0. These cases are addressed in Subclause 7.2.2.3.

In the most general case, coordinates of the Direct Positions are not stored explicitly, but obtained algorithmically from some implementation-dependent parameters. This case is covered by Subclause 7.2.2.4.

**Requirement 23:**
In a Grid Coverage, the Direct Position shall be given by a rectangular Grid.

**Requirement 24:**
In a Grid Coverage, Direct Position has exactly one value (taken from the coverage’s Range Type) associated, i.e.: $| \text{evaluate}(p) | = 1$ for all Direct Positions $p$.

### 7.2.2 Grid Axis Types

#### 7.2.2.1 General

The axes of a CRS are defined in the CRS, as per ISO 19111. Hence, every axis (set) definition contains a reference to the overall CRS it is embedded in. Additionally, axes describe the subset of coordinate values at which Direct Positions are given. The complete coordinate tuples (as per CRS) can be constructed as the cross product of these admissible axis values due to the lattice isomorphy of the grid (for example, such a grid does not contain “holes”). In Figure 9, type Axis – with its subtypes Index Axis, Regular Axis, and Irregular Axis – establishes how isolated, single axes can be described (see Subclause 7.2.2.2). Multiple axes making up a grid (or belonging to it, with further, independent axes existing) Grid axes are defined through Displacement Axis Nest (see Subclause 7.2.2.3). In the most general case, axis information is derived algorithmically, represented by Algorithmic Axis (see Subclause 7.2.2.4).

**7.2.2.2 Index, Regular, and Irregular Axes**

An Index Axis is a 1D Cartesian axis: there is no georeference, and admissible coordinates are at discrete, integer positions and unit-less.

**Requirement 25:**
An Index Axis shall be given by an axis identifier, a CRS, lower and upper bounds $lo$ and $hi$ with $lo, hi \in C$ and $lo \leq hi$. Direct Positions shall be defined for every coordinate tuple where the coordinate value of the Index Axis on hand is from the closed interval $S = \{ x \in \mathbb{Z} | lo \leq x \leq hi \}$. 
Note 4 The unit of measure of an Index Axis is 1, i.e.: coordinate values are unitless.

Note 5 As distances are known (always 1) there is no need to store such values as they can be computed when accessing particular Direct Positions.

A Regular Axis has an equi-distant spacing like an Index Axis, but is continuous and not constrained to integer positions and distances. Such an axis can be georeferenced, i.e.: it can have a spatial or temporal semantics attached.

**Requirement 26:**
A Regular Axis shall be given by an axis identifier, a 1D CRS, lower and upper bounds $lo$ and $hi$ with $lo, hi \in C$ and $lo \leq hi$, a resolution $r \in C$. Direct Positions shall be defined for every coordinate tuple where the coordinate value of the Regular Axis on hand is from the set $S = \{ x \in C \mid lo \leq x \times r \leq hi \}$.

Note 6 Unit of measure, coordinate set, as well as datum are defined in the CRS.

Note 7 Storing Direct Positions along irregular axes requires materialization of the initial point (such as the minimum position on the axis) and the (constant) offset; any storage location of a Direct Position can be computed from this.

The next level of generalization is an Irregular Axis. It is continuous, possibly georeferenced, and its distances are irregular. Such an axis can be georeferenced, i.e.: it can have a spatial or temporal semantics attached.

**Requirement 27:**
An Irregular Axis shall be given by an axis identifier, a 1D CRS, a set of positions $P = \{ p_1, \ldots, p_n \} \subseteq C$. Direct Positions shall be defined for every coordinate tuple where the coordinate value of the Irregular Axis on hand is from $P$.

Note 8 Storing Direct Positions along irregular axes requires materialization of the list of positions as they cannot be computed.
7.2.2.3 Displacement Axis Nest

A Displacement Axis Nest (or Warped Nest) is a set of continuous, possibly georeferenced axes, forming a subset of the CRS’s axes. Relative to a regular grid, each Direct Position is shifted by some individual offset within the CRS space spanned by the axes participating.

Requirement 28:
A Displacement Axis Nest shall be given by a list of axis identifiers with \( d > 0 \) items, a \( d \)-dimensional CRS. Direct Positions shall be defined for every coordinate tuple where the coordinate value of each axis participating in the Displacement Axis Nest on hand is given as a set of \( d \)-dimensional coordinates from the CRS.

Note 9 The parameter set is an application-specific data structure not defined in this document.

Note 10 Storing Direct Positions in this case requires materializing the coordinate values for each coordinate tuple of the CRS subspace spanned by the axes participating.

7.2.2.4 Algorithmic Axis

Algorithmic Axes are given by a set of continuous, possibly georeferenced axes, forming a subset of the CRS’s axes, where the Direct Positions have to be derived algorithmically from some otherwise abstract parameters (hence, the alternative name Transformation Model). Examples of such algorithmic axes include sensor models where, instead of the coordinate information, a set of sensor parameters (such as ground control parameters) is provided which needs to be fed into the model for deriving the actual Direct Positions.

Requirement 29:
An Algorithmic Axis set shall be given by a list of axis identifiers with \( n > 0 \) items, an \( n \)-dimensional CRS, a parameter set \( P \). Direct Positions shall be given through some algorithm parametrized with \( P \).

Note 11 The structure and meaning of \( P \) is not specified further in this document.

7.2.2.5 Combinations

By combining all the above axis types freely, any type of grid can be modelled. The list of possible axis types is not conclusive, some standard or application may define their own additional axis types. Figure 10 shows some sample grid types combining different axis types defined in this document.

In addition to the above construction principles for grids, ISO 19111 CRS handling allows expressing the following situations:

— Skewed grids (7, second from left) can be expressed as an affine transformation applied to some base CRS. This can be formulated by concatenating the base CRS with a suitable Engineering CRS (implementing the affine transformation) as established in ISO 19111.

![Figure 10 — Sample 2D and 3D grids, from left to right: regular, irregular, warped nest, combination of regular Lat/Long with irregular time, combination of warped Lat/Long nest with irregular time](image-url)
7.2.3 Grid Offsets

— Pixel-in-centre vs pixel-in-corner: Sometimes tools interpret values as sitting at the Direct Positions indicated (referred to as pixel-in-centre semantics), sometimes tools assume a half-pixel offset from the Direct Position indicated (referred to as pixel-in-corner semantics). The former situation does not require any special handling, the latter situation – for regular grids – can be expressed again through an affine CRS transformation, specifically: a half-pixel shift of the CRS, applied through a CRS concatenation as established in ISO 19111. Irregular grids require individual handling.

7.3 Grid Cells

7.3.1 Grid cells

Common practice is to consider the information stored at a direct position (such as radiance energy) not concentrated in the zero-extent point, but distributed over some area around this direct position. For example, in Charge-Coupled Device (CCD) sensor arrays each individual sensor collects photons on some finite surface, hence the electrical charge delivered is representative not only for the direct position, but represents an aggregated value for the whole area seen by the CCD sensor.

This concept is captured by the notion of a grid cell, which, in its full generality, is given by a neighbourhood around a direct position. Cells are constrained in that they do not overlap, while “empty” space not covered by any cell may exist within a coverage’s domain extent.

In case of regular grids all of its cells share the same shape and size, otherwise the grid cells in general are not equal in size and shape.

7.3.2 Pixel-in-center, Pixel-in-corner

If cells are positioned in a way that the direct positions corresponding to a cell sit in its center then this is commonly referred to as “pixel in center”. If cells are positioned such that the corresponding direct positions sit in the “upper-left”, “lower-left”, etc. of the cell’s corners is called “pixel in corner”. Figure 11 shows both situations for a regular 2D grid.

In case of a regular grid, all cell positions can be obtained from a “pixel in center” case by applying a translation by half the grid distance along each axis. Technically, this can be achieved through an Engineering Coordinate System operation with these offsets, concatenated with the grid’s CRS.

Note 12 Obviously, pixel-in-center and pixel-in-corner resemble just two special cases of a general shift to be applied; for example, the corner to be chosen might be any of 2n choices for an n-dimensional regular grid. In the most general case any coordinate position within the cell could act as such an “anchor point”. A general model accepted in the Atmospheric science community is the Arakawa grid system.

7.4 Grid Coverage

A Grid Coverage is a Multi-Point Coverage with the specific restriction that the Direct Positions are not arbitrary, but given by some underlying Grid as defined in Subclause 7.2; this holds recursively also for grid coverages built from partitions.

Requirement 30:
The sub-coverage partitions contained in a given Grid Coverage shall, in their entirety, satisfy all requirements established in Subclause 7.2 and Figure 12.

Requirement 31:
Function evaluate() shall be defined, for some coverage c and position p, as evaluate(p) = v for the corresponding point feature f ∈ c with f.contains(p), where contains() is defined in ISO 19107.
Figure 11 — Grid Cell with “pixel-in-center” (left) and “pixel-in-corner” sitting “upper-left” with direct position \((x,y)\) and corresponding grid cell marked with upward \(y\) axis (center) and downward \(y\) axis (right)

Note 13   By construction, Direct Positions in a Grid are pairwise disjoint. Hence, there can be only one associated value for any given Direct Position.

7.5 Rectified and Referenceable Grid Coverages

A grid may be defined in terms of an external coordinate reference system. This requires additional information about the location of the grid’s origin within the external coordinate reference system, the orientation of the grid axes, and a measure of the spacing between the grid lines. If the spacing is uniform, then there is an affine relationship between the grid and external coordinate system, and the grid (Figure 13) is called a rectified grid. If, in addition, the external coordinate reference system is related to the earth by a datum, the grid is a georectified grid. The grid lines of a rectified grid need not meet at right angles; the spacing between the grid lines is constant along each axis but need not be the same on every axis. The essential point is that the transformation of grid coordinates to coordinates of the external coordinate reference system is an affine transformation.

Note 14   The word rectified implies a transformation from an image space to another coordinate reference system. However, grids of this form are often defined initially in an earth-based coordinate system and used as a basis for collecting data from sources other than imagery.

Note 15   The internal grid coordinate system is an instance of an engineering coordinate reference system as specified by ISO 19111. Its datum is a set of one or more ground control points.

EXAMPLE   Figure 13 shows a two-dimensional grid in the 3-space determined by the axes \(X, Y,\) and \(Z\). The grid origin is at \(O\). There are two offset vectors labelled \(V_1\) and \(V_2\) which specify the orientation of the grid axes and the spacing between the grid lines. The coordinates of the grid points are of the form: \(O + aV_1 + bV_2\).

When the relationship between a grid and an external coordinate reference system is not adequate to specify it in terms of an origin, an orientation and spacing in that coordinate reference system, it may still be possible to transform the grid coordinates into coordinates in the coordinate reference system. This transformation need not be in analytic form; it may be a table, relating the grid points to coordinates in the external coordinate reference system. Such a grid is classified as a referenceable grid. If the external coordinate reference system is related to the earth by a datum, the grid is a georeferenceable grid. A referenceable grid is associated with information that allows the location of all points in the grid to be determined in the coordinate reference system, but the location of the points is not directly available from the grid coordinates, as opposed to a rectified grid where the location of the points in the coordinate reference system is derivable from the properties of the grid itself. The transformation produced by the information associated with a referenceable grid will produce a grid as seen in the coordinate reference system, but the grid lines of that grid need not be straight or orthogonal, and the grid cells may be of different shapes and sizes.
The terms rectified grid coverage and georeferenceable grid coverage can be described as follows:

— A Rectified Grid Coverage is a Grid Coverage where every axis is either an Index Axis or a Regular Axis.

— A Referenceable Grid Coverage is a Grid Coverage where at least one axis is neither an Index Axis nor a Regular Axis.
7.6 Further Grid Coverages

Coverages are sometimes based on tessellations composed of regular hexagons. Such tessellations are usually called hexagonal grids.

One example is Hexagonal Grid Coverages which are given by tessellations composed of regular hexagons. Such tessellations are usually called hexagonal grids. The centers of a set of regular hexagons that form such a tessellation correspond to the grid points of a quadrilateral grid (Figure 14). That grid can be described as a rectified grid in which the two offset vectors are of equal length but differ in direction by 60°. The length of a side of the hexagon is \( L = S \tan 30° \), where \( S \) is the length of the offset vector. This means that the values in the coverage range can be stored in a computer as a multi-dimensional array. The hexagons are the Thiessen polygons that are generated around the grid points.

Note 16 A set of Thiessen polygons generated from the grid points of any two-dimensional rectified grid described by two offset vectors that are equal in length but not orthogonal will be a set of congruent hexagons. The hexagons will be irregular – and, hence, out of scope – unless the offset vectors differ in direction by exactly 60°.
A Hexagonal Grid Coverage (Figure 15) evaluates a coverage at direct positions within a network of hexagons centred on a set of grid points. Evaluation is based on interpolation between the centres of the value hexagons surrounding the input position.

8 Multi-Curve Coverages

8.1 Overview

This Clause defines conformance class package CV_MultiCurveCoverage.

8.2 General Multi-Curve Coverages

A MultiCurveCoverage is a coverage consisting of a collection of curves.

EXAMPLE A coverage that assigns a route number, a name, a pavement width and a pavement material type to each segment of a road network can be represented as a MultiCurveCoverage.

Requirement 32: A MultiCurveCoverage shall contain only elements of the same type, which is ISO 19107 data type CurveData or a subtype thereof, as described by Figure 16.

Requirement 33: Function evaluate() shall be defined, for some coverage c and position p, as evaluate(p) = { v | ∃ curve feature f ∈ c: f.contains(p) } where contains() is defined in ISO 19107.

Note 1 For the avoidance of doubts, a MultiCurveCoverage can contain additional dimensions (such as time).

Figure 16 — Class MultiCurveCoverage
8.3 Segmented curve coverages

Segmented curve coverages are used to model phenomena that vary continuously or discontinuously along curves, which may be elements of a network. The domain of a segmented curve coverage is described by a set of curves and includes all the direct positions in all of the curves in the set (Figure 18).

![Diagram of CV_SegmentedCurveCoverage class]

9 Multi-Surface Coverages

9.1 Overview

This Clause defines conformance class package CV_MultiSurfaceCoverage.

9.2 General Multi-Surface Coverages

A MultiSurfaceCoverage is a coverage consisting of a collection of surfaces.

EXAMPLE A coverage that represents soil types typically has a spatial domain composed of surfaces with irregular boundaries.

Requirement 34:
A MultiSurfaceCoverage shall contain only elements implementing ISO 19107 data type SurfaceData or a subtype thereof, as described by Figure 18.

Requirement 35:
Function evaluate() shall be defined, for some coverage c and position p, as

\[ \text{evaluate}(p) = \{ v \mid \exists \text{ surface feature } f \in c : f \text{.contains}(p) \} \]

where \( \text{contains()} \) is defined in ISO 19107.

There are various practically relevant subtypes of multi-surface coverages, including polyhedral surfaces and their special case of Triangulated Irregular Networks (TINs). While in the previous version of 19123 TIN coverages were modelled separately they now can be obtained through subtyping of ISO 19107 Surface.

Note 1 For the avoidance of doubts, a MultiSurfaceCoverage can contain additional dimensions (such as time).
9.3 Further Surface Coverages

9.3.1 General

So far surface coverages have been considered which represent a bundle of surfaces not constrained further. Some applications consider surfaces establishing tessellations.

9.3.2 Thiessen Polygon Coverages

A finite collection of points on a plane determines a partition of the plane into a collection of polygons equal in number to the collection of points. A Thiessen polygon is generated from one of a defining set of points by forming the set of direct positions that are closer to that point than to any other point in the defining set. The specific point is called the centre of the resulting polygon. The boundaries between neighbouring polygons are the perpendicular bisectors of the lines between their respective centres. Each polygon shares each of its edges with exactly one other polygon. Each polygon contains exactly one point from the defining set. Thiessen polygons are also known as Voronoi Diagrams or Proximal Sets.

A Thiessen polygon network (see Figure 20 for an example) is a tessellation of a 2D space into surfaces bounded by Thiessen Polygons. A finite collection of points on a plane determines a partition of the plane into a collection of polygons equal in number to the collection of points. A Thiessen polygon is generated from one of a defining set of points by forming the set of direct positions that are closer to that point than to any other point in the defining set. The specific point is called the centre of the resulting polygon. The boundaries between neighbouring polygons are the perpendicular bisectors of the lines between their respective centres. Each polygon shares each of its edges with exactly one other polygon. Each polygon contains exactly one point from the defining set. Thiessen polygons are also known as Voronoi Diagrams or Proximal Sets.

EXAMPLE Figure 20 shows a collection of points with their (x, y) coordinates, the perpendicular of the lines that would be drawn between them, and the resultant polygons.

Evaluation of a Thiessen Polygon Coverage involves two steps. The first is to find the Thiessen polygon that contains the input Direct Position; the second is to interpolate the feature attribute values at the Direct Position from the geometry/value pairs at the centres of the surrounding Thiessen polygons.

Technically, a Thiessen Polygon Coverage is a specialization of MultiSurfaceCoverage as shown in Figure 18.
9.3.3 Triangulated Irregular Networks (TINs)

Class CV_TinCoverage is shown in Figure 21. The basic idea of a TIN is to partition of the points in the spatio-temporal domain of a discrete point coverage into a computationally unique set of non-overlapping triangles. Each triangle is formed by three of the points in the spatio-temporal domain of the discrete point coverage (Figure 22). The Delaunay triangulation method is commonly used to produce TIN tessellations with triangles that are optimally equiangular in shape, and are generated in such a manner that the circumscribing circle containing each triangle contains no point of the discrete point coverage other than those at the vertices of the triangle.

10 Multi-Solid Coverages

This Clause defines conformance class package CV_MultiSolidCoverage.

A MultiSolidCoverage is a coverage consisting of a collection of solids.

Requirement 36:

A MultiSolidCoverage shall contain only elements of the same type, which is ISO 19107 data type SolidData or a subtype thereof, as described by Figure 23.
**Requirement 37:**
Function $\text{evaluate}()$ shall be defined, for some coverage $c$ and position $p$, as

$$\text{evaluate}(p) = \{ v \mid \exists \text{ solid feature } f \in c : f.\text{contains}(p) \}$$

where $\text{contains}()$ is defined in ISO 19107.

**Note 1** For the avoidance of doubts, a MultiSolidCoverage can contain additional dimensions (such as time).

---

![Class TinCoverage](image1)

**Figure 21 — Class TinCoverage**

![Sample TIN coverage](image2)

**Figure 22 — Sample TIN coverage**

![Class MultiSolidCoverage](image3)

**Figure 23 — Class MultiSolidCoverage**
Annex A
(normative)

Conformance Tests

A.1 Conformance Class

This standard defines six conformance classes: Coverage Core (specification target: CV_Coverage), CV_MultiPointCoverage), Grid Coverage (specification target: CV_GridCoverage), Multi-Curve Coverage (specification target: CV_MultiCurveCoverage), Multi-Surface Coverage (specification target: CV_MultiSurfaceCoverage), and Multi-Solid Coverage (specification target: CV_MultiSolidCoverage).

Standardization targets are specifications containing provisions for coverages. A specification claiming conformance to this document shall implement the conformance class relevant to that specification target.

Conformance with this standard shall be assessed using all the relevant conformance test cases specified in this Annex A.

A.2 Conformance Class Coverage Core

A.2.1 Requirement 1

Test statement: An instantiation of package CV_Coverage shall have all instances and properties specified for this package, its contents, and its dependencies for CV_Coverage as per Figure 2 and Figure 3.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.2 Requirement 2


Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.3 Requirement 3

Test statement: The coverage CRS shall be described in accordance with ISO 19111 describes the semantics of CRSs and their axes, as well as their composition into higher-dimensional CRSs.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.4 Requirement 4

Test statement: A coverage CRS shall be identified by an ISO 19107 DirectPosition::RSID.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.
A.2.5 Requirement 5

Test statement: In a coverage CRS, coordinates with a spatial semantics shall be represented by an ISO 19107 DirectPosition.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.6 Requirement 6

Test statement: In a coverage CRS, coordinates with a temporal semantics shall be represented by an ISO 19108 TM_Position.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.7 Requirement 7

Test statement: The Range Type component shall be of type Record Type as defined in ISO 19103.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.8 Requirement 8

Test statement: The Range Type of a coverage at minimum shall provide the information necessary to decode the Range Set values and process them in a computer system, consisting of (i) data type information in some available typing system, (ii) null values, and (iii) unit of measure.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.9 Requirement 9

Test statement: Interpolation types allowed for the geometries contained in a coverage shall be given by the interpolation methods specified in ISO 19107 for the respective geometry type.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.10 Requirement 10

Test statement: A CV_CoverageByGeometryValuePair shall be structured as described in Figure 6 —.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.11 Requirement 11

Test statement: A CV_CoverageByPartitioning shall be structured as described in Figure 6 —.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.12 Requirement 12

Test statement: A CV_CoverageByDomainAndRange shall be structured as described in Figure 6 —.
Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.13 Requirement 13

Test statement: All sub-coverages in a partition shall share the same CRS and range type; their Domain Sets shall fulfil all constraints imposed on the particular coverage type of the super-coverage.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.14 Requirement 14

Test statement: A coverage shall not recursively contain itself in a partitioning hierarchy.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.15 Requirement 15

Test statement: A CV_CoverageByFunction shall be structured as described in Figure 6 — .

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.16 Requirement 16

Test statement: The Envelope of a Coverage shall contain all or a subset of the axes of the corresponding Domain Set.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.17 Requirement 17

Test statement: The Envelope of a Coverage should approximate its Domain Set as closely as possible, for all axes of the Domain Set present in Extent.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.18 Requirement 18

Test statement: If a Coverage contains an Envelope then this Envelope shall contain the Domain Set of this Coverage.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.2.19 Requirement 19

Test statement: The CRS component of the Coverage shall reference a CRS definition conformant with ISO 19111.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.
A.3 Conformance Class Multi-Point Coverage

A.3.1 Requirement 20

Test statement: A MultiPointCoverage shall contain only elements of the same type, which is ISO 19107 data type PointData or a subtype thereof, as described by Figure 8.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.3.2 Requirement 21

Test statement: Function evaluate() shall be defined, for some coverage c and position p, as evaluate(p) = { v | \exists point feature f \in c: f.contains(p) } where contains() is defined in ISO 19107.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.3.3 Requirement 22

Test statement: A Grid Coverage shall be a subtype of Multi-Point Coverage.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.4 Conformance Class Grid Coverage

A.4.1 Requirement 23

Test statement: In a Grid Coverage, the Direct Position shall be given by a rectangular Grid.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.4.2 Requirement 24

Test statement: In a Grid Coverage, Direct Position has exactly one value (taken from the coverage’s Range Type) associated, i.e.: | evaluate(p) | = 1 for all Direct Positions p.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.4.3 Requirement 25

Test statement: An Index Axis shall be given by an axis identifier, a CRS, lower and upper bounds lo and hi with lo, hi \in C and lo \leq hi. Direct Positions shall be defined for every coordinate tuple where the coordinate value of the Index Axis on hand is from the closed interval S = { x \in \mathbb{Z} | lo \leq x \leq hi }.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.4.4 Requirement 26

Test statement: A Regular Axis shall be given by an axis identifier, a 1D CRS, lower and upper bounds lo and hi with lo, hi \in C and lo \leq hi, a resolution r \in C. Direct Positions shall be defined for every coordinate tuple where the coordinate value of the Regular Axis on hand is from the set S = { x \in C | lo \leq x*r \leq hi }.
Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.4.5 Requirement 27

Test statement: An Irregular Axis shall be given by an axis identifier, a 1D CRS, a set of positions $P = \{ p_1, \ldots, p_n \} \subseteq C$. Direct Positions shall be defined for every coordinate tuple where the coordinate value of the Irregular Axis on hand is from $P$.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.4.6 Requirement 28

Test statement: A Displacement Axis Nest shall be given by a list of axis identifiers with $d > 0$ items, a $dD$ CRS. Direct Positions shall be defined for every coordinate tuple where the coordinate value of each axis participating in the Displacement Axis Nest on hand is given as a set of $d$-dimensional coordinates from the CRS.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.4.7 Requirement 29

Test statement: An Algorithmic Axis set shall be given by a list of axis identifiers with $n > 0$ items, an $nD$ CRS, a parameter set $P$. Direct Positions shall be given through some algorithm parametrized with $P$.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.4.8 Requirement 30

Test statement: The sub-coverage partitions contained in a given Grid Coverage shall, in their entirety, satisfy all requirements established in 7.2 and Figure 12 — .

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.4.9 Requirement 31

Test statement: Function $evaluate()$ shall be defined, for some coverage $c$ and position $p$, as $evaluate(p) = v$ for the corresponding point feature $f \in c$ with $f.contains(p)$, where $contains()$ is defined in ISO 19107.

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.5 Conformance Class Multi-Curve Coverage

A.5.1 Requirement 32

Test statement: A MultiCurveCoverage shall contain only elements of the same type, which is ISO 19107 data type CurveData or a subtype thereof, as described by Figure 16 — .

Test procedure: Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.
A.5.2 Requirement 33

**Test statement:** Function `evaluate()` shall be defined, for some coverage `c` and position `p`, as `evaluate(p) = \{ v | \exists \text{ curve feature } f \in c: f.contains(p) \}` where `contains()` is defined in ISO 19107.

**Test procedure:** Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.6 Conformance Class Multi-Surface Coverage

A.6.1 Requirement 34

**Test statement:** A MultiSurfaceCoverage shall contain only elements implementing ISO 19107 data type `SurfaceData` or a subtype thereof, as described by Figure 18 — .

**Test procedure:** Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.6.2 Requirement 35

**Test statement:** Function `evaluate()` shall be defined, for some coverage `c` and position `p`, as `evaluate(p) = \{ v | \exists \text{ surface feature } f \in c: f.contains(p) \}` where `contains()` is defined in ISO 19107.

**Test procedure:** Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.7 Conformance Class Multi-Solid Coverage

A.7.1 Requirement 36

**Test statement:** A MultiSolidCoverage shall contain only elements of the same type, which is ISO 19107 data type `SolidData` or a subtype thereof, as described by Figure 23 — 17.

**Test procedure:** Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.

A.7.2 Requirement 37

**Test statement:** Function `evaluate()` shall be defined, for some coverage `c` and position `p`, as `evaluate(p) = \{ v | \exists \text{ solid feature } f \in c: f.contains(p) \}` where `contains()` is defined in ISO 19107.

**Test procedure:** Inspect the specification text as well as any associated (machine or human readable) files, such as schema definitions, for conformance.
Annex B
(informative)

Interpolation methods

B.1 General

Evaluation of a continuous coverage involves interpolation between known feature attribute values associated with geometric objects in the spatio-temporal domain of the discrete coverage that is provided as control for the continuous coverage. There are several interpolation methods. Each is used in the context of specified geometric configurations (Table B.1).

The enumerated data type CV_InterpolationMethod includes the following methods: nearest neighbour, linear, quadratic, cubic, lost area, and barycentric. These are described in B.2 through B.10.

A set of interpolation methods which also apply to coverages is given in ISO 19107. This document defines further interpolation techniques below; some of these are identical to interpolations defined in ISO 19107, but are still kept to introduce them under the name that has been established by earlier versions of this document. Generally, an application or standard may define additional ways of interpolation.

Since CV_InterpolationMethod is a CodeList (Figure B.1) it may be extended in an application schema that specifies additional interpolation methods.

![Figure B.1 — Interpolation method codelist](image_url)

<table>
<thead>
<tr>
<th>Method</th>
<th>Coverage Type</th>
<th>Coverage Dimension</th>
<th>Subclause</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest Neighbour</td>
<td>Any</td>
<td>Any</td>
<td>B.2</td>
</tr>
<tr>
<td>Linear</td>
<td>Any</td>
<td>Any</td>
<td>B.3</td>
</tr>
<tr>
<td>Quadratic</td>
<td>Any</td>
<td>Any</td>
<td>B.4</td>
</tr>
<tr>
<td>Cubic</td>
<td>Any</td>
<td>Any</td>
<td>B.5</td>
</tr>
<tr>
<td>Lost Area</td>
<td>Thiessen Polygon, Hexagonal Grid</td>
<td>2</td>
<td>B.6</td>
</tr>
<tr>
<td>Barycentric</td>
<td>TIN</td>
<td>2</td>
<td>B.7</td>
</tr>
</tbody>
</table>
B.2 Nearest neighbour interpolation

Nearest neighbour interpolation can be applied to any coverage. It generates a feature attribute value at a Direct Position by assigning it the feature attribute value associated with the nearest Direct Position in the spatio-temporal domain of the coverage. Nearest neighbour interpolation extends a discrete coverage to a step function defined on the convex hull of the domain objects in the domain of the coverage. Nearest neighbour interpolation is the only interpolation method described in this document that can be used to interpolate nominal or ordinal range values.

B.3 Linear interpolation

Linear interpolation serves to compute values between Direct Positions in any-dimensional domain sets through a quadratic polynom. The mathematical principle is shown for the one-dimensional and two-dimensional cases.

1-D linear interpolation is commonly used to interpolate along curves. It is based on the assumption that feature attribute values at different positions along the curve differ in proportion to the distances between those positions:

Given two point value pairs \((p_s, v_s)\) and \((p_t, v_t)\), where \(p_s\) is the start point and \(p_t\) is the end point of a value segment, and \(v_s\) and \(v_t\) are the range values associated with those points, the range value \(v\) associated with the direct position \(x\) is:

\[
v(x) = v_s + (v_t - v_s) \left( \frac{x - p_s}{p_t - p_s} \right)
\]

Bilinear interpolation is used to compute range values at Direct Positions in two dimensions. Given a direct position \(p\) contained in a grid cell whose vertices are \(V, V + V_1, V + V_2,\) and \(V + V_1 + V_2\), and with range values at these vertices of \(w_1, w_2, w_3,\) and \(w_4\), respectively, there are unique numbers \(x\) and \(y\), with \(0 < x < 1, \) and \(0 < y < 1\) such that \(p = V + xV_1 + yV_2\). The range value at \(p\) is:

\[
w(x, y) = (1-x)(1-y) w_1 + x(1-y) w_2 + y(1-x) w_3 + xy w_4
\]

Note 1 In a Cartesian grid, \(V_1\) and \(V_2\) are the unit vectors \((0,1)\) and \((1,0)\).

B.4 Quadratic interpolation

Quadratic interpolation serves to compute values between Direct Positions in any-dimensional domain sets through a quadratic polynom. The mathematical principle is shown for the one-dimensional and two-dimensional cases.

1-D quadratic interpolation is used to interpolate along curves. It is based on the quadratic polynomial with coefficients \(a\) through \(c\):

\[
v(x) = a + bx + cx^2
\]

where

\(a\) is the range value at the start of a value segment and

\(v\) is the range value at distance \(x\) along the curve from the start.

Three point value pairs are needed to provide control values for calculating the coefficients of the function.
Biquadratic interpolation is used to compute range values at Direct Positions in two dimensions. It is based on the biquadratic polynomial:

\[ v = a + bx + cy + dx^2 + exy + gy^2 + hxy^2 + ix^2y^2 \]

**B.5 Cubic interpolation**

1-D cubic interpolation serves to compute values between Direct Positions in any-dimensional domain sets through a cubic polynomial. The mathematical principle is shown for the one-dimensional and two-dimensional cases.

Cubic interpolation is used to interpolate along curves. It is based on the cubic polynomial, based on coefficients \(a\) through \(d\):

\[ v(x) = a + bx + cx^2 + dx^3 \]

where

- \(a\) is the range value at the start of a value segment and
- \(v\) is the range value at distance \(x\) along the curve from the start.

Bicubic interpolation is used to compute range values at direct positions in two dimensions. Bicubic interpolation uses the following formula, based on coefficients \(a_0\) through \(a_{15}\):

\[ v(x) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 + a_6 x^2y + a_7 x y^2 + a_8 x^3 + a_9 x^2 y + a_{10} x y^3 + a_{11} x^3 y + a_{12} x y^3 + a_{13} x^3 y^2 + a_{14} x^2 y^3 + a_{15} x y^3 \]

**B.6 Lost area interpolation**

Lost area interpolation extends a Multi Point coverage to a continuous function, \(f\), defined on the convex hull of the domain of the point coverage.

Let \(D = \{x_1, x_2, \ldots, x_n\}\) be the domain of the point coverage, and let \(\{V_1, V_2, \ldots, V_n\}\) be the Thiessen polygons generated by the set \(D\).

Suppose it is desired to calculate \(f(q)\), where \(q\) is a Direct Position in the convex hull of \(D\). Begin by forming the Thiessen polygons generated by \(D\); then add \(p\) to the \(D\) and form the Thiessen polygons for the set of \(n+1\) points: \(\{x_1, x_2, \ldots, x_n, p\}\). The two sets of polygons are identical, except that each of the polygons coterminous with the polygon containing \(q\) “loses area” to the new polygon containing \(p\).

The interpolation forms the weighted average such that each feature attribute value contributes to the feature attribute value at \(p\) according to the amount of area its polygon lost to the polygon at \(p\). More formally:

a) Suppose that the discrete point coverage is characterized by the point value pairs: \(\{(x_1, v_1), (x_2, v_2), \ldots, (x_n, v_n)\}\).

b) Among the Thiessen polygon set formed by \(\{x_1, x_2, \ldots, x_n, p\}\), those coterminous with the polygon containing \(p\) are \(\{V_1, V_2, \ldots, V_k\}\).

c) The corresponding Thiessen polygons from the set generated by \(\{x_1, x_2, \ldots, x_n\}\) are \(\{V'_1, V'_2, \ldots, V'_k\}\).

d) The area lost by the \(i^{th}\) polygon is \(V'_i - V_i\).
e) The total area lost is $\sum (V'i - Vi)$ where the sum is over $i$ from 1 to $k$ (that is, the sum is over all polygons that lost area to the polygon containing $q$). Note that this sum is the same as the area of the Thiessen polygon containing $p$.

f) Then the interpolated feature attribute value at $p$ is:

$$f(p) = \frac{\sum vi^* (V'i - Vi)}{\sum (V'i - Vi)}$$

where the summations are over the same range: $i = 1, ..., k$.

**B.7 Barycentric interpolation**

Let $P$, $Q$, and $R$ denote the vertices of a triangle. For any direct position, $S$, in the triangle, there is a unique triple of numbers, $i$, $j$, and $k$, with $0 \leq i \leq 1$, $0 \leq j \leq 1$, and $0 \leq k \leq 1$, and with $i + j + k = 1$, such that

$$S = iP + jQ + kR$$

The numbers $(i, j, k)$ are the barycentric coordinates of $S$.

The name “barycentric” comes from the fact that using the equation above, $S$ is the centre of mass of a triangle with point masses of size $i$, $j$, and $k$ at the corners $P$, $Q$ and $R$ respectively. As one allocates mass to the three corners, the centre of mass can occupy any direct position in the triangle.

Given a value triangle composed of the CV_PointValuePairs $(p_1, v_1)$, $(p_2, v_2)$, and $(p_3, v_3)$, and a Direct Position, $S$, inside it, the barycentric coordinates of $S$ are $(i, j, k)$, where $S = ip_1 + jp_2 + kp_3$ and the feature attribute value at $S$ is $v = iv_1 + jv_2 + kv_3$.

**B.8 Other**

Other interpolation methods may be defined in an Application Schema that makes use of this standard.
Annex C
(informative)

Sequential enumeration

C.1 General

Linearization of coverage range values (and their pertaining direct positions), such as for storage, may follow a sequence rule that assigns order to space (Figure C.1). The rule may be as simple as Row then Column, or it may be more complex. Complex rules allow for Quadtrees, or more general structures in Riemann hyperspace, Hilbert space and other patterns. An example is a spiral search pattern that may be used in Search and Rescue. CV_SequenceRule is a data type that describes the mapping of grid coordinates to a position to attribute values along axis. CV_SequenceType is a code list that identifies methods for sequential enumeration of the grid points. Sample methods for sequential enumeration are described in this Annex C.

<table>
<thead>
<tr>
<th>CV_SequenceRule</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ type :CV_SequenceType = linear</td>
</tr>
<tr>
<td>+ scanDirection :Sequence&lt;CharacterString&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>«codelist» CV_SequenceType</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ linear</td>
</tr>
<tr>
<td>+ boustrophedonic</td>
</tr>
<tr>
<td>+ CantorDiagonal</td>
</tr>
<tr>
<td>+ spiral</td>
</tr>
<tr>
<td>+ Morton</td>
</tr>
<tr>
<td>+ Hilbert</td>
</tr>
</tbody>
</table>

Figure C.1 – Sequence Rule structure and codelist

A sequential enumeration specifies a traversal order of a coverage domain set. Two or more axis may be coded together to provide one linear order on a space. Such ordering is sometimes of value because it may provide properties to the number space that are or use in some operations on the data. For example, Riemann hyperspace is traversed by Morton order. This allows for variable size cells (such as in a quadtree in 2 dimensions) and obeys the Riemann criteria where values that are close together in the Morton order are also close together in physical space. Technically this is an extension of the Pythagorean theorem into hyperspace. CV_SequenceType provides a list of codes for identifying types of sequencing methods. This Annex explains those types in greater detail.

There are several sequencing rules based on incrementing or decrementing grid coordinate values in a simple fashion. More complex space filling curves can also be used. Space filling curves are generated by progressively subdividing a space in a regular way and connecting the elements resulting from each subdivision according to some rule. They can be used to generate a grid, but they can also be used to assign an ordering to the grid points or grid cells in a separately defined grid. They lend themselves more readily than simple incrementing methods to sequencing in grids that have irregular shapes or cells of variable size.

In every case, ordering of the grid cells starts by incrementing coordinates along one grid axis. At some point in the process, it begins to increment coordinates along a second grid axis, then a third, and so on until it has progressed in the direction of each of the grid axes. The figures in this annex provide examples. The attribute CV_SequenceRule.scanDirection provides a list of signed axis names that identifies the order in which scanning takes place. The list may include an additional element to support interleaving of feature attribute values (see C.8 for a more detailed discussion of interleaving).

Ordering is continuous if consecutive pairs of grid cells in the sequence are maximally connected. It is semi-continuous if consecutive pairs of grid cells are connected, but less than maximally connected, and discontinuous if consecutive pairs of cells are not connected.
EXAMPLE In the 2-dimensional case, a quadrilateral grid cell is connected to the eight cells with which it shares at least one corner. It is maximally connected to the four cells with which it shares an edge and two corners. In the three-dimensional case, a cell is maximally connected to those cells with which it shares a face.

Note 1 In the example diagrams of this annex, continuous segments of scan lines are shown as solid lines, and discontinuous segments are shown as dashed lines.

C.2 Linear scanning

In linear scanning (Figure C.2), feature attribute value records are assigned to consecutive grid points along a single grid line parallel to the first grid axis listed in scanDirection. Once scanning of that row is complete, assignment of feature attribute value records steps to another grid line parallel to the first and continues to step from grid line to grid line in a direction parallel to the second axis. If the grid is 3-dimensional, the sequencing process completes the assignment of feature attribute value records to all grid points in one plane, then steps to another plane, then continues stepping from plane to plane in a direction parallel to the third axis of the grid. The process can be extended to any number of axes. Linear scanning is continuous only along a single grid line.

![Figure C.2 — Examples of linear scanning in a 2-dimensional grid](image)

Note 2 The axes of 2-dimensional grids are often called “row” (horizontal) and “column” (vertical). In this case, scanning in (x,y) order is sometimes called row or row-major scanning.

C.3 Boustrophedonic scanning

In a variant of linear scanning, known as boustrophedonic or byte-offset scanning, the direction of the scan is reversed on alternate grid lines (Figure C.3). In the case of a 3-dimensional grid, it will also be reversed in alternate planes. Boustrophedonic scanning is continuous.

![Figure C.3 — Examples of boustrophedonic scanning in a 2-dimensional grid](image)

C.4 Cantor-diagonal scanning

Cantor-diagonal scanning, also called zigzag scanning, orders the grid points in alternating directions along parallel diagonals of the grid (Figure C.4). The scan pattern is affected by the direction of first step. Like linear scanning, Cantor-diagonal scanning can be extended to grids of three or more dimensions by repeating the scan pattern in consecutive planes. Cantor-diagonal scanning is semi-continuous within a single plane.
C.5 Spiral scanning

Spiral scanning (Figure C.5) can begin either at the centre of the grid (outward spiral), or at a corner (inward spiral). Like linear or Cantor-diagonal scanning, spiral scanning can be extended to grids of three or more dimensions by repeating the scan pattern in consecutive planes. Spiral scanning is continuous in any one plane, but continuity in grids of more than two dimensions can only be maintained by reversing the inward/outward direction of the scan in alternate planes.

C.6 Morton order

Morton ordering is typically based on a space-filling curve generated by progressively subdividing a space into quadrants and ordering the quadrants in a Z pattern as shown in Figure C.6. The ordering index for each grid point is computed by converting the grid coordinates to binary numbers and interleaving the bits of the resulting values. Given the list of the grid axes specified by CV_SequenceRule.scanDirection, the bits of the coordinate corresponding to an axis are less significant than those of the coordinate corresponding to the next axis in the list. Morton ordering can be extended to any number of dimensions. Morton ordering is discontinuous.

Note 3 Because of the shape of the curve formed by the initial ordering of quadrants, Morton ordering is also known as Z ordering.

A grid generated with the Morton ordering technique described will be square and its size in each direction will be a multiple of a power of two. However, the bit interleaving technique for generating an index can be used to
order the grid points in any grid, including grids that are irregular in shape or have grid cells of different sizes (Figure C.7).

Morton ordering can also be used with subdivisions of higher order than 2x2. A 3x3 subdivision for example, or any other odd number, preserves the location of the central cell in systems with hierarchical subdivisions (Figure C.7). For 3x3 subdivision the ordering index for each grid point is computed by converting the grid coordinates to base3 digits and interleaving the base3 digits of the resulting values. In the 2-dimensional case pairs of base3 digits can be combined to form a base 9 digit, in 3- or 4-dimensions groups of 3- or 4- base3 digits can be combined to form base 27 or base 81 digits, any of which can be coded as a single ASCII digit.

C.7 Hilbert order

Like Morton ordering, Hilbert ordering is based on a space-filling curve generated by progressively subdividing a space into quadrants, but the initial pattern of subdivision is different for Hilbert curves. Further subdivision involves replacement of parts of the curve by different patterns (Figure C.8), unlike the simple replication of a single pattern as in Morton ordering. There are two sets of patterns. The left-hand column of the figure includes those for which the sense of the scan directions is the same – both are positive or both negative. The right-hand column of the figure includes those for which the sense of the scan directions is opposite – one is positive and one is negative. A Hilbert curve can only be constructed with patterns from the same set; it uses all the patterns in that set.

Note 4 Because of the shape of the curve formed by the initial ordering of quadrants, Hilbert ordering is also known as Pi ordering.

Computation of the ordering index is more complicated for Hilbert ordering than for Morton ordering. Algorithms for the 2-dimensional case (Figure C.9) are described in [13] and [3]. 3-dimensional Hilbert curves are discussed in [2]. Hilbert ordering is continuous.

C.8 Interleaving of feature attribute values

When the range of a grid coverage includes more than one feature attribute, the feature attribute values may be interleaved in various ways within a list. Such interleaving can be described by including an element for the range in the list of axes provided by the scanDirection attribute of CV_SequenceRule. The index for the record of attributes is then incremented in the same way as the coordinates.

EXAMPLE Consider the 2 x 2 grid in Figure C.10. It has a range (r) of two attributes, A and B. Assuming a linear scan positive first in the x and then in the y direction, the scan order can be selected to access the feature attribute values in the different ways shown in Table C.1.
Figure C.8 — Replacement patterns for generating a Hilbert curve

Figure C.9 — Examples of Hilbert ordering in a 2-dimensional grid

Figure C.10 — Example of a 2-dimensional grid with a range of two attributes
Table C.1 — Examples of interleaving

<table>
<thead>
<tr>
<th>Order</th>
<th>Scan direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r, x, y</td>
</tr>
<tr>
<td>1</td>
<td>A_{11}</td>
</tr>
<tr>
<td>2</td>
<td>B_{11}</td>
</tr>
<tr>
<td>3</td>
<td>A_{21}</td>
</tr>
<tr>
<td>4</td>
<td>B_{21}</td>
</tr>
<tr>
<td>5</td>
<td>A_{12}</td>
</tr>
<tr>
<td>6</td>
<td>B_{12}</td>
</tr>
<tr>
<td>7</td>
<td>A_{22}</td>
</tr>
<tr>
<td>8</td>
<td>B_{22}</td>
</tr>
</tbody>
</table>
Annex D
(normative)

Data-Centric Coverage Specification

D.1 Overview

Historically, modelling approaches in ISO have evolved. Earlier, data structures have been described. More recently, ISO has moved to specifying interfaces instead because this better hides implementation details. Such a step of abstraction is important in particular with the separation of the original, single 19123:2003 specification into an abstract specification in 19123-1 and an implementation guidance in 19123-2.

The interface-centric approach to coverage modelling is adopted in the body of this standard. At the same time it normatively retains the data-centric coverage description from the older version of 19123 as Annex D. As such, coverage data structures of 19123:2003 represent one valid generic data structure out of many possible realizations satisfying the interface of 19123-1. Specifically, the 19123:2003 coverage structure specification is retained for compatibility with the many implementations that have realized these generic classes, albeit in their individual ways. This annex is not a retention of the old elements from ISO 19123:2005, but rather a definition of equivalent classes in terms of the new 19123-1 structure to satisfy the backward compatibility requirement.

The previous version of ISO 19123:2005 defines generic data classes which could be specialized by the developers of more detailed product specifications which could then be implemented. External organizations reference these generic data classes. These classes are REALIZED in these external standards. Since references to these classes exist in external standards it is important that equivalent generic data classes be maintained in this version of the document.

The current version of ISO 19123-1 takes a different approach. It defines an interface which provides a standardized window through which a variety of different data structures could produce the same result. In that way it is more flexible than the previous approach. But it is not in conflict with the previous approach. The previous generic data classes are still valid, and the external standards or product specifications that referenced them also remain valid. The new version of ISO 19123-1 is simply more flexible allowing many different implementation structures, not just the defined generic data classes presented in this annex.

This annex specifies the equivalent abstract data classes from 19123:2005 that are developed in this version to retain compatibility. These form a compliant data structure that can be interpreted through the interface structure defined in the revised 19123-1. The classes and attributes are mapped from the older structure to the new interface structure.

D.2 Generic Data Structure Coverage Schema

D.2.1 Generic Data Structure Package

This annex is described in a separate package from the rest of the model in the 19123-1 standard in order to ensure that the namespace is unique. That is, the class name CV_Coverage <<interface>> is used both to describe the root class of the interface structure, and the class name CV_Coverage <<featureType>> is used in this annex to describe the root class of the generic data structure. CV_Coverage <<featureType>> realizes CV_Coverage <<interface>>. The names are used this way to ensure backward compatibility. A mapping is shown from the generic data structure classes to the new interface classes CV_Coverage Feature Type Generic Data Class.
D.2.2 Class CV_Coverage as a Data Structure Template

D.2.2.1 Class description

The class CV_Coverage <<featureType>> is an instance of the <<metaclass>> GF_FeatureType (ISO 19109), which therefore represents a feature type. CV_Coverage <<featureType>> supports five attributes, one operation, and three associations. This is illustrated in Figure D.1.

![Diagram of CV_Coverage]

**Figure D.1 — CV_Coverage**
D.2.2.2 Attributes

D.2.2.2.1 domainExtent

The attribute domainExtent: EX_Extent[1.."] contains the extent of the domain of the coverage. The data type EX_Extent is defined in ISO 19115-1. Extents may be specified in space, time or space-time. The attribute domainExtent has been generalized in ISO 19123-1. This older narrower extent has been retained for backward compatibility.

D.2.2.2.2 rangeType

The attribute rangeType: RecordType describes the range of the coverage. The data type RecordType is defined in ISO 19103. It consists of a list of attribute name/data type pairs. A simple list is the most common form of rangeType, but RecordType can be used recursively to describe more complex structures. The rangeType for a specific coverage shall be specified in an application schema.

D.2.2.2.3 commonPointRule

The optional attribute commonPointRule: CV_CommonPointRule identifies the procedure to be used for evaluating the CV_Coverage at a position that falls either on a boundary between geometric objects or within the boundaries of two or more overlapping geometric objects, where the geometric objects are either CV_DomainObjects or CV_ValueObjects. This attribute is optional and takes the default value of the attribute as “average”. The attribute is not required in the case when there is only one extent per coverage; that is, when geometric objects do not overlap.

D.2.2.2.1 interpolationType

The optional attribute interpolationType: CV_InterpolationMethod is a code that identifies the interpolation method that shall be used to derive a feature attribute value at any direct position within the CV_ValueObject. The attribute is optional. The default value is “nearestNeighbor”. No value is needed for an analytical coverage (one that maps direct position to attribute value by using a mathematical function rather than by interpolation). Interpolation methods are described in clause 5.6 and in Annex B. The code list identifying the interpolation method is shown in Figure D.3.

D.2.2.2.2 interpolationParameterTypes

Although many interpolation methods use only the values in the coverage range as input to the interpolation function, there are some methods that require additional parameters. The optional attribute interpolationParameterTypes specifies the types of parameters that are needed to support the interpolation method identified by interpolationType. The data type RecordType is specified in ISO 19103. It is a dictionary of names and data types.

D.2.2.3 Operation

D.2.2.3.1 evaluate

The operation evaluate: (DirectPosition, Sequence <CharacterString>): Record accepts a DirectPosition as input and return a set of Records of feature attribute values for that direct position. The parameter list is a sequence of feature attribute names each of which identifies a field of the rangeType. If list is null, the operation returns a value for every field of the rangeType. Otherwise, it returns a value for each field included in list. Class DirectPosition is defined in ISO 19107; the data type Record is defined in ISO 19103. If the direct position passed is not in the domain of the coverage, then an error message shall be generated. If the input DirectPosition falls within two or more geometric objects within the domain, the operation shall return records of feature attribute values computed according to the value of the attribute commonPointRule.

The operation evaluate as defined in this Annex represents a subset of the capability of the probing function evaluate as specified in clause 5.3. This version of the evaluate function is retained for backward compatibility.
NOTE Normally, the operation will return a single record of feature attribute values.

D.2.2.4 Associations

D.2.2.4.1 Coordinate Reference System

The association Coordinate Reference System links the CV_Coverage to the coordinate reference system to which the objects in its domain are referenced. The class CRS is specified in ISO 19111-1. The multiplicity of the CRS role in the Coordinate Reference System association is one. This means that a coverage with the same range but with its domain defined in a different coordinate reference system is a different coverage.

D.2.2.4.2 Domain

The association Domain links the CV_Coverage to the set of CV_DomainObjects in the domain.

NOTE The Domain and Range associations used here correspond to the CV_CoverageByDomainAndRange as specified in clause 5.8.

D.2.2.4.3 Range

The association Range links the CV_Coverage to the set of CV_AttributeValues in the range. The range of a CV_Coverage shall be a homogeneous collection of records. That is, the range has a constant dimension over the entire domain, and each field of the record provides a value of the same attribute type over the entire domain.

NOTE This document does not specify how the Domain and Range associations are to be implemented. The relevant data may be generated in real time, it may be held in persistent local storage, or it may be electronically accessible from remote locations.

D.2.3 CV_DomainObject

D.2.3.1 Class description

CV_DomainObject represents an element of the domain of the CV_Coverage. It is an aggregation of objects that may include any combination of GM_Objects (ISO 19107), TM_GeometricPrimitives (ISO 19108), or spatial or temporal objects defined in other standards, such as the CV_GridPoint defined in this document.

D.2.3.2 Associations

D.2.3.2.1 TemporalComposition

The association TemporalComposition associates a CV_DomainObject to the set of TM_GeometricPrimitives of which it is composed.

D.2.3.2.2 SpatialComposition

The association SpatialComposition associates a CV_DomainObject to the set of GM_Object of which it is composed.

D.2.4 CV_AttributeValues

D.2.4.1 Class description

CV_AttributeValues represents an element from the range of the CV_Coverage. It has one attribute values that takes on the value Record, where Record is a single or list of numeric arguments.
D.2.4.2 Attribute

D.2.4.2.1 values

The attribute values is a Record containing one value (single or set of arguments) for each attribute.

EXAMPLE A coverage with a single (scalar) value (such as elevation). A coverage with a series (array/tensor) of values all defined in the same way (such as brightness values in different parts of the electromagnetic spectrum).

D.2.4.3 Associations

D.2.4.3.1 CoverageFunction

The association CoverageFunction associates a CV_AttributeValues object to the set of CV_FunctionalElement of which it is an element.

D.2.5 CV_FunctionalElement

D.2.5.1 Class description

CV_FunctionalElement represents the result of the coverage function. It is the union of both the attribute value and the domain element geometry value pair that controls the location of the value. It has two attributes valueObject and geometryValuePair. Both of these attributes are represented as separate classes related by a Control relationship.

D.2.5.2 Attributes

D.2.5.2.1 valueObject

The attribute valueObject corresponds to the class CV_ValueObject.

D.2.5.2.2 geometryValuePair

The attribute geometryValuePair corresponds to the class CV_GeometryValuePair.

D.2.6 CV_ValueObject

D.2.6.1 Class description

CV_ValueObject provides a basis for interpolating feature attribute values within a continuous CV_Coverage.

CV_ValueObjects may be generated in the execution of an evaluate operation, and need not be persistent.

D.2.6.2 Attributes and operations

D.2.6.2.1 geometry

The attribute geometry: CV_DomainObject is a CV_DomainObject constructed from the CV_DomainObjects of the CV_GeometryValuePairs that are linked to this CV_ValueObject by the association Control.

D.2.6.2.2 interpolationParameters

The optional attribute interpolationParameters:Record shall hold the values of the parameters required to execute the interpolate operation, as specified by the interpolationParameterType attribute of CV_Coverage (cf. Annex B).
D.2.6.3 interpolate

The operation interpolate (DirectPosition): Record shall accept a DirectPosition as input and return the record of feature attribute values computed for that DirectPosition.

D.2.6.4 Associations

D.2.6.4.1 Control

The association Control shall link this CV_ValueObject to the set of CV_GeometryValuePairs that provide the basis for constructing the CV_ValueObject and for evaluating a continuous CV_Coverage at DirectPositions within this CV_ValueObject.

D.2.7 CV_GeometryValuePair

D.2.7.1 Class description

The class CV_GeometryValuePair describes an element of a set that defines the relationships of a discrete coverage. Each member of this class consists of two parts: a domain object from the domain of the coverage to which it belongs and a record of feature attribute values from the range of the coverage to which it belongs.

CV_GeometryValuePairs may be generated in the execution of an evaluate operation, and need not be persistent. CV_GeometryValuePair is subclassed to restrict the pairing of a feature attribute value record to a specific subtype of domain object.

D.2.7.2 Attributes

D.2.7.2.1 geometry

The attribute geometry: CV_DomainObject shall hold the CV_DomainObject that is a member of this CV_GeometryValuePair.

D.2.7.2.2 value

The attribute value: Record shall hold the record of feature attribute values associated with this CV_DomainObject.

D.2.8 CV_CommonPointRule

D.2.8.1 Class description

The code list CV_CommonPointRule is a list of codes that identify methods for handling cases where the DirectPosition input to the evaluate operation falls within two or more of the geometric objects. The interpretation of these rules differs between discrete and continuous coverages attributes. In the case of a discrete coverage range attribute, each CV_GeometryValuePair provides one value for each attribute. The rule is applied to the set of values associated with the set of CV_GeometryValuePairs that contain the DirectPosition. In the case of a continuous coverage, a value for each attribute shall be interpolated for each CV_ValueObject that contains the DirectPosition. The rule shall then be applied to the set of interpolated values for each attribute.

D.2.8.2 Code List

The code list CV_CommonPointRule takes on the values shown in Figure D.2 and given in table D.1.
Table D.1 — Semantics of CV_CommonPointRule

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning (given some position p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>average</td>
<td>Arithmetic average over the set returned by evaluate(p)</td>
</tr>
<tr>
<td>low</td>
<td>Minimum value of the set returned by evaluate(p)</td>
</tr>
<tr>
<td>high</td>
<td>Maximum value of the set returned by evaluate(p)</td>
</tr>
<tr>
<td>all</td>
<td>The complete set returned by evaluate(p)</td>
</tr>
<tr>
<td>start</td>
<td>Start Value of the second CV_ValueSegment (only applicable to Multi-Cuve Coverages)</td>
</tr>
<tr>
<td>end</td>
<td>End Value of the first CV_ValueSegment (only applicable to Multi-Cuve Coverages)</td>
</tr>
</tbody>
</table>

**Requirement D1:** For a given CV_Coverage,

— In case of no Common Point Rule provided and more than one range value existing for a Direct Position p, evaluate(p) shall return the set of all these values.

— In case of a discrete coverage with a Common Point Rule provided, evaluate( ) shall provide one range value for each Direct Position, selected from the set of available range values in accordance with the Common Point Rule value.

— In case of a continuous coverage (i.e., when at least one interpolation method is indicated in the coverage, see 5.6) with a Common Point Rule provided, a value for each attribute shall be obtained by applying one of the interpolation methods provided for each geometric object in the coverage that contains the Direct Position, and a range value shall be selected in accordance with the Common Point Rule value.

**Requirement D2:** The semantics of a CV_CommonPointRule shall be given by Figure D.2 and Table D.1, but may be extended with further values whose interpretation is implementation-defined.

**D.2.9 CV_CommonPointRule**

**D.2.9.1 Class description**

The code list CV_InterpolationMethod is a list of codes that identify interpolation methods that may be used for evaluating continuous coverages.
D.2.9.2 Code List

The code list CV_InterpolationMethod takes on the values shown in Figure B.1 and given in Table B.1 of Annex B.

D.3 Coverage types

D.3.1 Discrete and Continuous Coverages

The standard 19123:2005 made a major distinction between Discrete and Continuous coverages and organized the subtypes of coverage based on these types. This version of 19123-1 has generalized the concept. Any axis may be discrete or continuous. Since this is a generalization of the concept it is fully backward compatible with the older approach. Therefore this annex has simplified the description and no longer organizes coverages as Discrete or Continuous. The defined generic subtypes of CV_ValueObject and CV_GeometryValuePair are simply listed as subtypes. Each has a use in a particular type of coverage. This is shown below:

- CV_ValueCurve and CVValueSegment together with CV_PointValuePair and CV_CurveValuePair are used in a Segmented Curve Coverage
- CV_ThiessenValuePolygon together with CV_PointValuePair are used in a Theissen Polygon Coverage
- CV_ValueTriangle together with CV_PointValuePair are used in a TIN Triangle Coverage
- CV_ValueHexagon together with CV_GridPointValuePair are used in a Hexagon Grid coverage
- CV_GridValueCell together with CV_GridPointValuePair are used in a Grid coverage

D.3.2 Multi-Point Coverages

A point coverage is characterized by a finite domain consisting of points which may be regularly or irregularly distributed. A point coverage may be discrete in the domain; that is, valid only at the location of the points. Such a discrete point coverage provides a basis for continuous coverage functions through a control relationship, where the evaluation of the continuous coverage function is accomplished by interpolation between the points of the discrete point coverage. Most interpolation algorithms depend upon a structured pattern of spatial relationships between the points. This requires either that the points in the spatial domain of the discrete point coverage be arranged in a regular way, or that the spatial domain of the continuous coverage be partitioned in a regular way in relation to the points of the discrete point coverage. Grid coverages employ the first method; Thiessen polygon and TIN coverages employ the second.

EXAMPLE A set of hydrographic soundings is a discrete point coverage. Interpolation between the points establishes a bathymetric surface, which is a continuous coverage.

The classes CVPointValuePair and CV_GridPointValuePair provide values for Multi-point Coverages.

The requirements for a multi-point coverage are described in clause 6.

D.3.3 Multi-Curve Coverage

A curve coverage is characterized by a finite spatial domain consisting of curves. In a discrete curve coverage the curves represent features such as roads, railroads or streams. They may be elements of a network.

EXAMPLE A discrete curve coverage assigns a route number, a name, a pavement width and a pavement material type to each segment of a road system. The curve is discrete in the domain and each domain element is associated with a single value in the range.
A continuous curve coverage assigns a value to locations along a curve. This is a coverage with a one dimensional domain. An example is a measurement of pavement thickness along a road. Such a continuous coverage may be interpolated.

The classes CV_ValueCurve and CV_ValueSegment provide values for Multi-curve Coverages. They reference CV_PointValuePair for the geometry of the points at the ends of curve segments.

The requirements for a multi-point coverage are described in clause 8.

D.3.4 Multi-Surface Coverage

A Multi- surface Coverage is a coverage whose domain consists of a collection of surfaces. In most cases, the surfaces that constitute the domain of a coverage are mutually exclusive and exhaustively partition the extent of the coverage. Surfaces or their boundaries may be of any shape. The boundaries of component surfaces often correspond to natural phenomena and are highly irregular.

EXAMPLE A discrete multi-surface coverage that represents soil types typically has a spatial domain composed of surfaces with irregular boundaries.

Any set of polygons can be used as a spatial domain for a discrete Multi-surface Coverage. Spatial domains composed of congruent polygons are very common. Often, these domains are composed of congruent rectangles or regular hexagons. The geometry of such a tessellation may be described in terms of a quadrilateral grid or a hexagonal grid. The spatial domain of a discrete surface coverage may also consist of the triangles that compose a TIN, or the polygons of a Thiessen polygon network. Based on the control points, a surface coverage may be interpolated at locations on the surface. That is, the control points at the corners of a TIN triangle may be used to drive an interpolation function that allows one to determine a value at any location on the TIN value triangle. Equivalently this may be done for a Thiessen value polygon where interpolation is based on interpolation between the centres of the CV_ThiessenValuePolygons surrounding the input position, or a grid value cell, value hexagon or general mesh where other interpolation methods may be applied. The geometric value of a surface is defined by the class CV_SurfaceValuePair.

The requirements for a multi-point coverage are described in clause 9.

D.3.5 Multi-Solid Coverage

A Multi-solid Coverage is a coverage whose domain consists of a collection of solids. Solids or their boundaries may be of any shape. Generally, the solids that constitute the domain of a coverage are mutually exclusive and exhaustively partition the extent of the coverage, but this is not required.

EXAMPLE Buildings in an urban area could be represented as a set of unconnected solids each with attributes such as building name, address, floor space and number of occupants.

As in the case of surfaces, the spatial domain of a discrete solid coverage may be a regular or semiregular tessellation of the extent of the coverage. The tessellation can be defined in terms of a three-dimensional grid, where the set of grid cells is the spatial domain of the coverage.

Based on the control points, a solid coverage may be interpolated at locations within the solid. That is, the control points along the edges of the solid may be used to drive an interpolation function that allows one to determine a value at any location within the solid. Additional control points may also be defined within the solid to drive an interpolation. The geometric value of a solid is defined by the class CV_SolidValuePair.

The requirements for a multi-point coverage are described in clause 10.

D.3.6 Grid Coverage

A mesh is a network composed of two or more sets of curves in which the members of each set intersect the members of the other sets. A grid is a type of mesh which is regular in some algorithmically defined manner.
Grid coverages employ a systematic tessellation of the domain. The principal advantage of such tessellations is that they support a sequential enumeration of the elements of the domain.

The class CV_Grid describes the geometric characteristics of a quadrilateral grid. CV_GridValuesMatrix is a subclass of CV_Grid that ties feature attribute values to grid geometry. It holds a sequence of records associated with a sequencing rule that specifies an algorithm for assigning records of feature attribute values to grid points. CV_SequenceRule is a data type that contains information for mapping grid coordinates to a position within the sequence of records of feature attribute values.

The requirements for a multi-point coverage are described in Clause 7.

**D.4 Subclassing CV_GeometryValueObject**

**D.4.1 General**

A series of template generic data models are defined in this annex that correspond to the coverage models that were defined in 19123:2005. These new models are compliant data structures that can be interpreted through the interface structure defined in the revised 19123-1.

Figure D.4 illustrates the subtyping of CV_GeometryValueObject and the associated CV_GeometryValuePair object.

**D.4.2 CV_ValueCurve**

**D.4.2.1 Class description**

A CV_ValueCurve is composed of a Curve with additional information that supports the determination of feature attribute values at any position on that curve. CV_ValueCurves depend upon the arc-length parameterization operations defined for Curve in ISO 19107.

**D.4.2.2 Attributes and operations**

**D.4.2.3 geometry**

The attribute geometry: Curve shall be the Curve (per ISO 19107) that is the basis of this CV_ValueCurve.

**D.4.2.4 segment**

The operation segment (DirectPosition*, Distance*): Set<CV_ValueSegment> shall accept a DirectPosition as input and return the set of CV_ValueSegments nearest to that DirectPosition. This operation shall invoke the parameterForPosition operation defined for Curve (per ISO 19107) to obtain the Distance parameter corresponding to the input.

DirectPosition. The operation parameterForPosition returns the parameter value for the position on the Curve closest to the input DirectPosition. In certain cases, the parameterForPosition may return more than one parameter value. The operation segment will normally return a single CV_ValueSegment. There are three cases for which it could return multiple CV_ValueSegments:

a) The CV_ValueCurve is not simple. The position on the curve that is closest to the input DirectPosition is a point of self-intersection. The operation parameterForPoint returns two or more parameter values. In this case, the operation segment shall raise an exception.

b) There are two or more positions on the CV_ValueCurve that are at the same minimal distance from the input DirectPosition. The operation parameterForPoint returns two or more parameter values. In this case, the operation segment shall raise an exception.
D.4.3 CV_CurveValuePair

D.4.3.1 Class description

CV_CurveValuePair is the subtype of CV_GeometryValuePair that has Curve (per ISO 19107) as the value of its geometry attribute.

D.4.3.2 Attribute

D.4.3.2.1 geometry

The attribute geometry: Curve shall be the Curve (per ISO 19107) that is the basis of this CV_ValueCurve. A curve may either be expressed as a set of curves or as a set of CV_ValueSegment that are terminated by points at their ends.
D.4.4 CV_ValueSegment

D.4.4.1 Class description

A segment is a 1-dimensional geometric object that has a beginning (start) and end point. The limits of a CV_ValueSegment are specified by two values of the arc-length parameter of the Curve (per ISO 19107) underlying its parent CV_ValueCurve.

D.4.4.2 Attributes

D.4.4.3 startParameter

The attribute \textit{startParameter: Distance} is the value of the arc-length parameter of the parent curve at the start of this CV_ValueSegment.

D.4.4.3.1 endParameter

The attribute \textit{endParameter: Distance} is the value of the arc-length parameter of the parent curve at the end of this CV_ValueSegment.

D.4.4.4 Associations

D.4.4.4.1 Control

The association \textit{Ends} links the CV_ValueSegment to the CV_PointValuePair that provide control values for interpolation. Linear interpolation requires a minimum of two control values, usually those at the beginning and end of the CV_ValueSegment. Additional control values are required to support interpolation by higher order functions.

D.4.5 CV_PointValuePair

D.4.5.1 Class description

CV_PointValuePair is the subtype of CV_GeometryValuePair that has a Point (per ISO 19107) as the value of its geometry attribute.

D.4.5.2 Attribute

D.4.5.2.1 geometry

The attribute \textit{geometry: Point} shall hold the geometry of the CV_PointValuePair.

D.4.5.3 Class description

CV_ThiessenValuePolygon is a subclass of CV_ValueObject. Individual CV_ThiessenValuePolygons may be generated during the evaluation of a CV_ThiessenPolygonCoverage, and need not be persistent. A Thiessen polygon is a polygon whose boundaries define the area consisting of points that are closer to the center point than any other point in the defining set. The boundaries are the perpendicular bisectors of the lines between the points in the dual TIN. Also referred to as Voronoi Diagrams.

D.4.6 CV_ThiessenValuePolygon

D.4.6.1 Class description

CV_ThiessenValuePolygon is a subclass of CV_ValueObject. Individual CV_ThiessenValuePolygons may be generated during the evaluation of a CV_ThiessenPolygonCoverage, and need not be persistent. A Thiessen polygon is a polygon whose boundaries define the area consisting of points that are closer to the center point...
than any other point in the defining set. The boundaries are the perpendicular bisectors of the lines between the points in the dual TIN. Also referred to as Voronoi Diagrams.

D.4.6.2 Attributes

D.4.6.2.1 7.3.2 geometry

The attribute geometry: Polygon shall hold the geometry of the Thiessen polygon centred on the CV_PointValuePair identified by the association Control.

D.4.6.3 Associations

D.4.6.3.1 Control

The association Control links a CV_ThiessenValuePolygon to the CV_PointValuePair at its centre.

D.4.7 CV_ValueTriangle

D.4.7.1 Class description

CV_ValueTriangle is a subclass of CV_ValueObject that consists of three CV_PointValuePairs where the Points (per ISO 19107) are non-collinear. CV_ValueTriangles are used for interpolation of a coverage. A CV_ValueTriangle is defined by three non-collinear points and associated values used in interpolation of a coverage function. Usually associated with a TIN or a Triangulated Spline.

D.4.7.2 Attributes and operations

D.4.7.3 geometry

The attribute geometry:Triangle holds the Triangle (per ISO 19107) that defines the relative position of the three CV_PointValuePairs at its vertices.

D.4.7.4 point

The operation point:(DirectPosition):Sequence<Number> accepts a direct position inside a CV_ValueTriangle and returns the barycentric coordinates of the position as a sequence of numbers.

Constraint: {non-collinear: The three associated points in the associated CV_PointValuePair are not co-linear.}

D.4.7.5 Association

D.4.7.5.1 Control

The association Control shall link this CV_ValueTriangle to the CV_PointValuePair at its vertices.

D.4.8 CV_ValueHexagon

D.4.8.1 Class description

CV_ValueHexagon is a subclass of CV_ValueObject that describes a value object of hexagon shape.

D.4.8.2 Attribute

D.4.8.3 geometry

The attribute geometry:Polygon (per ISO 19107) shall hold the geometry of the CV_ValueHexagon centred on the CV_GridPointValuePair identified by the association Control.
D.4.8.4 Association

D.4.8.4.1 Control

The association Control links a CV_ValueHexagon to the CV_GridPointValuePair at its centre.

D.4.9 8.12 CV_GridValueCell

D.4.9.1 8.12.1 Class description

CV_GridValueCell is a subclass of CV_ValueObject that supports interpolation within a CV_GeneralGridCoverage. A CV_GridValueCell is a collection of CV_GridPointValuePair with a geometric structure defined by a CV_GridCell.

D.4.9.2 Attribute

D.4.9.2.1 geometry

The attribute geometry:CV_GridCell holds the CV_GridCell that defines the structure of the CV_GridPointValuePair that support the interpolation of a feature attribute value at a DirectPosition within the CV_GridCell.

D.4.9.3 Association

D.4.9.3.1 Control

The association Control links a CV_GridValueCell to the CV_GridPointValuePair at its corners.

D.4.10 CV_GridPointValuePair

D.4.10.1 Class description

CV_GridPointValuePair is a subclass of CV_GeometryValuePair composed of a CV_GridPoint and a point attribute value Record.

D.4.10.2 Attribute

D.4.10.2.1 point

The attribute point: CV_GridPoint shall be the geometry member of the CV_GridPointValuePair. It shall be one of the CV_GridPoints linked to the CV_ValuesMatrix inherited from CV_Grid.

D.4.11 CV_GridCell

D.4.11.1 Class description

A CV_GridCell is delineated by the grid lines of CV_Grid. Its corners are associated with the CV_GridPoints at the intersections of the grid lines that bound it.

D.4.12 CV_CurveValuePair

D.4.12.1 Class description

CV_CurveValuePair is the subtype of CV_GeometryValuePair that has Curve (per ISO 19107) as the value of its geometry attribute.
D.4.12.2 Attribute

D.4.12.2.1 geometry

The attribute geometry: Curve shall be the geometry member of the CV_CurveValuePair. It shall be one of the CV_GridPoints linked to the CV_ValuesMatrix inherited from CV_Grid.

D.4.13 CV_SurfaceValuePair

D.4.13.1 Class description

CV_SurfaceValuePair is the subtype of CV_GeometryValuePair that has Surface (per ISO 19107) as the value of its geometry attribute. Used in the definition of discrete (step function) surface coverage functions.

D.4.13.2 Attribute

D.4.13.2.1 geometry

The attribute geometry: Surface shall hold the geometry of the CV_SurfaceValuePair.

D.4.14 CV_SolidValuePair

D.4.14.1 Class description

CV_SolidValuePair is the subtype of CV_GeometryValuePair that has a Solid (per ISO 19107) as the value of its geometry attribute.

D.4.14.2 Attribute

D.4.14.2.1 geometry

The attribute geometry: Solid shall hold the geometry of the CV_SolidValuePair.

D.4.15 CV_Grid

D.4.15.1 Class description

The class CV_Grid contains the geometric characteristics of a quadrilateral grid.

D.4.15.2 Attributes

D.4.15.2.1 dimension

The attribute dimension: Integer shall identify the dimensionality of the grid.

D.4.15.2.2 axisNames

The attribute axisNames: CharacterString(Sequence) shall list the names of the grid axes.

D.4.15.2.3 extent

The optional attribute extent: CV_GridEnvelope shall specify the limits of a section of the grid.
D.4.16 GridEnvelope

D.4.16.1 Class description

CV_GridEnvelope is a data type that provides the grid coordinate values for the diametrically opposed corners of the CV_Grid. It has two attributes.

D.4.16.2 Attributes

D.4.16.2.1 low

The attribute low: CV_GridCoordinate shall be the minimal coordinate values for all grid points within the CV_Grid.

D.4.16.2.2 high

The attribute high: CV_GridCoordinate shall be the maximal coordinate values for all grid points within the CV_Grid.

D.4.17 CV_GridPoint

D.4.17.1 Class description

CV_GridPoint is the class that represents the intersections of the grid lines.

D.4.17.2 Attribute

D.4.17.2.1 gridCoord

The attribute gridCoord : CV_GridCoordinate holds the set of grid coordinates that specifies the location of the CV_GridPoint within the CV_Grid.

D.4.18 CV_GridCoordinate

D.4.18.1 Class description

CV_GridCoordinate is a data type for holding the grid coordinates of a CV_GridPoint.

D.4.18.2 Attribute

D.4.18.3 coordValues

The attribute coordValues: Integer(sequence) holds one integer value for each dimension of the grid. The ordering of these coordinate values shall be the same as that of the elements of CV_Grid.axisNames. The value of a single coordinate shall be the number of offsets from the origin of the grid in the direction of a specific axis.

D.4.19 CV_GridValuesMatrix

D.4.19.1 Class description

CV_GridValuesMatrix is a subclass of CV_Grid that ties feature attribute values to grid geometry. It has three attributes. It holds a sequence of records associated with a sequencing rule that specifies an algorithm for assigning records of feature attribute values to grid points. The geometry represented by the various offset vectors is in the image plane of the grid. For example, for orthorectified grids, these vectors are in a spatial reference system.
D.4.19.2 Attributes

D.4.19.3 values

The attribute values: Record(sequence) shall be a sequence of N feature attribute value records where N is the number of grid points within the section of the grid specified by the attribute extent from CV_Grid.

D.4.19.4 sequencingRule

The attribute sequencingRule: CV_SequenceRule shall describe how the grid points are ordered for association to the elements of the sequence values. Default is row major.

D.4.19.5 startSequence

The attribute startSequence: CV_GridCoordinate shall identify the grid point to be associated with the first record in the values sequence.

D.4.20 CV_GridCell

D.4.20.1 Class description

A CV_GridCell is delineated by the grid lines of CV_Grid. Its corners are associated with the CV_GridPoints at the intersections of the grid lines that bound it.

D.4.21 CV_SequenceRule

D.4.21.1 Class description

CV_SequenceRule is a data type that contains information for mapping grid coordinates to a position within the sequence of records of feature attribute values.

D.4.21.2 Attributes

D.4.21.3 type

The attribute type: CV_SequenceType identifies the type of sequencing method that shall be used. The default value shall be “linear”.

D.4.21.4 scanDirection

The attribute scanDirection: CharacterString(sequence) is a list of signed axisNames that indicates the order in which grid points shall be mapped to position within the sequence of records of feature attribute values. An additional element may be included in the list to allow for interleaving of feature attribute values.

D.4.22 CV_SequenceType

CV_SequenceType is a code list that identifies methods for sequential enumeration of the grid points. Methods for sequential enumeration are described in Annex C.
Annex E
(informative)

Backward Compatibility

E.1 General

This revision of the ISO 19123-1 standard is completely backward compatible with the previous version ISO 19123 in that all ISO 19123 concepts can be expressed by ISO 19123-1; in particular, Annex E retains the data-centric view of 19123.

The standard has been extended and clarified and some fields have been made optional and some items have been left out to achieve the appropriate level of abstraction; however, the elements of the standard that were defined in the previous version and are present in this version remain technically unchanged.

Any profile, application schema or other standard that made reference to the previous version of this standard, ISO 19123, remains valid as long as the versions of other standards normatively referenced in this document (such as ISO 19107, ISO 19109, and ISO 19111) remain valid.

E.2 Changes

The changes that have been made to the previous version of the standard are listed below.

- The standard has been renamed to be “Part 1: Fundamentals”, since a new “Part 2: Coverage Implementation Schema” has been published.

- The scope has been revised to include Mesh, and the text has been simplified.

- The approach to standardization taken in the document has been changed. This version of 19123-1 defines an interface for coverages through which many different implementation structures may be referenced in a compatible manner. The previous version of the standard 19123 defined a single generic data structure for coverages. The data structure defined in 19123 remains valid as one of the many possible data structures that may be accessed through the interface. As such this data structure is summarized in normative Annex D. This allows for backward compatibility since implementations that referenced the foundation classes defined in 19123 may still reference these same classes.

- The concept of Discrete and Continuous coverages has been generalized to address axes. That is, any axis (domain of range axis) may be discrete or continuous. This has greatly simplified the structure of the standard. Since this is a generalization of the previous concept it is backward compatible.

- The informative Annex on “UML notation” has been deleted since this material is now described in ISO 19103.

- The UML diagrams have been redrawn for clarity and to follow the new conventions established in TC211. Some of the errors in the model consisted of duplicate definitions of attributes and operations in subclasses. This was a result of the older version of the UML modelling tool used when this standard was first developed. Redundant attributes and operations have been removed except for those places where an attribute is deliberately overwritten to establish a default value.
Some errors in 19123 have been corrected. The original model was correct but original documentation did not align with the original model.

The bibliography has been revised to include additional references and has been reformatted.

ISO 19123-1 relies on the current versions of ISO 19111 and ISO 19107, which have been significantly updated as compared to the versions referenced in ISO 19123.

Definition of interpolation now is based on the interpolation definition of ISO 19107 in order to avoid duplicate and diverging definitions.

The annex on “Sequential enumeration” has been made normative.

All operations except evaluate and interpolate are deleted.

Information on an Image Coordinate Reference System from ISO 19111 has been removed from the revised version of 19111 and material added to Clause 7 of this document.

---

![Diagram](https://via.placeholder.com/150)

**Figure E.1 — Mapping between coverage types**
Bibliography


[3] GOODCHILD, M. F. and GRANDFIELD, A. W., Optimizing Raster Storage: An Examination of Four Alternatives, IAAuto-Carto 6, 1983


[6] International Association of Oil & Gas Producers, EPSG Data Set. Available at: http://www.epsg.org/


