

**Geografisk information – Modell för att beskriva
rumsliga aspekter**

Geographic information – Spatial schema

Den internationella standarden ISO 19107:2003 gäller som svensk standard. Detta dokument innehåller den officiella engelska versionen av ISO 19107:2003.

The International Standard ISO 19107:2003 has the status of a Swedish Standard. This document contains the official English version of ISO 19107:2003.

Dokumentet består av 175 sidor.

Upplysningar om **sakinnehållet** i standarden lämnas av SIS, Swedish Standards Institute, tel 08 - 555 520 00.

Standarder kan beställas hos SIS Förlag AB som även lämnar **allmänna upplysningar** om svensk och utländsk standard.

Postadress: SIS Förlag AB, 118 80 STOCKHOLM
Telefon: 08 - 555 523 10. *Telefax:* 08 - 555 523 11
E-post: sis.sales@sis.se. *Internet:* www.sis.se

Contents

Page

Foreword.....	viii
Introduction	ix
1 Scope	1
2 Conformance	1
2.1 Overview	1
2.2 Conformance classes	3
3 Normative references	4
4 Terms and definitions	4
5 Symbols, notation and abbreviated terms	14
5.1 Presentation and notation	14
5.1.1 Unified Modeling Language (UML) concepts	14
5.1.2 Attributes, operations, and associations	14
5.1.3 Stereotypes	17
5.1.4 Data types and collection types	18
5.1.5 Strong substitutability	19
5.2 Organization	20
5.3 Abbreviated terms	22
6 Geometry packages	22
6.1 Semantics	22
6.2 Geometry root package	24
6.2.1 Semantics	24
6.2.2 GM_Object	25
6.3 Geometric primitive package	32
6.3.1 Semantics	32
6.3.2 GM_Boundary	33
6.3.3 GM_ComplexBoundary	34
6.3.4 GM_PrimitiveBoundary	34
6.3.5 GM_CurveBoundary	34
6.3.6 GM_Ring	34
6.3.7 GM_SurfaceBoundary	34
6.3.8 GM_Shell	35
6.3.9 GM_SolidBoundary	35
6.3.10 GM_Primitive	35
6.3.11 GM_Point	38
6.3.12 Bearing	39
6.3.13 GM_OrientablePrimitive	40
6.3.14 GM_OrientableCurve	42
6.3.15 GM_OrientableSurface	42
6.3.16 GM_Curve	43
6.3.17 GM_Surface	44
6.3.18 GM_Solid	46
6.4 Coordinate geometry package	47
6.4.1 DirectPosition	47
6.4.2 GM_PointRef	48
6.4.3 GM_Envelope	48
6.4.4 TransfiniteSet<DirectPosition>	49
6.4.5 GM_Position	49
6.4.6 GM_PointArray, GMPointGrid	49
6.4.7 GM_GenericCurve	49
6.4.8 GM_CurveInterpolation	53

ISO 19107:2003(E)

6.4.9	GM_CurveSegment	54
6.4.10	GM_LineString	55
6.4.11	GM_LineSegment	56
6.4.12	GM_GeodesicString	57
6.4.13	GM_Geodesic.....	58
6.4.14	GM_ArcString	58
6.4.15	GM_Arc.....	60
6.4.16	GM_Circle.....	62
6.4.17	GM_ArcStringByBulge.....	62
6.4.18	GM_ArcByBulge	63
6.4.19	GM_Conic.....	64
6.4.20	GM_Placement.....	66
6.4.21	GM_AffinePlacement	67
6.4.22	GM_Clothoid	67
6.4.23	GM_OffsetCurve	68
6.4.24	GM_Knot.....	70
6.4.25	GM_KnotType	71
6.4.26	GM_SplineCurve.....	71
6.4.27	GM_PolynomialSpline.....	71
6.4.28	GM_CubicSpline	72
6.4.29	GM_SplineCurveForm.....	73
6.4.30	GM_BSplineCurve	73
6.4.31	GM_Bezier	74
6.4.32	GM_SurfaceInterpolation.....	75
6.4.33	GM_GenericSurface	75
6.4.34	GM_SurfacePatch.....	77
6.4.35	GM_PolyhedralSurface	78
6.4.36	GM_Polygon.....	78
6.4.37	GM_TriangulatedSurface	80
6.4.38	GM_Triangle.....	80
6.4.39	GM_Tin	81
6.4.40	GM_ParametricCurveSurface.....	82
6.4.41	GM_GriddedSurface.....	85
6.4.42	GM_Cone.....	86
6.4.43	GM_Cylinder	86
6.4.44	GM_Sphere.....	86
6.4.45	GM_BilinearGrid	87
6.4.46	GM_BicubicGrid	87
6.4.47	GM_BSplineSurfaceForm	87
6.4.48	GM_BSplineSurface	88
6.5	Geometric aggregate package	89
6.5.7	Semantics.....	89
6.5.8	GM_Aggregate	89
6.5.9	GM_MultiPrimitive	89
6.5.10	GM_MultiPoint	90
6.5.11	GM_MultiCurve	91
6.5.12	GM_MultiSurface	91
6.5.13	GM_MultiSolid.....	91
6.6	Geometric complex package.....	92
6.6.7	Semantics.....	92
6.6.8	GM_Complex.....	93
6.6.9	GM_Composite	94
6.6.10	GM_CompositePoint	95
6.6.11	GM_CompositeCurve.....	96
6.6.12	GM_CompositeSurface	97
6.6.13	GM_CompositeSolid	97
7	Topology packages.....	98
7.4	Semantics.....	98
7.5	Topology root package.....	100

7.5.1	Semantics	100
7.5.2	TP_Object.....	101
7.6	Topological primitive package	105
7.6.1	Semantics	105
7.6.2	TP_Boundary.....	105
7.6.3	TP_ComplexBoundary.....	105
7.6.4	TP_PrimitiveBoundary.....	105
7.6.5	TP_EdgeBoundary	106
7.6.6	TP_FaceBoundary.....	107
7.6.7	TP_SolidBoundary	107
7.6.8	TP_Ring.....	107
7.6.9	TP_Shell	107
7.6.10	TP_Primitive	108
7.6.11	TP_DirectedTopo	109
7.6.12	TP_Node.....	112
7.6.13	TP_DirectedNode	113
7.6.14	TP_Edge	114
7.6.15	TP_DirectedEdge	115
7.6.16	TP_Face.....	115
7.6.17	TP_DirectedFace	117
7.6.18	TP_Solid	117
7.6.19	TP_DirectedSolid	118
7.6.20	TP_Expression	118
7.7	Topological complex package.....	121
7.7.1	Semantics	121
7.7.2	TP_Complex	121
8	Derived topological relations.....	123
8.1	Introduction	123
8.2	Boolean or set operators.....	124
8.2.1	Form of the Boolean operators	124
8.2.2	Boolean Relate	124
8.2.3	Relation to set operations	125
8.3	Egenhofer operators.....	125
8.3.1	Form of the Egenhofer operators	125
8.3.2	Egenhofer relate	125
8.3.3	Relation to set operations	126
8.4	Full topological operators	126
8.4.1	Form of the full topological operators	126
8.4.2	Full topological relate	126
8.5	Combinations	126
Annex A	(normative) Abstract test suite.....	127
A.1	Geometric primitives	127
A.2	Geometric complexes.....	130
A.3	Topological complexes	132
A.4	Topological complexes with geometric realization.....	134
A.5	Boolean operators	136
Annex B	(informative) Conceptual organization of terms and definitions	138
B.1	Introduction	138
B.2	General terms	138
B.3	Collections and related terms.....	139
B.4	Modelling terms.....	139
B.5	Positioning terms.....	140
B.6	Geometric terms.....	140
B.7	Topological terms	143
B.8	Relationship of geometric and topological complexes	146
Annex C	(informative) Examples of spatial schema concepts	148
C.1	Geometry.....	148

ISO 19107:2003(E)

Annex D (informative) Examples for application schemata 154
D.1 Introduction..... 154
D.2 Simple Topology..... 154
D.3 Feature Topology 158
D.4 MiniTopo..... 159
Bibliography..... 165

Figures

Figure 1 — UML example association 16
 Figure 2 — UML example package dependency 20
 Figure 3 — Normative clause as UML package dependencies 21
 Figure 4 — Geometry package: Class content and internal dependencies 23
 Figure 5 — Geometry basic classes with specialization relations 24
 Figure 6 — GM_Object..... 26
 Figure 7 — GM_Boundary 33
 Figure 8 — GM_Primitive 36
 Figure 9 — GM_Point..... 38
 Figure 10 — GM_OrientablePrimitive 41
 Figure 11 — GM_Curve 43
 Figure 12 — GM_Surface..... 45
 Figure 13 — GM_Solid..... 46
 Figure 14 — DirectPosition 48
 Figure 15 — Curve segment classes 50
 Figure 16 — Linear, arc and geodesic interpolation 56
 Figure 17 — Arcs..... 59
 Figure 18 — Conics and placements 65
 Figure 19 — Spline and specialty curves 69
 Figure 20 — Surface patches..... 76
 Figure 21 — Polygonal surface 79
 Figure 22 — TIN construction 81
 Figure 23 — GM_ParametricCurveSurface and its subtypes 83
 Figure 24 — GM_Aggregate 90
 Figure 25 — GM_Complex..... 94
 Figure 26 — GM_Composite..... 95
 Figure 27 — GM_CompositePoint 96
 Figure 28 — GM_CompositeCurve 96
 Figure 29 — GM_CompositeSurface 97
 Figure 30 — GM_CompositeSolid..... 98
 Figure 31 — Topology packages, class content and internal dependencies 99
 Figure 32 — Topological class diagram 100
 Figure 33 — Relation between geometry and topology 101
 Figure 34 — TP_Object..... 102
 Figure 35 — Boundary and coboundary operation represented as associations 103
 Figure 36 — Important classes in topology 104
 Figure 37 — Boundary relation data types..... 106
 Figure 38 — TP_Primitive 108
 Figure 39 — TP_DirectedTopo subclasses..... 110
 Figure 40 — TP_DirectedTopo 110

Figure 41 — TP_Node 113

Figure 42 — TP_Edge 114

Figure 43 — TP_Face 116

Figure 44 — TP_Solid 117

Figure 45 — TP_Expression 119

Figure 46 — TP_Complex 122

Figure C.1 — A data set composed of the GM_Primitives 149

Figure C.2 — Simple cartographic representation of sample data 151

Figure C.3 — A 3D Geometric object with labeled coordinates 152

Figure C.4 — Surface example 153

Figure D.1 — Packages and classes for simple topology 155

Figure D.2 — Topology and geometry classes in simple topology 156

Figure D.3 — Feature components in simple topology 157

Figure D.4 — Theme based feature topology 159

Figure D.5 — Geometric example of MiniTopo topology structure 160

Figure D.6 — MiniTopo 161

Figure D.7 — Classic MiniTopo record illustration 163

Tables

Table 1 — Conformance classes for geometric primitives 3

Table 2 — Conformance classes for geometric complexes 3

Table 3 — Conformance classes for topological complexes 3

Table 4 — Conformance classes for topological complexes with geometric realizations 3

Table 5 — Conformance classes for Boolean operators 3

Table 6 — Package and classes 21

Table 7 — Various types of parametric curve surfaces 84

Table 8 — Meaning of Boolean intersection pattern matrix 124

Table 9 — Meaning of Egenhofer intersection pattern matrix 125

Table 10 — Meaning of full topological intersection pattern matrix 126

Table D.1 — Correspondence between original MiniTopo pointers and the current model 164

ISO 19107:2003(E)

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 19107 was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics*.

Introduction

This International Standard provides conceptual schemas for describing and manipulating the spatial characteristics of geographic features. Standardization in this area will be the cornerstone for other geographic information standards.

A feature is an abstraction of a real world phenomenon; it is a geographic feature if it is associated with a location relative to the Earth. Vector data consists of geometric and topological primitives used, separately or in combination, to construct objects that express the spatial characteristics of geographic features. Raster data is based on the division of the extent covered into small units according to a tessellation of the space and the assignment to each unit of an attribute value. This International Standard deals only with vector data.

In the model defined in this International Standard, spatial characteristics are described by one or more spatial attributes whose value is given by a geometric object (GM_Object) or a topological object (TP_Object). Geometry provides the means for the quantitative description, by means of coordinates and mathematical functions, of the spatial characteristics of features, including dimension, position, size, shape, and orientation. The mathematical functions used for describing the geometry of an object depend on the type of coordinate reference system used to define the spatial position. Geometry is the only aspect of geographic information that changes when the information is transformed from one geodetic reference system or coordinate system to another.

Topology deals with the characteristics of geometric figures that remain invariant if the space is deformed elastically and continuously — for example, when geographic data is transformed from one coordinate system to another. Within the context of geographic information, topology is commonly used to describe the connectivity of an n -dimensional graph, a property that is invariant under continuous transformation of the graph. Computational topology provides information about the connectivity of geometric primitives that can be derived from the underlying geometry.

Spatial operators are functions and procedures that use, query, create, modify, or delete spatial objects. This International Standard defines the taxonomy of these operators in order to create a standard for their definition and implementation. The goals are to:

- a) Define spatial operators unambiguously, so that diverse implementations can be assured to yield comparable results within known limitations of accuracy and resolution.
- b) Use these definitions to define a set of standard operations that will form the basis of compliant systems, and, thus act as a test-bed for implementers and a benchmark set for validation of compliance.
- c) Define an operator algebra that will allow combinations of the base operators to be used predictably in the query and manipulation of geographic data.

Standardized conceptual schemas for spatial characteristics will increase the ability to share geographic information among applications. These schemas will be used by geographic information system and software developers and users of geographic information to provide consistently understandable spatial data structures.

Geographic information — Spatial schema

1 Scope

This International Standard specifies conceptual schemas for describing the spatial characteristics of geographic features, and a set of spatial operations consistent with these schemas. It treats vector geometry and topology up to three dimensions. It defines standard spatial operations for use in access, query, management, processing, and data exchange of geographic information for spatial (geometric and topological) objects of up to three topological dimensions embedded in coordinate spaces of up to three axes.

2 Conformance

2.1 Overview

Clauses 6 and 7 of this International Standard use the Unified Modeling Language (UML) to present conceptual schemas for describing the spatial characteristics of geographic features. These schemas define conceptual classes that shall be used in application schemas, profiles and implementation specifications. The document concerns ONLY externally visible interfaces and places no restriction on the underlying implementations other than what is needed to satisfy the interface specifications in the actual situation such as:

- Interfaces to software services using techniques such as COM or CORBA
- Interfaces to databases using techniques such as SQL
- Data interchange using encoding as defined in ISO 19118.

Few applications will require the full range of capabilities described by this conceptual schema. This clause, therefore, defines a set of conformance classes that will support applications whose requirements range from the minimum necessary to define data structures to full object implementation. This flexibility is controlled by a set of UML types that can be implemented in a variety of manners. Implementations that define full object functionality must implement all operations defined by the types of the chosen conformance class, as is common for UML designed object implementations. Implementations that choose to depend on external “free functions” for some or all operations, or forgo them altogether, need not support all operation, but shall always support a data type sufficient to record the state of each of the chosen UML type as defined by its member variables. Common names for “metaphorically identical” but technically different entities are acceptable. The UML model in this International Standard defines abstract types, application schemas define conceptual classes, various software systems define implementation classes or data structures, and the XML from the encoding standard (ISO 19118) defines entity tags. All of these reference the same information content. There is no difficulty in allowing the use of the same name to represent the same information content even though at a deeper level there are significant technical differences in the digital entities being implemented. This “allows” types defined in the UML model to be used directly in application schemas.

There are 39 conformance options for application schemas that define types for the instantiation of geometric or topological objects. They are differentiated on the basis of three criteria.

The first two criteria (complexity and dimensionality) determine the types defined in this schema that shall be implemented according to an application schema that conforms to a given conformance option. In defining the dimensionality of object types to be implemented, the application schema will be required to specify which of

ISO 19107:2003(E)

the interpolation types for curves or surfaces they wish to implement. Curve implementations, for those application schemas including 1-dimensional objects, shall always include a “linear” interpolation technique. Application schema including 1-dimensional objects should always include a mechanism to approximate any curve as a line string to allow for transfer of data into simpler schema where needed. Surface implementations, for those application schemas including 2-dimensional objects, shall always include a “planar” interpolation technique. Application schema should always include a mechanism to approximate any surface as collections of planar surface patches to allow for transfer of data into simpler schema where needed. Additional curve and surface interpolation mechanism are optional, but if implemented, they shall follow the definition included in this International Standard.

The third criterion (functional complexity) determines the member elements (attributes, association roles and operations) of those types that shall be implemented. The most limited of such schema would define only data types, and may be used in the transfer of data or the passing of operational parameters to service providers.

The first criterion is level of data complexity. Four levels are identified:

- Geometric primitives
- Geometric complexes
- Topological complexes
- Topological complexes with geometric realization

NOTE Schemas for what is commonly called “spaghetti” data use only unstructured collections of geometric primitives. If single definitions of each component of geometry are required, then geometric complexes are introduced into the schema. Primitives within the same geometric complex share only boundaries. If the schema requires explicit topological information then the geometric complex is expanded to include the structure of a topological complex. The types of object included in a complex are controlled by the dimension of that complex. What is commonly called “chain-node” topology is a 1-dimensional topological complex. What is commonly called “full topology” in a cartographic 2D environment is a 2-dimensional topological complex realized by geometric objects in a 2D coordinate system.

The second criterion is dimensionality. There are four levels for simple geometry:

- 0-dimensional objects
- 0- and 1-dimensional objects
- 0-, 1-, and 2-dimensional objects
- 0-, 1-, 2- and 3-dimensional objects

However, 0-dimensional complexes provide no useful information beyond that provided by 0-dimensional geometric primitives, so conformance classes are only defined for complexes of 1-, 2-, and 3-dimensions.

The third criterion is level of functional complexity. There are three levels.

- Data types only
- Simple operations
- Complete operations

Clause 8 of this International Standard defines three groups of Boolean operators that may be used to derive topological relations between geometric and topological objects. This International Standard defines four conformance classes for application schemas that implement these operators.

2.2 Conformance classes

To conform to this International Standard, an implementation shall satisfy the requirements of the Abstract test suite (ATS) in Annex A for a specified conformance class. Table 1 through Table 5 identify the clauses of the ATS that apply for each conformance class.

Table 1 — Conformance classes for geometric primitives

Dimension	Data Types	Simple Operations	Complete Operations
0	A.1.1.1	A.1.2.1	A.1.3.1
1	A.1.1.2	A.1.2.2	A.1.3.2
2	A.1.1.3	A.1.2.3	A.1.3.3
3	A.1.1.4	A.1.2.4	A.1.3.4

Table 2 — Conformance classes for geometric complexes

Dimension	Data Types	Simple Operations	Complete Operations
1	A.2.1.1	A.2.2.1	A.2.3.1
2	A.2.1.2	A.2.2.2	A.2.3.2
3	A.2.1.3	A.2.2.3	A.2.3.3

Table 3 — Conformance classes for topological complexes

Dimension	Data Types	Simple Operations	Complete Operations
1	A.3.1.1	A.3.2.1	A.3.3.1
2	A.3.1.2	A.3.2.2	A.3.3.2
3	A.3.1.3	A.3.2.3	A.3.3.3

Table 4 — Conformance classes for topological complexes with geometric realizations

Dimension	Data Types	Simple Operations	Complete Operations
1	A.4.1.1	A.4.2.1	A.4.3.1
2	A.4.1.2	A.4.2.2	A.4.3.2
3	A.4.1.3	A.4.2.3	A.4.3.3

Table 5 — Conformance classes for Boolean operators

Set operators	A.5.1
Egenhofer operators	A.5.2
Full topological operators	A.5.3
All operators	A.5.4

ISO 19107:2003(E)

3 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 19109:—¹⁾, *Geographic information — Rules for application schema*

ISO 19111:—¹⁾, *Geographic information — Spatial referencing by coordinates*

ISO/IEC 11404:1996, *Information technology — Programming languages, their environments and system software interfaces — Language-independent datatypes*

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply. The terms are listed alphabetically in this clause. In Annex B they are organized by their conceptual relationships.

4.1

application

manipulation and processing of data in support of user requirements [ISO 19101]

4.2

application schema

conceptual schema for data required by one or more **applications** [ISO 19101]

4.3

bag

finite, unordered collection of related items (**objects** or values) that may be repeated

NOTE Logically, a bag is a set of pairs <item, count>.

4.4

boundary

set that represents the limit of an entity

NOTE **Boundary** is most commonly used in the context of geometry, where the set is a collection of points or a collection of objects that represent those points. In other arenas, the term is used metaphorically to describe the transition between an entity and the rest of its domain of discourse.

4.5

buffer

geometric object that contains all **direct positions** whose distance from a specified **geometric object** is less than or equal to a given distance

4.6

circular sequence

sequence which has no logical beginning and is therefore equivalent to any circular shift of itself; hence the last item in the sequence is considered to precede the first item in the sequence

4.7

class

description of a **set** of **objects** that share the same attributes, operations, methods, relationships, and semantics [ISO/TS 19103]

1) To be published.

NOTE A class may use a **set of interfaces** to specify collections of **operations** it provides to its environment. The term was first used in this way in the general theory of object oriented programming, and later adopted for use in this same sense in UML.

4.8

closure

union of the **interior** and **boundary** of a **topological** or **geometric object**

4.9

coboundary

set of topological primitives of higher topological dimension associated with a particular **topological object**, such that this **topological object** is in each of their **boundaries**

NOTE If a node is on the boundary of an edge, that edge is on the coboundary of that node. Any orientation parameter associated to one of these relations would also be associated to the other. So that if the node is the end node of the edge (defined as the end of the positive directed edge), then the positive orientation of the node (defined as the positive directed node) would have the edge on its coboundary, see Figure 35.

4.10

composite curve

sequence of curves such that each curve (except the first) starts at the end point of the previous curve in the **sequence**

NOTE A composite curve, as a set of **direct positions**, has all the properties of a curve.

4.11

composite solid

connected **set of solids** adjoining one another along shared **boundary surfaces**

NOTE A composite solid, as a set of **direct positions**, has all the properties of a solid.

4.12

composite surface

connected **set of surfaces** adjoining one another along shared **boundary curves**

NOTE A composite surface, as a set of **direct positions**, has all the properties of a surface.

4.13

computational geometry

manipulation of and calculations with geometric representations for the implementation of **geometric operations**

EXAMPLE Computational geometry operations include testing for geometric inclusion or intersection, the calculation of **convex hulls** or **buffer zones**, or the finding of shortest distances between **geometric objects**.

4.14

computational topology

topological concepts, structures and algebra that aid, enhance or define **operations** on **topological objects** usually performed in **computational geometry**

4.15

connected

property of a **geometric object** implying that any two **direct positions** on the object can be placed on a **curve** that remains totally within the object

NOTE A topological object is connected if and only if all its **geometric realizations** are connected. This is not included as a definition because it follows from a theorem of topology.

4.16

connected node

node that starts or ends one or more **edges**

ISO 19107:2003(E)

4.17

convex hull

smallest **convex set** containing a given **geometric object** [Dictionary of Computing [7]]

NOTE "Smallest" is the set theoretic smallest, not an indication of a measurement. The definition can be rewritten as "the intersection of all convex sets that contain the geometric object".

4.18

convex set

geometric set in which any **direct position** on the straight-line segment joining any two **direct positions** in the geometric set is also contained in the **geometric set** [Dictionary of Computing [7]]

NOTE Convex sets are "simply connected", meaning that they have no interior holes, and can normally be considered topologically isomorphic to a Euclidean ball of the appropriate dimension. So the surface of a sphere can be considered to be geodesically convex.

4.19

coordinate

one of a **sequence** of N-numbers designating the position of a **point** in N-dimensional space [ISO 19111]

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

4.20

coordinate dimension

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

4.21

coordinate reference system

coordinate system that is related to the real world by a datum [ISO 19111]

4.22

coordinate system

set of mathematical rules for specifying how **coordinates** are to be assigned to **points** [ISO 19111]

4.23

curve

1-dimensional **geometric primitive**, representing the continuous image of a line

NOTE The **boundary** of a **curve** is the **set of points** at either end of the **curve**. If the curve is a cycle, the two ends are identical, and the curve (if topologically closed) is considered to not have a boundary. The first **point** is called the **start point**, and the last is the **end point**. Connectivity of the curve is guaranteed by the "continuous image of a line" clause. A topological theorem states that a continuous image of a connected set is connected.

4.24

curve segment

1-dimensional **geometric object** used to represent a continuous component of a **curve** using homogeneous interpolation and definition methods

NOTE The **geometric set** represented by a single curve segment is equivalent to a **curve**.

4.25

cycle

⟨geometry⟩ **spatial object** without a **boundary**

NOTE Cycles are used to describe boundary components (see **shell**, **ring**). A cycle has no boundary because it closes on itself, but it is bounded (i.e., it does not have infinite extent). A circle or a sphere, for example, has no boundary, but is bounded.

4.26

direct position

position described by a single set of **coordinates** within a **coordinate reference system**

4.27**directed edge**

directed topological object that represents an association between an **edge** and one of its orientations

NOTE A directed edge that is in agreement with the orientation of the edge has a + orientation, otherwise, it has the opposite (–) orientation. Directed edge is used in **topology** to distinguish the right side (–) from the left side (+) of the same edge and the **start node** (–) and **end node** (+) of the same edge and in **computational topology** to represent these concepts.

4.28**directed face**

directed topological object that represents an association between a **face** and one of its orientations

NOTE The orientation of the **directed edges** that compose the exterior **boundary** of a directed face will appear positive from the direction of this vector; the orientation of a directed face that bounds a **topological solid** will point away from the **topological solid**. Adjacent solids would use different orientations for their shared boundary, consistent with the same sort of association between adjacent faces and their shared edges. Directed faces are used in the **coboundary** relation to maintain the spatial association between **face** and **edge**.

4.29**directed node**

directed topological object that represents an association between a **node** and one of its orientations

NOTE Directed nodes are used in the **coboundary** relation to maintain the spatial association between **edge** and **node**. The orientation of a node is with respect to an edge, “+” for end node, “–” for start node. This is consistent with the vector notion of “result = end - start”.

4.30**directed solid**

directed topological object that represents an association between a **topological solid** and one of its orientations

NOTE Directed solids are used in the **coboundary** relation to maintain the spatial association between **face** and **topological solid**. The orientation of a solid is with respect to a face, “+” if the upNormal is outward, “–” if inward. This is consistent with the concept of “up = outward” for a surface bounding a solid.

4.31**directed topological object**

topological object that represents a logical association between a **topological primitive** and one of its orientations

4.32**domain**

well-defined **set** [ISO/TS 19103]

NOTE Domains are used to define the domain and range of operators and **functions**.

4.33**edge**

1-dimensional **topological primitive**

NOTE The **geometric realization** of an edge is a **curve**. The **boundary** of an edge is the **set** of one or two **nodes** associated to the edge within a **topological complex**.

4.34**edge-node graph**

graph embedded within a **topological complex** composed of all of the **edges** and **connected nodes** within that **complex**

NOTE The **edge-node graph** is a subcomplex of the complex within which it is embedded.

ISO 19107:2003(E)

4.35

end node

node in the **boundary** of an **edge** that corresponds to the **end point** of that **edge** as a **curve** in any valid **geometric realization** of a **topological complex** in which the **edge** is used

4.36

end point

last point of a **curve**

4.37

exterior

difference between the universe and the **closure**

NOTE The concept of exterior is applicable to both **topological** and **geometric complexes**.

4.38

face

2-dimensional **topological primitive**

NOTE The **geometric realization** of a face is a **surface**. The **boundary** of a face is the **set of directed edges** within the same **topological complex** that are associated to the face via the boundary relations. These can be organized as **rings**.

4.39

feature

abstraction of real world phenomena [ISO 19101]

NOTE A feature may occur as a type or an **instance**. Feature type or feature instance should be used when only one is meant.

4.40

feature attribute

characteristic of a **feature** [ISO 19101]

NOTE A feature attribute has a name, a data type, and a value **domain** associated to it. A feature attribute for a feature **instance** also has an attribute value taken from the value domain.

4.41

function

rule that associates each element from a **domain** (source, or domain of the function) to a unique element in another domain (target, co-domain, or range)

4.42

geographic information

information concerning phenomena implicitly or explicitly associated with a location relative to the Earth [ISO 19101]

4.43

geometric aggregate

collection of **geometric objects** that has no internal structure

NOTE No assumptions about the spatial relationships between the elements can be made.

4.44

geometric boundary

boundary represented by a **set of geometric primitives** of smaller **geometric dimension** that limits the extent of a **geometric object**

4.45**geometric complex**

set of disjoint **geometric primitives** where the **boundary** of each **geometric primitive** can be represented as the union of other **geometric primitives** of smaller dimension within the same **set**

NOTE The **geometric primitives** in the set are disjoint in the sense that no **direct position** is **interior** to more than one **geometric primitive**. The set is closed under **boundary operations**, meaning that for each element in the **geometric complex**, there is a collection (also a **geometric complex**) of **geometric primitives** that represents the boundary of that element. Recall that the **boundary** of a point (the only 0D primitive object type in geometry) is empty. Thus, if the largest dimension geometric primitive is a solid (3D), the composition of the boundary operator in this definition terminates after at most three steps. It is also the case that the boundary of any object is a **cycle**.

4.46**geometric dimension**

largest number n such that each **direct position** in a **geometric set** can be associated with a subset that has the **direct position** in its **interior** and is similar (isomorphic) to R^n , Euclidean n -space

NOTE Curves, because they are continuous images of a portion of the real line, have geometric dimension 1. Surfaces cannot be mapped to R^2 in their entirety, but around each point position, a small neighbourhood can be found that resembles (under continuous functions) the interior of the unit circle in R^2 , and are therefore 2-dimensional. In this International Standard, most surface patches (instances of GM_SurfacePatch) are mapped to portions of R^2 by their defining interpolation mechanisms.

4.47**geometric object**

spatial object representing a **geometric set**

NOTE A **geometric object** consists of a **geometric primitive**, a collection of **geometric primitives**, or a geometric complex treated as a single entity. A geometric object may be the spatial representation of an **object** such as a **feature** or a significant part of a **feature**.

4.48**geometric primitive**

geometric object representing a single, **connected**, homogeneous element of space

NOTE Geometric primitives are non-decomposed **objects** that present information about geometric configuration. They include **points**, **curves**, **surfaces**, and **solids**.

4.49**geometric realization**

geometric complex whose **geometric primitives** are in a 1-to-1 correspondence to the **topological primitives** of a **topological complex**, such that the **boundary** relations in the two complexes agree

NOTE In such a realization the topological primitives are considered to represent the **interiors** of the corresponding geometric primitives. Composites are closed.

4.50**geometric set**

set of direct positions

NOTE This set in most cases is infinite.

4.51**graph**

set of nodes, some of which are joined by **edges**

NOTE In geographic information systems, a graph can have more than one **edge** joining two **nodes**, and can have an **edge** that has the same **node** at both ends.