

**DIN EN ISO 19107**

ICS 35.240.70

**Entwurf**

Einsprüche bis 2018-01-24  
Vorgesehen als Ersatz für  
DIN EN ISO 19107:2005-05

**Geoinformation –  
Raumbezugsschema (ISO/DIS 19107:2017);  
Englische Fassung prEN ISO 19107:2017**

Geographic information –  
Spatial Schema (ISO/DIS 19107:2017);  
English version prEN ISO 19107:2017

Information géographique –  
Schéma spatial (ISO/DIS 19107:2017);  
Version anglaise prEN ISO 19107:2017

**Anwendungswarnvermerk**

Dieser Norm-Entwurf mit Erscheinungsdatum 2017-11-24 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfs besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal von DIN unter [www.din.de/go/entwuerfe](http://www.din.de/go/entwuerfe) bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter [www.entwuerfe.normenbibliothek.de](http://www.entwuerfe.normenbibliothek.de), sofern dort wiedergegeben;
- oder als Datei per E-Mail an [nabau@din.de](mailto:nabau@din.de) möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter [www.din.de/go/stellungnahmen-norm-entwuerfe](http://www.din.de/go/stellungnahmen-norm-entwuerfe) oder für Stellungnahmen zu Norm-Entwürfen der DKE unter [www.dke.de/stellungnahme](http://www.dke.de/stellungnahme) abgerufen werden;
- oder in Papierform an den DIN-Normenausschuss Bauwesen (NABau), 10772 Berlin, Burggrafenstr. 6, 10787 Berlin.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 258 Seiten

DIN-Normenausschuss Bauwesen (NABau)

## Nationales Vorwort

Dieses Dokument (prEN ISO 19107:2017) wurde vom Technischen Komitee ISO/TC 211 „Geographic information/Geomatics“ in Zusammenarbeit mit dem Technischen Komitee CEN/TC 287 „Geoinformation“, dessen Sekretariat von BSI (Vereinigtes Königreich) gehalten wird, erarbeitet.

Das zuständige nationale Gremium ist der Arbeitsausschuss NA 005-03-03 AA „Geoinformation (SpA zu CEN/TC 287 und ISO/TC 211)“ im DIN-Normenausschusses Bauwesen (NABau).

Dieser europäische Norm-Entwurf enthält unter Berücksichtigung des DIN-Präsidialbeschlusses 1/2004 nur die englische Originalfassung von EN ISO 19107:2017 und ISO 19107:2017.

Für die in diesem Dokument zitierten Internationalen Normen wird im Folgenden auf die entsprechenden Deutschen Normen hingewiesen:

|             |                          |
|-------------|--------------------------|
| ISO 6709    | siehe DIN EN ISO 6709    |
| ISO 19101-1 | siehe DIN EN ISO 19101-1 |
| ISO 19103   | siehe DIN EN ISO 19103   |
| ISO 19107   | siehe DIN EN ISO 19107   |
| ISO 19108   | siehe DIN EN ISO 19108   |
| ISO 19109   | siehe DIN EN ISO 19109   |
| ISO 19111   | siehe DIN EN ISO 19111   |
| ISO 19111-2 | siehe DIN EN ISO 19111-2 |
| ISO 19118   | siehe DIN EN ISO 19118   |
| ISO 19125-1 | siehe DIN EN ISO 19125-1 |
| ISO 19125-2 | siehe DIN EN ISO 19125-2 |
| ISO 19136   | siehe DIN EN ISO 19136   |
| ISO 19141   | siehe DIN EN ISO 19141   |
| ISO 19148   | siehe DIN EN ISO 19148   |
| ISO 80000-2 | siehe DIN EN ISO 80000-2 |

## Änderungen

Gegenüber DIN EN ISO 19107:2005-05 wurden folgende Änderungen vorgenommen:

- a) Revision/Korrektur aus technischer Sicht, die Ergänzung notwendiger Erweiterungen, die sich aus der praktischen Anwendung ergeben haben und eine Behebung von Fehlern;
- b) Entfernung überflüssiger Klassen und Konstrukte Integration in andere, so sind z. B. alle Aggregations-typen nun in der Klasse „Collection“ zusammengefasst;
- c) diese Fassung ist rückwärtskompatibel und Anwendungsschemata auf der Basis der alten DIN EN ISO 19107 sind nach wie vor gültig;
- d) Anhang D des neuen Standards enthält eine Gegenüberstellung der Konstrukte alter und neuer Versionen;
- e) die Conformance-Klassen wurden mit dem Ziel erweitert, Implementierungen mehr Möglichkeiten zu geben.

## Nationaler Anhang NA (informativ)

### Literaturhinweise

DIN EN ISO 6709, *Standarddarstellung für geographische Punkte durch Koordinaten*

DIN EN ISO 19101-1, *Geoinformation — Referenzmodell — Teil 1: Grundsätze*

DIN EN ISO 19103, *Geoinformation — Konzeptuelle Modellierungssprache*

DIN EN ISO 19107, *Geoinformation — Raumbezugsschema*

DIN EN ISO 19108, *Geoinformation — Zeitliches Schema*

DIN EN ISO 19109, *Geoinformation — Regeln zur Erstellung von Anwendungsschemata*

DIN EN ISO 19111, *Geoinformation — Koordinatenreferenzsysteme*

DIN EN ISO 19111-2, *Geoinformation — Koordinatenreferenzsysteme — Teil 2: Erweiterung auf parametrisierte Werte*

DIN EN ISO 19118, *Geoinformation — Kodierung*

DIN EN ISO 19125-1, *Geoinformation — Simple feature access — Teil 1: Gemeinsame Architektur*

DIN EN ISO 19125-2, *Geoinformation — Simple feature access — Teil 2: Structured Query Language (SQL)*

DIN EN ISO 19136, *Geoinformation — Geography Markup Language (GML)*

DIN EN ISO 19141, *Geoinformation — Schema für sich bewegende Objekte*

DIN EN ISO 19148, *Geoinformation — Lineares Bezugssystem*

DIN EN ISO 80000-2, *Größen und Einheiten — Teil 2: Mathematik*

## Nationaler Anhang NB (informativ)

### Begriffe

Die Benummerung der folgenden Begriffe und Abkürzungen ist identisch mit der Benummerung in der englischen Fassung.

#### 4 Begriffe

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe. Alle Begriffe in einer normativen Verweisung gelten auch, wenn sie im Kontext dieser Verweisung verwendet werden. Die Definitionen gängiger Datentypen wurden ISO 19103, ISO/IEC 11404 oder OMG-UML entnommen. Jeder Begriff in den normativen Verweisungen in Abschnitt 3 darf verwendet werden.

Gängige Wörter aus der Geometrie, wie Punkt, Kurve Linie, Flächenkörper usw. tragen die gängigen Bedeutungen, es sei denn, sie werden als Klassifikator-Bezeichnungen (üblicherweise Schnittstellen) verwendet; in diesem Fall sind sie eine digitale Darstellung eines geometrischen Konzepts. Gängige mathematische Begriffe, die hier nicht definiert sind, tragen ihre gängigen mathematischen Bedeutungen (siehe [3], ISO 80000-2, ISO/IEC 11404 oder ein Standardwerk zum Thema, wie die „N. Bourbaki“-Reihe<sup>2)</sup>, gegenwärtig im Springer-Verlag in Französisch, Englisch und Deutsch veröffentlicht). Es sollte beachtet werden, dass mathematische Begriffe kontextsensitiv sein können und leicht mit allgemeinsprachlichen Begriffen verwechselt werden können. Beispielsweise sind Begriffe wie „offene“ Menge, „abgeschlossene“ Kurve, „abgeschlossene“ Menge, „rationale“ Funktion, „Begrenzung“, „Inneres“, „Abgeschlossenheit“, „Äußeres“, „Funktion“ und andere allgemeinsprachlich, besitzen jedoch in der Mathematik und in dieser Internationalen Norm sehr spezifische Bedeutungen. Wo es zur Verhinderung von Verwechslungen erforderlich war, wurden bestehende Definitionen überarbeitet, um ihre Bedeutung in dieser Internationalen Norm zu verdeutlichen. Mathematische Begriffe umfassen gängiges Vokabular aus Geometrie, Topologie, Infinitesimalrechnung, Geodäsie und Differentialgeometrie. Viele dieser Begriffe können ausreichend geläufig sein, so dass ihre Aufnahme nicht notwendig ist. Sie sind hier enthalten, um Verwechslungen insbesondere mit Begriffen wie den vorstehend genannten zu verhindern, die in anderem Zusammenhang andere Bedeutungen haben.

**4.1**  
**abstrakte Wurzel**  
(en: abstract root)  
<Programmierung> üblicher Wurzel-Klassifikator einer Kategorie, der allen anderen Klassifikatoren in der Kategorie übergeordnet ist

Anmerkung 1 zum Begriff: Die Klasse „Any“ ist in einigen Programmiersprachen die abstrakte Wurzel aller Klassen. Daher ist sie *de facto* die Vereinigung aller Klassen. In dieser Internationalen Norm ist Geometrie die (benannte und ausdrückliche) abstrakte Wurzel für alle Geometrieobjekte. In dem Modul „Geometrie“ und jedem seiner Teilmodule (einschließlich der in ihren Anforderungsklassen) wird jede Schnittstelle ein Untertyp von Geometrie sein, entweder direkt oder transitiv.

---

<sup>2)</sup> N. Bourbaki ist das Pseudonym für die „Association des collaborateurs de Nicolas Bourbaki“ (de: Gesellschaft der Mitarbeiter von Nicolas Bourbaki) von Mathematikern, die erstmals 1935 veröffentlichte und sich die „Formalisierung“ der Mathematik zum Ziel gesetzt hatte. Die Gruppe hält ein Büro an der „École Normale Supérieure“ in Paris. Siehe [https://en.wikipedia.org/wiki/Nicolas\\_Bourbaki](https://en.wikipedia.org/wiki/Nicolas_Bourbaki). Ihre Veröffentlichungen genießen in der mathematischen Fachwelt hohes Ansehen.

## **4.2**

### **Bogen**

(en: arc)

<Geometrie> Abschnitt einer Kurve

## **4.3**

### **baryzentrische Koordinaten**

(en: barycentric coordinates)

<analytische Geometrie> Punkt in einem n-dimensionalen Koordinatensystem mit  $n+1$  Zahlen,  $[u_0, u_1, u_2, u_3, \dots, u_n]$ ,  $0 \leq u_i \leq 1$ ,  $\sum u_i = 1$ , in dem die Lage eines Punkts eines n-Simplex (von beliebiger Dimension) durch einen gewichteten Massenmittelpunkt zweier Massen festgelegt wird, die auf ihren Eckpunkten liegen, unter Anwendung von Vektoralgebra von  $\mathbb{R}^n$  verwendet in dem Koordinatenreferenzsystem

Anmerkung 1 zum Begriff: Obwohl es  $n+1$  Koordinaten in einem baryzentrischen Koordinatensystem gibt, beträgt die topologische Dimension  $n$ , weil die Nebenbedingung (summiert sich auf 1,0) einen Freiheitsgrad verliert (sobald es  $n$  Ordinaten gibt, wird die verbleibende bestimmt, wie  $u_n = 1,0 - \sum_{i=0}^{n-1} u_i$ ). Die Koordinaten für den Simplex sind sämtlich nicht-negativ, aber das System kann durch Verwendung negativer Zahlen über die Simplexe erweitert werden. Sind alle Ordinaten positiv, dann liegt der Punkt im (im Inneren des) n-Simplex. Beträgt eine von ihnen 1,0 und die andere 0, ist dies der Eckpunkt eines Simplex. Beträgt eine von ihnen null und die anderen sind noch jeweils größer gleich null, dann liegt der Punkt auf dem  $n-1$ -Simplex, gegenüber dem auf null gesetzten Scheitelpunkts. Sind einige negativ, liegt der Punkt außerhalb des Simplex. Die Koordinaten sind abhängig von dem zugrundeliegenden Koordinatenreferenzsystem der Quelldaten.

## **4.4**

### **Peilung**

(en: bearing)

in der Horizontalebene gemessene(r) Winkel, Tangente oder Richtung in einem Punkt

Anmerkung 1 zum Begriff: Diese Definition (im Gegensatz zu der nach ISO 19162:2015) ist für diese Internationale Norm erforderlich, weil der Begriff in anderen Definition verwendet wird, z. B. erstes geodätisches Problem und zweites geodätisches Problem. Die beiden Definition sind nahezu äquivalent, weil die Tangente einer Kurve auf einer Fläche eine Tangente zur Fläche ist und eine Richtung festlegt. Ein übliches 2-D-Maß der Peilung kann ein Winkel sein, der in äquivalenter Form von Norden aus im Uhrzeigersinn gemessen wird oder ein Einheitstangentialvektor. Wenn das Koordinatensystem räumlich dreidimensional ist, kann der horizontale Peilungswinkel zu seiner Vollständigkeit auch einen vertikalen Höhenwinkel erfordern. Wenn eine Referenzkurve (wie in ISO 19162 verwendet) durch die Bogenlänge parametrisiert wird, dann ist ihre „Ableitung“ ein Einheitsvektor. Wenn eine andere Parametrisierung „ $t$ “ angewendet wird, dann sollte die Ableitung normiert sein ( $\vec{\tau}/\|\vec{\tau}\|$ ;  $\dot{c}(t) = \vec{\tau}$ ; ). Das ist nützlich, da das Parametrisieren der Bogenlänge rechentechnisch schwierig sein kann. Die numerische Darstellung eines Vektors ist abhängig vom Koordinatensystem. Die Peilung ist nicht abhängig vom Koordinatensystem, sondern kann in jedem plausiblen System dargestellt werden. Die Peilung ist nicht abhängig von ihren verschiedenen Darstellungen.

## **4.5**

### **bikontinuierlich**

(en: bicontinuous)

<Mathematik> umkehrbar, stetig und mit einer stetigen Inverse

## **4.6**

### **Begrenzung**

(en: boundary)

**Menge**, die die Grenzen einer Entität darstellt

Anmerkung 1 zum Begriff: Der Begriff „Begrenzung“ wird am häufigsten im Kontext der Geometrie benutzt, bei der die Menge aus einer Sammlung von Punkten oder einer Sammlung von Objekten, die von diesen Punkten dargestellt wird, besteht. In anderen Bereichen wird der Begriff im übertragenen Sinne gebraucht, um den Übergang zwischen einer Entität und dem übrigen betroffenen Domain zu bezeichnen.

#### 4.7

##### **Puffer**

(en: buffer)

Geometrieobjekt, das alle Punkte enthält und nur diejenigen Punkte, deren Abstand von einem festgelegten Geometrieobjekt kleiner gleich einem, bei seiner Konstruktion verwendeten, angegebenen Abstand ist

#### 4.8

##### **abgeschlossene Hülle**

(en: closure)

Vereinigung des Inneren und der Begrenzung eines Topologieobjekts oder Geometrieobjekts

#### 4.9

##### **Korand**

(en: coboundary)

Menge von topologischen Primitiven von höherer topologischer Dimension, die mit einem bestimmten Topologieobjekt assoziiert ist, sodass dieses Topologieobjekt in jeder ihrer Begrenzungen liegt

Anmerkung 1 zum Begriff: Falls ein Knoten auf der Begrenzung einer Kante liegt, ist diese Kante der Korand dieses Knotens. Jeder Orientierungsparameter, der mit einer dieser Relationen assoziiert ist, würde auch mit der anderen assoziiert sein. Der Korand eines Knotens kann als „Stern“ bezeichnet werden.

#### 4.10

##### **konform**

(en: conformal)

winkeltreu

Anmerkung 1 zum Begriff: Einige Projektionen sind konform. Beispielsweise stellt eine Mercatorprojektion der Winkel zwischen Kurven winkeltreu dar, so dass wenn zwei Kurven in einer Mercatorprojektionsebene einander im Winkel von 90° schneiden, dann schneiden auch die Urbild-Kurven auf dem Ellipsoid einander im Winkel von 90°, wie Linien stetiger geographischer Breite und Linien stetiger geographischer Länge.

#### 4.11

##### **zusammenhängend**

(en: connected)

Eigenschaft eines topologischen Raums, die besagt, dass nur der gesamte Raum oder die leere Menge die einzigen Teilmengen sind, die sowohl offen als auch abgeschlossen sind

Anmerkung 1 zum Begriff: Die formale Definition von zusammenhängend ist, dass ein beliebiges Paar von lokal offenen Mengen, deren Vereinigung der gesamte Raum ist, eine nicht-leere Schnittmenge haben müssen.

ein topologischer Raum  $T$  ist zusammenhängend, wenn und nur wenn  
 $[X, Y \text{ sind offen}] \Rightarrow [X, Y \subset T, X \cup Y = T \Rightarrow X \cap Y \neq \emptyset]$  (2)

Diese formale Definition ist schwierig anzuwenden. Der nachstehend definierte Begriff „wegweise zusammenhängend“ ist äquivalent zu dem in dieser Internationalen Norm beabsichtigten. Die Verwendung oder „endlich genaue“ Koordinaten macht die Darstellung von Mengen, die zusammenhängend, aber nicht wegweise zusammenhängend sind, unmöglich. In allen Fällen wird „zusammenhängend“ verwendet, aber „wegweise zusammenhängend“ ist einfacher zu prüfen und zu visualisieren.

#### 4.12

##### **zusammenhängender Knoten**

(en: connected node)

<Topologie> Knoten, der Anfang oder Ende einer oder mehrerer Kanten ist

#### 4.13

##### Kontrollpunkt

(en: control point)

<analytische Geometrie> Punkt, der in der Konstruktion einer Geometrie verwendet wird, der ihre Form teilweise kontrolliert aber nicht notwendigerweise auf dieser Geometrie liegt

Anmerkung 1 zum Begriff: Ein Mittelpunkt eines Bogens ist ein Kontrollpunkt; Pole in B-Spline-Kurven sind Kontrollpunkte.

#### 4.14

##### konvex

(en: convex)

<Geometrie> enthält alle Punkte auf einer „Linie“, die zwei innenliegende Punkte verbindet

Anmerkung 1 zum Begriff: Die Definition von „konvex“ erfordert eine Definition von „Linie“. Bei Koordinatensystemen ist das der übliche linear interpolierte Bogen, aber in diesem Zusammenhang ist die „Linie“ auf einer geometrischen Bezugsoberfläche ein „geodätischer Bogen“. Das Standardverfahren in diesem Dokument ist die lineare Interpolation.

#### 4.15

##### konvexe Hülle

(en: convex hull)

<Geometrie> kleinste konvexe Menge, die ein gegebenes Geometrieobjekt enthält

Anmerkung 1 zum Begriff: „Kleinste“ ist die theoretisch kleinste Menge, nicht ein Ergebnis einer Messung. Die Definition kann umformuliert werden in: „die Schnittmenge aller konvexen Mengen, die das Geometrieobjekt enthalten“. Eine weitere Definition in einem euklidischen Raum  $\mathbb{E}^n$  ist die Vereinigung aller Linien mit beiden Endpunkten im gegebenen Geometrieobjekt.

$$\begin{aligned} C = A.\text{convexHull} &\Leftrightarrow \\ [C.\text{convex} = \text{TRUE}] \wedge [A \subset C] \wedge \\ [[B.\text{convex} = \text{true}, A \subset B] \Rightarrow [A \subseteq C \subseteq B]] \end{aligned} \quad (3)$$

#### 4.16

##### Koordinate

(en: coordinate)

Zahl in einer Folge von Zahlen zum Bezeichnen der Lage eines Punktes

Anmerkung 1 zum Begriff: In einem Koordinatenreferenzsystem werden die Zahlen durch Einheiten qualifiziert. Die Anzahl der Verschiebungen (im Allgemeinen als „Ordinaten“ bezeichnet) in einer Koordinate gibt nicht die Dimension an. Wenn es eine Nebenbedingung gibt, kann die Dimension kleiner sein. Siehe „Koordinatendimension“.

#### 4.17

##### Koordinatendimension

(en: coordinate dimension)

<analytische Geometrie> Anzahl der Einzelentscheidungen, die zur Beschreibung einer Position in einem Koordinatensystem erforderlich sind

Anmerkung 1 zum Begriff: Die Koordinatendimension repräsentiert die Anzahl der getroffenen Entscheidungen, und Nebenbedingungen können diese Auswahl beschränken. Eine baryzentrische Koordinate, die (n+1)-Versätze aufweist, der zugrundeliegende Raum ist jedoch n-dimensional. Homogene Koordinaten  $(wx, wy, wz, w)$  sind genau genommen dreidimensional, weil die Wahl von „w“ nicht die Position beeinflusst, d. h.  $(wx, wy, wz, w) = (x, y, z, 1) \rightarrow (x, y, z)$ , was nicht von w beeinflusst ist. Die Dimension wird höchstens die Anzahl der Zahlen in der Koordinate sein, sie kann jedoch niedriger sein, wenn die Koordinaten in irgend einer Weise Nebenbedingungen unterliegen.

#### 4.18

##### Koordinatenreferenzsystem

(en: coordinate reference system)

<Differentialgeometrie, Geodäsie> Koordinatensystem, das durch ein Datum auf ein Objekt bezogen ist

[ISO 19111:2007]

Anmerkung 1 zum Begriff: Die ursprüngliche Definition von Koordinatenreferenzsystem (CRS) verwendet ein Geometrieobjekt (ein geodätisches Datum), das sich durch ein Datum auf die reale Welt bezieht. ISO 19111-2 erweitert dies auf jeden „Parameter“, der im Wesentlichen als Graph der Relation des Parameters dargestellt werden kann. Dieser Graph liegt im Vektorprodukt des Datums, des Bereichsraums der Parameterfunktion und des Parameterraums (der hingegen mehrdimensional sein darf). Deshalb ist jedes in diesen Internationalen Normen verwendete Koordinatensystem logischerweise ein CRS. Da dies nicht die ursprüngliche Absicht von ISO 19111-2 war, wird diese Internationale Norm CRS für Koordinatensysteme verwenden, die mit einem geodätischen Datum assoziiert sind und verwendet den allgemeineren Begriff „Koordinatensystem“ für alle weiteren, die entweder größer oder kleiner sind. Hier bedeutet CRS Koordinaten mit Raumbezug. In jedem Fall, bei dem das zugrundeliegende Datum (Fläche) nicht eben ist, ist das Koordinatensystem nichteuklidisch und die Metriken sind nichtpythagoreisch.

Anmerkung 2 zum Begriff: Falls in späteren Versionen von ISO 19111 die vorstehend zitierte Definition geändert wird, sollte das keinen Einfluss auf diese Norm haben. Im Fall, dass dynamische Daten verwendet werden, würden Messungen, die durch geometrische Operationen von Objekten in dieser Norm durchgeführt wurden, nur für die für das Datum und die zugehörigen Koordinaten verwendete Zeit gelten. Variiert das Datum, können die Messungen ebenfalls variieren, aber die Definitionen der Maße bleiben die gleichen, die tatsächlichen Werte können jedoch eine zeitabhängige Funktion sein.

#### 4.19

##### Krümmungsvektor

(en: curvature vector)

<Differentialgeometrie> zweite Ableitung einer durch die Bogenlänge parametrisierten Kurve in einem Punkt

Notation: Wenn  $c(s) = (x(s), y(s), z(s))$  eine Kurve im dreidimensionalen kartesischen Raum ist ( $\mathbb{E}^3$ ) und  $s$  die Bogenlänge entlang  $c(s)$ , dann ist der Einheitstangentenvektor  $\dot{c}(s) = (\dot{x}(s), \dot{y}(s), \dot{z}(s))$ , d. h. die Ableitung der Koordinatenwerte von „ $c$ “ in Bezug auf „ $s$ “. Der Krümmungsvektor beträgt  $\ddot{c}(s) = (\ddot{x}(s), \ddot{y}(s), \ddot{z}(s))$ . Der Krümmungsvektor kann durch die Inverse des Radius eines Kreises durch beliebige drei nahegelegene Punkte auf der Kurve angenähert werden (von der Kurve zum Mittelpunkt des Kreises weisend).

#### 4.20

##### Zyklus

(en: cycle)

<Geometrie, Topologie> beschränktes räumliches Objekt mit einer leeren Begrenzung

Anmerkung 1 zum Begriff: Zyklen werden zur Beschreibung von Begrenzungskomponenten verwendet. Ein Zyklus hat üblicherweise keine Begrenzung, da er selbstabschließend ist, aber er ist beschränkt (d. h., er hat keine unendliche Ausdehnung). Beispielsweise besitzen ein Kreis oder eine Kugel keine Begrenzung (d. h. die Begrenzung ist leer), aber sie sind beschränkt.

#### 4.21

##### Datenpunkt

(en: data point)

<analytische Geometrie> Punkt, der auf der Geometrie liegt

Anmerkung 1 zum Begriff: Die Eckpunkte in einem Polygonzug sind Datenpunkte; die Punkte, die zur Konstruktion eines polynomen Splines verwendet werden, sind Datenpunkte. Datenpunkte können als Kontrollpunkte verwendet werden, sie werden jedoch häufig nach Konstruktion der Geometrie abgeleitet.

#### 4.22

##### **Durchmesser**

(en: diameter)

<Metrik> größter Abstand zwischen zwei Punkten in einer Menge von Punkten

#### 4.23

##### **gerichtete Kante**

(en: directed edge)

<Topologie> gerichtetes Topologieobjekt, dass eine Zuordnung zwischen einer Kante und einer ihrer Orientierungen darstellt

Anmerkung 1 zum Begriff: Eine gerichtete Kante, die mit der Orientierung der Kante übereinstimmt, hat eine + Orientierung, anderenfalls hat sie die gegensätzliche (–) Orientierung. Gerichtete Kanten werden in der Topologie zur Unterscheidung der rechten Seite (–) von der linken Seite (+) derselben Kante sowie des Startknotens (–) und Endknotens (+) derselben Kante verwendet, und in der rechnerischen Topologie zur Darstellung dieser Konzepte.

#### 4.24

##### **gerichtete Masche**

(en: directed face)

<Topologie> gerichtetes Topologieobjekt, dass eine Zuordnung zwischen einer Fläche und einer ihrer Orientierungen darstellt

Anmerkung 1 zum Begriff: Die Orientierung der gerichteten Kanten, die die äußere Begrenzung einer gerichteten Masche bilden, wird durch die Richtung dieses Vektors positiv erscheinen; die Orientierung einer gerichteten Masche, die einen topologischen Körper beschränkt, wird vom topologischen Körper fort weisen. Benachbarte Körper würden verschiedene Orientierungen für ihre gemeinsame Begrenzung verwenden, übereinstimmend mit derselben Art von Assoziation zwischen benachbarten Flächen und ihren gemeinsamen Kanten. Gerichtete Maschen werden in der Korandrelation verwendet, um die räumliche Assoziation zwischen Masche und Kante zu bewahren.

#### 4.25

##### **gerichteter Knoten**

(en: directed node)

<Topologie> gerichtetes Topologieobjekt, dass eine Zuordnung zwischen einem Knoten und einer seiner Orientierungen darstellt.

Anmerkung 1 zum Begriff: Gerichtete Knoten werden in der Korand-Relation verwendet, um die räumliche Assoziation zwischen Kante und Knoten zu bewahren. Die Orientierung eines Knotens in Bezug auf eine Kante ist „+“ bei einem Endknoten und „–“ bei einem Startknoten. Das stimmt mit den Vektorbegriff von „Ergebnis = Ende – Start“ überein.

#### 4.26

##### **gerichteter Körper**

(en: directed solid)

<Topologie> gerichtetes Topologieobjekt, dass eine Zuordnung zwischen einem topologischen Körper und einer seiner Orientierungen darstellt

Anmerkung 1 zum Begriff: Gerichtete Körper werden in Korand-Relation verwendet, um die räumliche Zuordnung zwischen Masche und topologischem Körper zu erhalten. Die Orientierung eines Körpers ist hinsichtlich einer Fläche „+“, wenn der aufsteigende Normalenvektor auswärts gerichtet ist und „–“, wenn er einwärts gerichtet ist. Das entspricht dem Konzept von „auf = auswärts“ bei einer Fläche, die einen Körper beschränkt.

#### 4.27

##### Abstand

(en: distance)

<Geometrie, metrische Räume> Mindestlänge einer Kurve, die die beiden Punkte oder Geometrien verbindet

Anmerkung 1 zum Begriff: Die übliche Metrik für zwei Punkte in einem Koordinatenraum setzt eine zugrundeliegende Ebene voraus und ist ein euklidischer Abstand. Wenn eine zugrundeliegende Bezugsfläche keine Ebene ist, dann wird der Abstand durch diese Mindestlänge aller Kurven zwischen den beiden Punkten definiert. Diese Flächen sind Hauptbeispiele der nichteuklidischen Geometrie, bei der das Parallelenaxiom in Euklids Elementen nicht gilt. In mathematischem Sinne ist Abstand die „größte untere Schranke“ der Kurvenlänge. Das Wort Minimum wird auch gebraucht, aber es sollte nicht erwartet werden, dass ein tatsächliches Minimum eintritt, sondern nur, dass eine beliebige größere Zahl eine Länge in dieser Menge aufweisen wird, die kleiner ist.

#### 4.28

##### Bereich

(en: domain)

<Mathematik> wohldefinierter Bereich

Anmerkung 1 zum Begriff: Bereiche werden zum Definieren des Bereichs und Wertebereichs von Operatoren und Funktionen verwendet. Wenn eine Funktion „f“ Werte aus dem Quellbereich „X“ zu Werten im Zielbereich „Y“ abbildet, dann ist die Notation „f:X→Y“. Das Verhältnis zwischen der Funktion und ihrem Quellbereich wird als „Bereich“ bezeichnet. Das Verhältnis zwischen der Funktion und ihrem Zielbereich wird als „Bereich“ bezeichnet.

#### 4.29

##### Kante

(en: edge)

<Topologie> eindimensionales topologisches Primitiv

Anmerkung 1 zum Begriff: Die geometrische Realisierung einer Kante ist eine Kurve. Die Begrenzung einer Kante ist die Menge von einem oder zwei Knoten, die mit der Kante in einem topologischen Komplex assoziiert sind.

#### 4.30

##### Kante-Knoten-Graph

(en: edge-node graph)

<Topologie> Graph, der in einen topologischen Komplex eingebettet ist, der sich aus allen Kanten und zusammenhängenden Knoten in diesem Komplex zusammensetzt

Anmerkung 1 zum Begriff: Der Kante-Knoten-Graph ist ein Teilkomplex des Komplexes, in dem er eingebettet ist.

#### 4.31

##### Ellipsoid

(en: ellipsoid)

<Geodäsie> geometrische Referenzfläche, die in einen dreidimensionalen Euklidischen Raum eingebettet ist, der durch ein Rotationsellipsoid repräsentiert wird

Anmerkung 1 zum Begriff: Im Fall der Erde erfolgt die Rotation um die Polachse. Das führt zu einem abgeplatteten Ellipsoid dessen Brennpunkt im nominellen Erdmittelpunkt liegt.

Anmerkung 2 zum Begriff: Die beiden gängigen Algorithmen zur Berechnung der geographischen Breite auf einem Ellipsoid und auf einer Kugel (wie bei Kugelkoordinaten verwendet) sind nur dann äquivalent, wenn das Ellipsoid eine Kugel ist, deren Radien in alle Richtungen gleich lang sind. Das Problem ist, dass eine vom Mittelpunkt eines allgemeinen Ellipsoids ausgehende Radiallinie, die Oberfläche des Ellipsoids nicht immer orthogonal schneidet. Im Allgemeinen schneiden Schnittebenen durch den Mittelpunkt die Oberfläche nicht orthogonal und deshalb sind die Kurven, die den Großkreisen einer Kugel entsprechen, keine Geodäten auf dem Ellipsoid.

Anmerkung 2 zum Begriff: Die Topologie eines Ellipsoids ist vom  $\mathbb{E}^3$ -Raum übernommen, in dem es eingebettet ist. Der Unterschied besteht darin, dass Metriken, wie Abstand und Richtung auf dem Ellipsoid, auf Kurven beschränkt sind, die vollständig auf der Oberfläche verlaufen und auf Vektoren, die tangential zur Oberfläche sind.

#### **4.32**

##### **leere Menge**

$\emptyset$

(en: empty set)

<Mathematik> Menge, die keine Elemente enthält

Anmerkung 1 zum Begriff: Mengen sind gleich, wenn sie exakt dieselben Elemente enthalten. Da zwei beliebige leere Mengen exakt dieselben Elemente enthalten würden (nach Definition keine), sind sie, nach Definition, gleich. Die leere Menge ( $\emptyset$ ) kann als eine geometrische Entität angesehen werden, weil sämtliche in ihr enthaltenen Elemente Punkte sind. Das ist eine gegenstandslose Aussage, weil die Menge  $\emptyset$  keine Elemente enthält und daher die Aussage „Für alle...“ nicht prüfbar und daher für jeden ihrer nicht existierenden Fälle wahr ist. Es gibt viele wahre, aber gegenstandslose Aussagen in Nachweisen von  $\emptyset$ . Das kann für manche Programmierer verwirrend sein, weil viele Systeme typischere Mengen verwenden, in denen die Klasse der Entitäten eine Klasse für die Container-Menge bestimmt. In der Mathematik wird „Klasse“ nicht berücksichtigt und es werden ausschließlich Mengen betrachtet, so dass eine leere Menge von Erdferkeln und eine leere Menge von Zebras in der Mathematik dieselbe Menge sind (ist?). Eine weitere Verwechslungsmöglichkeit besteht darin, dass  $\emptyset$  nicht die Datenbank-Null ist, die von Codd eingeführt wurde und die in relationalen und anderen Abfragesprachen in 3-wertiger Logik verwendet wird. „Null“ bedeutet „unbekannt“ und viele Aussagen, die Null enthalten sind unentscheidbar (weder nachweisbar wahr noch nachweisbar falsch). Die leere Menge ist kein „Mangel an Wissen“ sondern die Gewissheit der Nichtexistenz der Elemente in der Menge. Die meisten Aussagen, die mit „für alle Elemente in  $\emptyset$ “ beginnen, wahr sind, aber gegenstandslos. Die meisten Aussagen, die mit „es existiert ein Element in  $\emptyset$ “ beginnen, sind stets kategorisch falsch. Es ist beinahe unmöglich eine unentscheidbare Aussage über  $\emptyset$  zu konstruieren. Null und  $\emptyset$  stehen nicht in Zusammenhang. „Leer“ kann „ungültig“ oder „vollständig leer“ bedeuten.

#### **4.33**

##### **Endknoten**

(en: end node)

<Topologie> Knoten in der Begrenzung einer Kante, die dem Endpunkt dieser Kante entspricht

#### **4.34**

##### **Endpunkt**

(en: end point)

<Geometrie, Topologie> letzter Punkt einer Kurve

#### **4.35**

##### **Fehlerbudget**

(en: error budget)

<Metrik> Aussage oder Verfahren zum Beschreiben von Art und Ausmaß der Fehler, die die Berechnungsergebnisse beeinträchtigen

Anmerkung 1 zum Begriff: In den üblichsten Fällen beschreiben in dieser Internationalen Norm Fehlerbudgets metrische Berechnungen unter Verwendung darstellender Geometrieobjekte, um Echtwelt-Metriken zu schätzen, wie Abstand und Fläche.

#### **4.36**

##### **Exponentialabbildung**

(en: exponential map)

Funktion, die Tangentialvektoren in einem Punkt zum Endpunkt einer Geodäte abbildet, beginnend an diesem Punkt mit einer Austrittsrichtung gleich der des Vektors und einer Länge gleich der des Vektors

Anmerkung 1 zum Begriff: Siehe erstes geodätisches Problem für eine Erklärung des Prozesses zur Berechnung dieser Abbildung.

#### 4.37

##### **Äußeres**

(en: exterior)

<Geometrie, Topologie> Unterschied zwischen der Grundmenge und der abgeschlossenen Hülle

Anmerkung 1 zum Begriff: Der Begriff des Äußeren ist sowohl auf das Topologieobjekt als auch auf das Geometrieobjekt anwendbar.

#### 4.38

##### **Masche**

(en: face)

<Topologie> zweidimensionales topologisches Primitiv

Anmerkung 1 zum Begriff: Die geometrische Realisierung einer Masche ist eine Fläche. Die Begrenzung einer Masche besteht aus einer Menge gerichteter Kanten innerhalb desselben topologischen Komplexes, die über die Grenzverhältnisse mit der Masche assoziiert sind. Diese können in Form von Ringen organisiert sein.

#### 4.39

##### **erstes geodätisches Problem**

##### **direktes geodätisches Problem**

(en: first geodetic problem, direct geodetic problem)

<Differentialgeometrie, Geodäsie> wenn ein bestimmter Punkt in einer Fläche und die Richtung und der Abstand von diesem Punkt zu einem zweiten Punkt entlang einer Geodäte gegeben sind, das Problem, diesen zweiten Punkt zu bestimmen

Anmerkung 1 zum Begriff: Dieses „Problem“ definiert eine Abbildung aus dem Vektorraum an einem Punkt (jeder durch eine Richtung und eine Länge vorgegebene Vektor) zu Punkten der Erdfigur, die das Problem für diese Richtung und diesen Abstand erfüllt. Wenn z. B. der Abstand „r“ festgelegt wird und alle Richtungen genommen werden, ist die sich daraus ergebende Geometrie der Kreis, dessen Mittelpunkt im Ursprung des Radius „r“ liegt. Diese Internationale Norm wird massiven Gebrauch von dieser Abbildung machen, siehe Exponentialabbildung und das zweite geodätische Problem.

#### 4.40

##### **Grundfläche**

(en: footprint)

zweidimensionale Ausdehnung oder Projektion eines 3-D-Objekts auf einer horizontalen Oberfläche

#### 4.41

##### **Funktion**

(en: function)

<Mathematik, Programmierung> Regel, die jedes Element eines Bereichs („Quellbereich“ oder „Bereich“ der Funktion) mit einem einzelnen Element in einem anderen Bereich („Zielbereich“, „Wertebereich“ oder „Wertemenge“ der Funktion) verbindet

Anmerkung 1 zum Begriff: Bereiche werden zum Definieren des Bereichs und Wertebereichs von Funktionen verwendet. Wenn eine Funktion „f“ Werte aus dem Quellbereich „X“ zu Werten im Zielbereich „Y“ abbildet, dann ist die Notation „f:X→Y“. Das Verhältnis zwischen der Funktion und ihrem Quellbereich wird üblicherweise als „Bereich“ bezeichnet. Das Verhältnis zwischen der Funktion und ihrem Zielbereich wird als „Wertemenge“ bezeichnet.

#### 4.42

##### **Geodäte**

##### **geodätische Linie**

(en: geodesic, geodesic line)

<Differentialgeometrie, Geodäsie> Linie auf einer Fläche mit einem tangentialen Krümmungsvektor mit Nulllänge

Anmerkung 1 zum Begriff: Ein Krümmungsvektor einer Geodäte ist senkrecht; die Oberfläche hat daher die minimale Krümmung jeder auf diese Oberfläche eingeschränkten Kurve. Das wird häufig als die kürzeste Abstandskurve zwischen zwei Punkten bezeichnet, dies ist jedoch nicht immer ausreichend, da einige Punkte (insbesondere auf Ellipsoiden und Kugeln) häufig durch mehr als eine Geodäte verbunden sind. Beispielsweise werden auf einem Ellipsoid die Punkte mit  $(\varphi, \lambda) = (0, 0)$  und  $(0, 180)$  durch vier gesonderte Geodäten (zwei polare (die kürzeren) und zwei äquatoriale) verbunden. Die Exponentialabbildung ist nur für einen kleinen Bereich sicher eineindeutig (abhängig von der Lage des Mittelpunkts und der Kurvenform der Oberfläche).

#### 4.43

##### **geodätischer Kreis**

(en: geodesic circle)

<Differentialgeometrie, Geodäsie> Menge von Punkten in gleichem Abstand von einem gegebenen Punkt (auf dem Datum)

Anmerkung 1 zum Begriff: Die geodätischen Kreise, deren Mittelpunkt auf einem (beliebigen) Pol liegt, sind die Linien stetiger geographischer Breite. Kreise in einem Tangentialraum, deren Mittelpunkt im Ursprung (entsprechend des Tangentialpunkts) liegt, werden als geodätische Kreise abgebildet, durch die Exponentialabbildung auf der geometrischen Bezugsfläche, deren Mittelpunkt der Tangentialpunkt ist.

#### 4.44

##### **geometrische Gesamtheit**

(en: geometric aggregate)

Sammlung von Geometrieobjekten ohne innere Struktur

Anmerkung 1 zum Begriff: Es können keine Annahmen über die räumlichen Zusammenhänge zwischen den Elementen getroffen werden.

#### 4.45

##### **geometrische Begrenzung**

(en: geometric boundary)

Begrenzung, die durch eine Menge von geometrischen Primitiven repräsentiert wird, die das Ausmaß eines Geometrieobjekts begrenzen

#### 4.46

##### **geometrischer Komplex**

(en: geometric complex)

Menge disjunkter geometrischer Primitive, bei denen die jeweilige Begrenzung durch eine Vereinigung von anderen geometrischen Primitiven kleinerer Dimension innerhalb derselben Menge dargestellt werden kann

Anmerkung 1 zum Begriff: Die geometrischen Primitive in der Menge sind in dem Sinne disjunkt, dass kein Punkt im Inneren mehr als eines geometrischen Primitivs liegt. Die Menge ist unter Begrenzungsoperationen abgeschlossen, das heißt, es gibt für jedes Element im geometrischen Komplex eine Sammlung (also einen geometrischen Komplex) von geometrischen Primitiven, die die Begrenzung dieses Elements repräsentiert. Es wird daran erinnert, dass die Begrenzung eines Punkts (der einzigen nulldimensionalen Objektart in der Geometrie) leer ist. Wenn die höchste Dimension eines geometrischen Primitivs ein Körper (3D) ist, bricht daher der Begrenzungsoperator in dieser Definition nach höchstens drei Schritten ab. Das gilt auch, wenn die Begrenzung eines beliebigen Objekts ein Zyklus ist.

#### 4.47

##### **geometrische Dimension**

(en: geometric dimension)

<Geometrie, Topologie> größte Zahl  $n$ , sodass jeder Punkt in einer Menge von Punkten mit einer Teilmenge assoziiert werden kann, die den Punkt in ihrem Inneren enthält und die topologisch isomorph zu  $\mathbb{E}^n$ , euklidischer  $n$ -Raum, ist

Anmerkung 1 zum Begriff: Kurven besitzen die geometrische Dimension 1, weil sie stetige Abbildungen eines Abschnitts einer realen Linie sind. Flächen können nicht immer in ihrer Gänze als  $\mathbb{R}^2$  abgebildet werden, aber um jede Punktposition kann eine kleine Umgebung gefunden werden, die (bei stetigen Funktionen) dem Inneren des Einheitskreises in  $\mathbb{R}^2$  gleicht und sie sind daher zweidimensional. In dieser Internationalen Norm werden die meisten Flächen (Instanzen von Fläche) durch ihre definierenden Interpolationsmechanismen als Abschnitte von  $\mathbb{R}^2$ , abgebildet.

#### 4.48

##### **Geometrieobjekt**

(en: geometric object)

<Geometrie> räumliches Objekt, das eine geometrische Menge repräsentiert

Anmerkung 1 zum Begriff: Ein Geometrieobjekt besteht aus einem geometrischen Primitiv, einer Sammlung geometrischer Primitive oder einem geometrischen Komplex, der als Einzelobjekt behandelt wird. Ein Geometrieobjekt kann die räumliche Repräsentation eines Feature-Objekts sein.

#### 4.49

##### **geometrisches Primitiv**

(en: geometric primitive)

<Geometrie> Geometrieobjekt, das ein einzelnes, zusammenhängendes, homogenes (isotropes) Element des Raums darstellt

Anmerkung 1 zum Begriff: Geometrische Primitive sind zusammenhängende Objekte, die Informationen zur geometrischen Konfiguration liefern. Sie umfassen Punkte, Kurven, Oberflächen und Körper. Viele Geometrieobjekte verhalten sich wie Primitive (dieselben Schnittmengen unterstützend, wie für geometrische Primitive definiert); sie setzen sich aber tatsächlich aus einer Anzahl anderer Primitive zusammen. Allgemeine Sammlungen können Gruppen sein und sie können sich möglicherweise nicht wie ein Primitiv verhalten (wie die Linien eines komplexen Netzes, das nicht zusammenhängend ist und sich daher nicht auf eine einzelne Linie zurückführen lässt). Nach dieser Definition ist ein geometrisches Primitiv topologisch offen, weil die Begrenzungspunkte nicht isotrop zu den innenliegenden Punkten sind. Es wird vorausgesetzt, dass die Geometrie abgeschlossen ist. Bei Punkten ist die Begrenzung leer.

#### 4.50

##### **geometrische Realisierung**

(en: geometric realization)

<Geometrie, Topologie> geometrischer Komplex, bei dem die geometrischen Primitive zu den topologischen Primitiven eines topologischen Komplexes in eindeutiger Entsprechung stehen, sodass die Relationen der Begrenzung in beiden Komplexen übereinstimmen

Anmerkung 1 zum Begriff: In einer derartigen Realisierung wird davon ausgegangen, dass die topologischen Primitive das Innere der entsprechenden geometrischen Primitive repräsentieren, obwohl die Primitive als solche abgeschlossen sind.

#### 4.51

##### **geometrische Bezugsfläche**

(en: geometric reference surface)

<Geometrie> Fläche in einem euklidischen Raum, üblicherweise  $\mathbb{E}^3$ , die eine Näherung zu der Fläche der Erde darstellt, möglicherweise beschränkt auf einen kleinen Bereich, aber häufig den gesamten Globus umfassend

#### 4.52

##### **geometrische Menge**

(en: geometric set)

<Geometrie> Menge von Punkten

Anmerkung 1 zum Begriff: Diese Menge ist in den meisten Fällen unendlich, ausgenommen, wenn die Menge aus einer Liste von Punktorten besteht. Stetige Kurven, Flächen und Volumen sind unendliche Mengen von Punkten. Einige Systeme definieren „ausgeartete“ Kurven, die tatsächlich Punkte sind.

#### 4.53

##### **Skalarprodukt**

(en: inner product)

<Vektorgeometrie> bilineare, symmetrische Funktion von Vektorenpaaren  $\langle \vec{v}_1, \vec{v}_2 \rangle \rightarrow \mathbb{R}$  zu einer realen Zahl, wie  $\langle \vec{v}, \vec{v} \rangle = \|\vec{v}\|^2$  und  $\langle \vec{v}_1, \vec{v}_2 \rangle = \|\vec{v}_1\| \|\vec{v}_2\| \cos(\theta)$ , dabei ist „ $\theta$ “ der Winkel zwischen  $\vec{v}_1$  und  $\vec{v}_2$

Anmerkung 1 zum Begriff: In der Differentialgeometrie werden Skalarprodukte auf die Differentiale angewendet, die die lokalen Tangentialräume bilden. In dieser Internationalen Norm werden es üblicherweise Vektoren sein, die tangential zu einer Datumsfläche sind, eingebettet in einem  $\mathbb{E}^3$  euklidischen/kartesischen Raum.

#### 4.54

##### **isolierter Knoten**

(en: isolated node)

<Topologie> Knoten, der mit keiner Kante in Relation steht

Anmerkung 1 zum Begriff: Isolierte Knoten sind üblicherweise kein Gegenstand der meisten algebraischen oder kombinatorischen Topologien. In einem Netzwerk sind sie nicht von Interesse, da sie kein Bestandteil eines Weges sein können. Algebraische Topologie verwendet stets topologische Systeme, da die Begrenzung jedes Objekts der Dimension „ $n$ “ immer eine Vereinigung einer Menge von Objekten der Dimension „ $n-1$ “ ist. Darum ist die Dimension der leeren Menge „ $-1$ “.

#### 4.55

##### **isomorph**

(en: isomorphic)

<Mathematik> Isomorphismus aufweisend

#### 4.56

##### **Isometrie**

(en: isometry)

<Mathematik> Abbildung zwischen metrischen Räumen, die die Metrik wahrt

Anmerkung 1 zum Begriff: Die formale Definition lautet:  $f: X \rightarrow Y \ni \forall x, y \in X. distance(x, y) = distance(f(x), f(y))$ .

#### 4.57

##### **Isomorphismus**

(en: isomorphism)

<Mathematik> Beziehung zwischen zwei Bereichen (z. B. zwei Komplexen), sodass es eindeutige strukturtreue Funktionen von jedem Bereich zum anderen gibt und die Zusammensetzung der beiden Funktionen, in beliebiger Reihenfolge der Identitätsfunktion entspricht

Anmerkung 1 zum Begriff: Ein geometrischer Komplex ist isomorph zu einem topologischen Komplex, wenn ihre Elemente in einer eindeutigen dimensions- und begrenzungsstreuenden Korrespondenz zu einander stehen.

$$[A, B \text{ isomorph}] \Leftrightarrow [\exists f: A \rightarrow B, g: B \rightarrow A \ni f \circ g = Id_A, g \circ f = Id_B] \quad (4)$$

#### 4.58

##### Homomorphismus

(en: homomorphism)

<Mathematik> Beziehung zwischen zwei Bereichen, sodass es eine strukturtreue Funktion von einem Bereich zum anderen gibt

Anmerkung 1 zum Begriff: Ein Homomorphismus unterscheidet sich von einem Isomorphismus dadurch, dass keine inverse Funktion erforderlich ist. In einem Isomorphismus gibt es zwei Homomorphismen, die funktionale Inversen voneinander sind. Stetige Funktionen sind topologische Homomorphismen, weil sie „topologische Eigenschaften“ bewahren. Die Abbildung von topologischen Komplexen zu ihren geometrischen Realisierungen bewahrt das Konzept der Begrenzung und ist daher ein Homomorphismus.

#### 4.59

##### Maximum

max

kleinste obere Schranke

lub

(en: maximum, least upper bound)

<Mathematik> kleinstes Element, dass größer gleich allen Elementen einer Menge ist, die in einem geordneten Bereich enthalten ist ( $\leq$ )

Anmerkung 1 zum Begriff:  $[\forall a \in A \Rightarrow \max(A) \geq a] \wedge [\forall b \ni [(b \ni [\forall a \in A \Rightarrow b \geq a] \Rightarrow [\max(A) \leq b]]]$

Jede Zahl ist eine obere Grenze von  $\emptyset$  (leere Menge) als eine Menge von Zahlen, weil jede gegebene Zahl größer als jede Zahl in  $\emptyset$  ist (eine zugelassene gegenstandslose, aber dennoch wahre Aussage, weil es keine Zahl in  $\emptyset$  gibt). Das bedeutet, dass das  $\max(\emptyset)$  kleiner sein muss, als jede Zahl; daher „ $-\infty$ “.

#### 4.60

##### Minimum

min

größte untere Schranke

glb

(en: minimum, greatest lower bound)

<Mathematik> größtes Element, dass kleiner gleich allen Elementen einer Menge ist, die in einem geordneten Bereich enthalten ist ( $\leq$ )

Anmerkung 1 zum Begriff:  $[\forall a \in A \Rightarrow \min(A) \leq a] \wedge [\forall b \ni [(b \ni [\forall a \in A \Rightarrow b \leq a] \Rightarrow [\min(A) \geq b]]]$

Jede Zahl ist eine untere Schranke von  $\emptyset$ , betrachtet als eine Menge von Zahlen, wede jede gegebene Zahl geringer ist als jede Zahl in  $\emptyset$  (eine zugelassene gegenstandslose, aber dennoch wahre Aussage, weil es keine Zahl in  $\emptyset$  gibt). Das bedeutet, dass das  $\min(\emptyset)$  größer sein muss, als jede Zahl; daher „ $+\infty$ “.

#### 4.61

##### monoton

(en: monotonic)

<Mathematik> weder steigend noch fallend

Anmerkung 1 zum Begriff: Eine steigende Folge nimmt nicht ab. Eine streng steigende Folge nimmt immer zu. Eine fallende Folge nimmt nicht zu. Eine streng fallende Folge nimmt immer ab.

$$[x(i) \text{ ist steigend}] \Leftrightarrow [i < j \Rightarrow x(i) \leq x(j)]$$

$$[x(i) \text{ ist streng steigend}] \Leftrightarrow [i < j \Rightarrow x(i) < x(j)]$$

$$[x(i) \text{ ist fallend}] \Leftrightarrow [i < j \Rightarrow x(i) \geq x(j)] \quad (5)$$

$$[x(i) \text{ ist streng fallend}] \Leftrightarrow [i < j \Rightarrow x(i) > x(j)]$$

$$[x(i) \text{ ist fallend}] \vee [x(i) \text{ ist steigend}] \Leftrightarrow [x(i) \text{ ist monoton}]$$

#### 4.62

##### **n-Zelle**

(en: n-disc)

<Topologie, Geometrie> Geometrie, die isomorph ist zur Menge der Punkte  $X$  in  $\mathbb{E}^n$ , sodass  $\|X\| \leq 1$ ; Menge aller Punkte in  $\mathbb{E}^n$  kleiner gleich einer Einheit des Abstands vom Ursprung

Anmerkung 1 zum Begriff: Die 0-Zelle ist ein Punkt. Die 1-Zelle ist eine Linie. Die 2-Zelle ist ein Kreis und sein Inneres. Die 3-Zelle ist eine Kugel und ihr Inneres.

#### 4.63

##### **n-Simplex**

(en: n-simplex)

<Geometrie, Topologie> konvexe Hülle von  $n+1$  Punkten an allgemeiner Position in einem mindestens  $n$ -dimensionalen Raum oder eine topologisch isomorphe Abbildung einer derartigen Geometrie

Anmerkung 1 zum Begriff: Ein  $n$ -Simplex ist  $n$ -dimensional und topologisch isomorph zu einer  $n$ -Zelle. Ein 0-Simplex ist ein Punkt. Ein 1-Simplex ist eine Kurve (üblicherweise eine Linie) mit zwei 0-Simplexen (Punkten) auf ihrer Begrenzung; ein 2-Simplex ist eine Dreiecksfläche mit drei 1-Simplexen auf seiner Begrenzung. Im Allgemeinen weist die Begrenzung eines  $n$ -Simplex  $n$   $(n-1)$ -Simplexe auf. Ein  $n$ -Simplex und eine  $n$ -Zelle sind topologisch isomorph. Die Begrenzung eines  $n$ -Simplex ist isomorph zu einer  $(n-1)$ -Sphäre. In allen Fällen repräsentiert die vorangestellte Zahl die topologische Dimension des Objekts.

#### 4.64

##### **n-Sphäre**

(en: n-sphere)

<Geometrie, Topologie> Geometrie, die isomorph zur Menge der Punkte  $X$  in  $\mathbb{E}^{n+1}$  ist, sodass  $\|X\| = 1$  gilt; Menge aller Punkte in  $\mathbb{E}^{n+1}$ , in einer Einheit Abstand vom Ursprung

Anmerkung 1 zum Begriff: Eine  $n$ -Sphäre ist isomorph zur Begrenzung einer  $n+1$ -Zelle.

#### 4.65

##### **Umgebung**

(en: neighborhood)

<Topologie, metrische Räume> offene Menge von Punkten, die einen spezifischen Punkt in ihrem Inneren enthält

#### 4.66

##### **Knoten**

(en: node)

<Topologie> nulldimensionales topologisches Primitiv

Anmerkung 1 zum Begriff: Die Begrenzung eines Knotens ist die leere Menge.

#### 4.67

##### **Normalkrümmungsvektor**

(en: normal curvature vector)

<Differentialgeometrie, Geodäsie> Projektion des Krümmungsvektors der Kurve lotrecht zur Tangentialebene zu der Fläche an dem Punkt

Anmerkung 1 zum Begriff: Der Normalkrümmungsvektor einer Kurve ist ausschließlich abhängig von der Kurve und der Fläche (die eine Nebenbedingung der Kurve ist). Der Normalkrümmungsvektor einer auf eine Fläche beschränkten Kurve ist parallel zum Normalenvektor der Fläche. Falls der Normalkrümmungsvektor überall gleich dem Krümmungsvektor der Kurve ist, dann ist die Kurve eine Geodäte.

#### 4.68

##### **Normalenvektor**

(en: normal)

<Differentialgeometrie, Geodäsie> lotrechter Vektor (orthogonal) zum Geometrieobjekt (Kurve oder Fläche) in dem Punkt

Anmerkung 1 zum Begriff: Der Normalenvektor ist parallel zu dem „aufsteigenden“ Normal bei Flächen in  $\mathbb{E}^3$ , definiert in der Flächen-Schnittmenge nach 6.4.17.2.

#### 4.69

##### **normale Schnittkurve**

(en: normal section curve)

<Differentialgeometrie, Geodäsie> ebenes Kurvensegment, das das Normal in einem ihrer Endpunkte enthält

Anmerkung 1 zum Begriff: Die übliche Konstruktion beginnt, indem einer der Endpunkte, das Normal zur Fläche an diesem Endpunkt und die Position des anderen Endpunkts gewählt werden und eine Ebene in  $\mathbb{E}^3$  erstellt wird, die dann vom Datum geschnitten wird. Diese Kurve ist eine Normalschnittkurve dieser beiden Punkte, die das Normal zur Fläche am ersten Punkt enthält. Auf der Kugel ist das die Geodäte und die beiden Normalschnittkurven sind gleich. Auf einem ellipsoiden Datum ist dasselbe für zwei Punkte auf dem Äquator oder für zwei Punkte auf derselben Meridianebene wahr, aber im Allgemeinen ist dies falsch. Der Grund dafür ist, dass in diesen Fällen die Ebene durch die beiden Punkte und den Mittelpunkt des Ellipsoids Symmetrieebenen für das Ellipsoid sind, die das Ellipsoid in symmetrische „spiegelbildliche“ Hälften teilen. Im Allgemeinen liegt die Geodäte zwischen den beiden Punkten zwischen den beiden Normalschnittkurven.

#### 4.70

##### **offene Menge**

(en: open set)

<Metrik, Topologie, Geometrie> eine Metrik enthaltend oder topologisch offene Umgebung von jedem ihrer Punkte

Anmerkung 1 zum Begriff: In einem metrischen Raum ist eine Menge  $X$  offen, wenn jeder Punkt „ $x$ “ in der Menge in einer kleinen Kugel enthalten ist, die eine Teilmenge ist von  $X$ :  
[ $X$  ist offen]  $\Leftrightarrow [x \in X] \Rightarrow [\exists \varepsilon > 0 \ni \text{Abstand}(x, y) < \varepsilon] \Rightarrow [y \in X]$ . Eine Topologie ist eine Menge aus Teilmengen eines Raums, die als offen angesehen werden.

#### 4.71

##### **Zerlegung der Eins**

(en: partition of unity)

<Mathematik> Menge von realwertigen Funktionen über den gesamten Bereich, deren arithmetische Summe bei jedem Bereichswert 1 beträgt.

#### 4.72

##### **wegweise zusammenhängend**

(en: path connected)

Eigenschaft eines Geometrieobjekts, die besagt, dass beliebige zwei Punkte auf dem Objekt auf einer Kurve liegen können, die vollständig innerhalb des Objekts verläuft

Anmerkung 1 zum Begriff: Es gibt allgemeinere Geometrien, die zusammenhängend, aber nicht wegweise zusammenhängend sind, diese sind jedoch nicht als Sammlungen von „digitalen“ Geometrien, wie nach diesem Dokument festgelegt, darstellbar. Keines der Geometrieobjekte, die mit den Verfahren nach dieser Internationalen Norm auf einem standardmäßigen Digitalrechner konstruiert werden können, kann zusammenhängend, jedoch nicht wegweise zusammenhängend sein. Siehe die Anmerkungen zum Begriff „zusammenhängend“.

#### 4.73

##### **ebenes Kurvensegment**

##### **ebene Kurve**

(en: plane curve segment, plane curve)

<Geometrie> Kurve in  $\mathbb{E}^3$ , die in einer Ebene enthalten ist

Anmerkung 1 zum Begriff: In einem zweidimensionalen Koordinatensystem wird die Prüfung der Gültigkeit einer ebenen Kurve im dreidimensionalen Raum der geometrische Bezugsfläche oder des Datums des Koordinatenreferenzsystems durchgeführt.

#### 4.74

##### **planarer topologischer Komplex**

(en: planar topological complex)

<Topologie> topologischer Komplex, der eine geometrische Realisierung aufweist, die in einem euklidischen 2-Raum ( $\mathbb{E}^2$ ) eingebettet sein kann

#### 4.75

##### **pythagoräische Metrik**

(en: pythagorean metric)

<euklidische Geometrie> Abstandsmaß in einem  $\mathbb{E}^n$ -Koordinatenraums, unter Verwendung eines quadratischen Mittelwerts (en: root-mean sum) der Differenzen zwischen den einzelnen Koordinatenversätzen

Anmerkung 1 zum Begriff:  $P = (p_i), Q = (q_i), distance(P, Q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$ . Die Nachweise der pythagoräischen Metriken sind sämtlich von der lokalen „Ebenheit“ des Raums abhängig. Kartesische Koordinatenräume, die pythagoräische Metriken aufweisen, werden euklidische Räume genannt ( $\mathbb{E}^n$ ). Im Bereich der Koordinatenreferenzsysteme sind ausschließlich „ingenieurtechnische Koordinatensysteme“ euklidisch. Jedes CRS, das ein gekrümmtes Datum verwendet, ist nach Definition nichteuklidisch und kann nicht „wahrheitsgetreu“ pythagoräische Metriken verwenden, ausgenommen zu Zwecken der Näherung. Diese Näherungen gelten für topologische Aussagen, aber nicht für tatsächliche Messwerte ohne Anpassungen.

#### 4.76

##### **Loxodrome**

(en: rhumb line, loxodrome)

<Geometrie, Navigation> Kurve, die Meridiane der geographischen Länge bei einer konstanten Peilung schneidet

#### 4.77

##### **Wertebereich**

##### **Wertemenge**

(en: range, co-domain)

<Mathematik> akzeptable Zielwerte einer Funktion

#### 4.78

##### **zeilenweise Form**

(en: row-major form)

<Mathematik, Informatik> Speichermechanismus für mehrdimensionale Matrix-Gruppierungen in linearem Speicher, der diese so organisiert, dass jede Zeile an aufeinanderfolgenden Stellen gespeichert wird, sowie dass die vollständigen Zeilen hintereinander gespeichert werden und fortlaufend in ähnlicher Weise für jeden zusätzlichen Index

Anmerkung 1 zum Begriff: Wenn die Indizes (i, j) mit der Anzahl der Zeilen „r“ und der Spalten „c“ sind, dann wird die Abbildung von den mehrdimensionalen Orten zu den linearen Speicherorten angegeben durch:

$$\begin{aligned} [i, j \in \mathbb{Z} \ni 1 \leq i \leq r; 1 \leq j \leq c] &\Rightarrow [(i, j) \rightarrow (i - 1)c + j] \\ [i, j, k \in \mathbb{Z} \ni 1 \leq i \leq r; 1 \leq j \leq c; 1 \leq k \leq f] &\Rightarrow [(i, j, k) \rightarrow ((i - 1)c + j - 1)f + k] \end{aligned} \quad (6)$$

Anmerkung 1 zum Begriff: Die Matrix  $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$  wird in zeilenweiser Form als [1 2 3 4 5 6 7 8 9] gespeichert.

Für höhere Dimensionen wird dasselbe Muster rekursiv angewendet.

#### 4.79

##### **zweites geodätisches Problem**

##### **inverses geodätisches Problem**

(en: second geodetic problem, inverse geodetic problem)

<Differentialgeometrie> Problem, dass zwei gegebene Punkte die Anfangsrichtung und Länge einer sie verbindenden Geodäte bestimmen

Anmerkung 1 zum Begriff: Siehe erstes geodätisches Problem, das diese Operation umkehrt.

#### 4.80

##### **Abschnitt, Segment**

(en: segment)

<Topologie, Geometrie> kleinster Teil einer Geometrie, üblicherweise als Teil einer Komposition

#### 4.81

##### **einfach**

(en: simple)

<Topologie, Geometrie> homogen (alle Punkte haben isomorphe Umgebungen) und mit einer einfachen Begrenzung

Anmerkung 1 zum Begriff: Das Innere ist überall lokal isomorph zu einer offenen Zelle in einem euklidischen Koordinatenraum mit der entsprechenden Dimension  $D^n = \{P | \|P\| < 1.0\}$ . Die Begrenzung ist um eine Dimension kleiner. Das meint im Wesentlichen, dass das Objekt sich selbst weder schneidet noch berührt. Wird im Allgemeinen für eine Kurve verwendet, die, mit der möglichen Ausnahme von Begrenzungspunkten, sich selbst weder schneidet noch berührt. Einfache abgeschlossene Kurven sind isomorph zu einem Kreis.

#### 4.82

##### **räumlich dimensioniert, adj.**

(en: spatial dimension)

<Topologie, Geometrie> Anzahl der unabhängigen Entscheidungen in einem Koordinatensystem, die erforderlich ist, um eine Position zu verorten

Anmerkung 1 zum Begriff: Diese Definition ist logisch äquivalent zur topologischen Dimension der räumlichen Projektion des CRS. Sie beschreibt den Raum als ein Ziel der Geometrie. Grammatikalisch kann der Begriff ein Nomen sein, aber zur Beschreibung von Raum verwendet werden, wie: „die räumliche Dimension von CRS84 ist 2“.

#### 4.83

##### **räumliche Dimension, Nomen**

(en: spatial dimension)

<Topologie, Geometrie> jede in einem Koordinatensystem getroffene unabhängige Entscheidung, um eine Position zu verorten

Anmerkung 1 zum Begriff: „Ein euklidischer Raum mit einer räumlichen Dimension von 3,  $\mathbb{E}^3$ , verwendet üblicherweise mit „x“, „y“ und „z“ bezeichnete Achsen. Ihre erste räumliche Dimension ist „x“, die zweite ist „y“ und die dritte ist „z“.“ Im raumbezogenen Kontext beschreibt die adjektivische Verwendung den Raum, aber das Singular-Nomen kann eine Achse gesondert bezeichnen.

#### 4.84

##### **Objekt mit Raumbezug**

(en: spatial object)

<Topologie, Geometrie> Objekt zur Darstellung einer räumlichen Eigenschaft eines Features

#### 4.85

##### **Operator mit Raumbezug**

(en: spatial operator)

<Topologie, Geometrie> Funktion oder Verfahren, die/das mindestens einen Parameter mit Raumbezug in ihrem/seinem Bereich oder Wertebereich aufweist

Anmerkung 1 zum Begriff: Jede UML-Operation an einem Objekt mit Raumbezug würde als ein Operator mit Raumbezug eingestuft werden, wie die Abfrageoperatoren nach dieser Internationalen Norm, Abschnitt 8.

#### 4.86

##### **Startknoten**

(en: start node)

<Topologie, Graphentheorie> Knoten in der Begrenzung einer Kante, die dem Anfangspunkt dieser Kante als Kurve entspricht

#### 4.87

##### **Anfangspunkt**

(en: start point)

<Geometrie> erster Punkt einer Kurve

#### 4.88

##### **Teilkomplex**

(en: subcomplex)

<Topologie, Geometrie> Komplex, dessen sämtliche Elemente auch in einem größeren Komplex enthalten sind

Anmerkung 1 zum Begriff: Da die Definitionen von geometrischem Komplex und topologischem Komplex ausschließlich erfordern, dass diese unter Begrenzungsoperationen abgeschlossen sind, ist die Menge aller Primitive einer bestimmten Dimension und darunter immer ein Teilkomplex des ursprünglichen, größeren Komplexes. Folglich enthält jeder vollständig planare topologische Komplex einen Kante-Knoten-Graph als einen Teilkomplex.

#### 4.89

##### **Tangente**

(en: tangent)

<Differentialgeometrie, Infinitesimalrechnung> Richtung, die die instantane Richtung einer Kurve in einem Punkt angibt

Anmerkung 1 zum Begriff: Die Tangente wird üblicherweise durch Ableitung einer funktionalen Darstellung einer Kurve berechnet, aber sie kann mit einer Sekante (zwei Schnittpunkte), die von dem Punkt einen anderen nahegelegenen Punkt auf der Kurve durchläuft, genähert werden. Je näher der zweite Punkt dem ersten liegt, desto besser ist die Annäherung an die Richtung der Tangente.

#### 4.90

##### **tangentialem Krümmungsvektor**

##### **geodätischer Krümmungsvektor**

(en: tangential curvature vector, geodesic curvature vector)

<Differentialgeometrie, Geodäsie> Projektion des Krümmungsvektors einer Kurve auf der Tangentialebene zur Fläche in dem Punkt

#### 4.91

##### **Tangentialraum**

##### **Tangentialebene**

(en: tangent space, tangent plane)

Sammlung von Tangentialvektoren bei Kurven, die durch den Punkt verlaufen

#### 4.92

##### **Tangentialvektor**

(en: tangent vector)

erste Ableitung einer durch die Bogenlänge parametrisierten Kurve

Anmerkung 1 zum Begriff: Wenn  $c(s) = (x(s), y(s), z(s))$  eine Kurve in einem dreidimensionalen kartesischen Raum ( $\mathbb{E}^3$ ) ist und  $s$  die Bogenlänge entlang  $c$  ist, dann ist die Tangente  $\vec{t}(s) = \dot{c}(s) = (\dot{x}(s), \dot{y}(s), \dot{z}(s))$ , d. h. die Ableitung der Koordinatenwerte von  $c$  in Bezug auf  $s$ . Der Krümmungsvektor beträgt  $\kappa(s) = \ddot{c}(s) = (\ddot{x}(s), \ddot{y}(s), \ddot{z}(s))$ .

#### 4.93

##### **topologische Begrenzung**

(en: topological boundary)

<Geometrie, Topologie> Begrenzung, repräsentiert durch eine Menge von orientierten topologischen Primitiven von kleinerer topologischer Dimension, die die Ausdehnung eines Topologieobjekts begrenzt oder durch geometrische Realisierung einer solchen Menge

Anmerkung 1 zum Begriff: Die Begrenzung eines topologischen Komplexes ist die Begrenzung der geometrischen Realisierung des topologischen Komplexes.

#### 4.94

##### **topologischer Komplex**

(en: topological complex)

<Geometrie, Topologie> Sammlung topologischer Primitive, die unter den Begrenzungsoperationen abgeschlossen ist

Anmerkung 1 zum Begriff: Abgeschlossen unter Begrenzungsoperation bedeutet, dass wenn ein topologisches Primitiv in dem topologischen Komplex liegt, auch seine Begrenzungsobjekte im topologischen Komplex liegen.

#### 4.95

##### **topologischer Ausdruck**

(en: topological expression)

<Geometrie, Topologie> Sammlung orientierter topologischer Primitive, die wie ein multivariates Polynom behandelt wird

Anmerkung 1 zum Begriff: Topologische Ausdrücke werden für viele Berechnungen in der rechentechnischen Topologie angewendet.

#### 4.96

##### **Topologieobjekt**

(en: topological object)

<Geometrie, Topologie> räumliches Objekt, das räumliche Eigenschaften darstellt, die unter stetiger Transformation invariant sind

Anmerkung 1 zum Begriff: Ein Topologieobjekt ist ein topologisches Primitiv oder ein topologischer Komplex.

#### 4.97

##### **topologisches Primitiv**

(en: topological primitive)

<Geometrie, Topologie> Topologieobjekt, das ein einzelnes, homogenes, unzerlegbares Element darstellt

Anmerkung 1 zum Begriff: Ein topologisches Primitiv entspricht dem Inneren eines geometrischen Primitivs derselben Dimension in einer geometrischen Realisierung.

**4.98**

**topologischer Körper**

(en: topological solid)

<Geometrie, Topologie> dreidimensionales topologisches Primitiv

Anmerkung 1 zum Begriff: Die Begrenzung eines topologischen Körpers besteht aus einer Menge von gerichteten Maschen.

**4.99**

**erzwungene Typumwandlung**

(en: type coercion)

<Programmierung> Umwandlung einer Art von Wert in einen andersartigen Wert mit ähnlichem Inhalt

Anmerkung 1 zum Begriff: Punkt und direkte Position sind in diesem Kontext informationstechnisch identisch (in eindeutiger Relation zueinander). Bei einer gegebenen direkten Position, kann ein Punkt konstruiert werden. Bei einem gegebenen Punkt, kann aus seinen Koordinaten eine direkte Position abgeleitet werden. Wird die erzwungene Typumwandlung unterstützt, kann ein Punkt dort verwendet werden, wo eine direkte Position abgefragt wurde und umgekehrt. Die meisten Programmiersprachen verwenden eine erzwungene Typumwandlung, andere verwenden jedoch „Cast“-Operatoren (eine Art von Konstruktor), die erfordern, dass der Programmierer die erzwungene Typumwandlung startet. „Stark substanziell“ bezieht sich darauf, dass eine Subtyp-Instanz immer in einen ihrer Supertypen umgewandelt werden kann.

— Leerseite —

## Contents

|                                                         |     |
|---------------------------------------------------------|-----|
| Foreword .....                                          | xv  |
| Introduction.....                                       | xvi |
| 1 Scope .....                                           | 1   |
| 2 Conformance .....                                     | 1   |
| 2.1 Requirements class conformance targets .....        | 1   |
| 2.1.1 Geometry metrics (geodesy) .....                  | 3   |
| 2.1.2 Topological dimensionality .....                  | 4   |
| 2.1.3 Interpolation schemes .....                       | 4   |
| 2.1.4 Structural complexity .....                       | 5   |
| 2.1.5 Functional complexity .....                       | 5   |
| 2.2 Conformance classes .....                           | 6   |
| 2.3 Requirements classes .....                          | 7   |
| 3 Normative references .....                            | 8   |
| 4 Terms and definitions.....                            | 9   |
| 5 Symbols and abbreviated terms .....                   | 25  |
| 5.1 Presentation and notation .....                     | 25  |
| 5.1.1 Unified Modeling Language (UML) .....             | 25  |
| 5.1.2 Naming conventions .....                          | 25  |
| 5.2 Organization.....                                   | 25  |
| 5.3 Abbreviated terms .....                             | 26  |
| 6 Coordinates and Core Geometry .....                   | 27  |
| 6.1 Semantics .....                                     | 27  |
| 6.2 Requirements Class Coordinate.....                  | 27  |
| 6.2.1 Codelists to specify capabilities .....           | 27  |
| 6.2.2 Coordinate systems for Geometry — Semantics ..... | 28  |
| 6.2.2.1 Coordinates, points and locations .....         | 28  |
| 6.2.2.2 Coordinate systems .....                        | 29  |
| 6.2.2.3 Metrics and distance measures .....             | 31  |
| 6.2.2.4 Underlying geometric space .....                | 31  |
| 6.2.3 GeometricReferenceSurface.....                    | 32  |
| 6.2.3.1 Interface GeometricReferenceSurface.....        | 32  |
| 6.2.3.2 CodeList: GeometricSurfaceType.....             | 33  |
| 6.2.3.3 Interface PlaneMap.....                         | 34  |
| 6.2.3.4 Interface Sphere.....                           | 34  |
| 6.2.3.5 Interface Ellipsoid .....                       | 35  |
| 6.2.3.6 Interface GeoSurface .....                      | 35  |
| 6.2.3.7 Interface Geoid .....                           | 35  |
| 6.2.4 Interface ReferenceSystem .....                   | 35  |
| 6.2.4.1 Semantics .....                                 | 35  |
| 6.2.4.2 Attribute id .....                              | 36  |
| 6.2.4.3 Attribute rsid .....                            | 36  |
| 6.2.4.4 Attribute dimension .....                       | 36  |
| 6.2.4.5 Attribute axis .....                            | 36  |
| 6.2.4.6 Attribute datum .....                           | 36  |
| 6.2.4.7 Role parameter .....                            | 37  |
| 6.2.5 CodeList ReferenceSystemTypes .....               | 37  |
| 6.2.6 Interface CompoundReferenceSystem .....           | 37  |
| 6.2.6.1 Semantics .....                                 | 37  |
| 6.2.6.2 Role projection .....                           | 37  |

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| 6.2.6.3 Attribute rsid .....                                                 | 37 |
| 6.2.7 Interface HomogeneousCoordinateSystem .....                            | 38 |
| 6.2.8 Interface GeometricCoordinateSystem .....                              | 38 |
| 6.2.8.1 Semantics .....                                                      | 38 |
| 6.2.8.2 Attribute name .....                                                 | 38 |
| 6.2.8.3 Attribute spatialDimension .....                                     | 38 |
| 6.2.8.4 Attribute temporalDimension .....                                    | 39 |
| 6.2.8.5 Attribute parametricDimension .....                                  | 39 |
| 6.2.8.6 Attribute permutation .....                                          | 39 |
| 6.2.8.7 Attribute spatialProjection .....                                    | 40 |
| 6.2.8.8 Attribute temporalProjection .....                                   | 40 |
| 6.2.8.9 Operation: csDistance .....                                          | 40 |
| 6.2.8.10 Operation: distance .....                                           | 40 |
| 6.2.8.11 Operation: pointAtDistance .....                                    | 40 |
| 6.2.8.12 Operation: geoLocate .....                                          | 41 |
| 6.2.8.13 Operation: bearing .....                                            | 41 |
| 6.2.8.14 Operation: csInnerProduct .....                                     | 41 |
| 6.2.9 Datatype DirectPosition .....                                          | 42 |
| 6.2.9.1 Semantics .....                                                      | 42 |
| 6.2.9.2 Attribute rsid .....                                                 | 42 |
| 6.2.9.3 Attribute dimension .....                                            | 42 |
| 6.2.9.4 Attribute coordinate .....                                           | 42 |
| 6.2.9.5 Derived Role referenceSystem .....                                   | 42 |
| 6.2.9.6 Operation: spatialProjection .....                                   | 42 |
| 6.2.9.7 Operation: pointProjection .....                                     | 43 |
| 6.2.9.8 Operations derivable from Point by type coercion .....               | 43 |
| 6.2.9.8.1 Semantics .....                                                    | 43 |
| 6.2.9.8.2 Operation: vectorToPoint (toPoint: DirectPosition): Vector .....   | 44 |
| 6.2.9.8.3 Operation: pointAtDistance (bearing: Vector): DirectPosition ..... | 44 |
| 6.2.10 Union Datatype RSID .....                                             | 44 |
| 6.2.11 Codelist Axis .....                                                   | 44 |
| 6.2.12 Role metadata: AxisDescription .....                                  | 44 |
| 6.2.13 Datatype Axis Description .....                                       | 44 |
| 6.2.14 Codelist SpatialAxis .....                                            | 45 |
| 6.2.15 Codelist SphericalAxis .....                                          | 45 |
| 6.2.16 Codelist TemporalAxis .....                                           | 45 |
| 6.2.17 Codelist ParametricAxis .....                                         | 46 |
| 6.2.18 Codelist Datum .....                                                  | 46 |
| 6.2.19 Datatype Parameter .....                                              | 46 |
| 6.2.19.1 Semantics .....                                                     | 46 |
| 6.2.19.2 Attribute name .....                                                | 46 |
| 6.2.19.3 Attribute value .....                                               | 47 |
| 6.2.20 Datatype Permutation, Projection .....                                | 47 |
| 6.2.20.1 Semantics .....                                                     | 47 |
| 6.2.20.2 Attribute outOrder .....                                            | 47 |
| 6.2.20.3 Attribute axis .....                                                | 47 |
| 6.2.21 Interface ReferenceDirection .....                                    | 47 |
| 6.2.22 Datatype Bearing .....                                                | 47 |
| 6.2.22.1 Semantics .....                                                     | 47 |
| 6.2.22.2 Attribute angle .....                                               | 48 |
| 6.2.22.3 Attribute direction .....                                           | 49 |
| 6.2.22.4 Attribute reference: ReferenceDirection .....                       | 49 |
| 6.2.22.5 Attribute rotation: Rotation .....                                  | 49 |
| 6.2.22.6 Constructor: bearing .....                                          | 49 |
| 6.2.23 Codelist Rotation .....                                               | 49 |

|          |                                                                               |    |
|----------|-------------------------------------------------------------------------------|----|
| 6.2.24   | Codelist RelativeDirection .....                                              | 49 |
| 6.2.25   | Codelist FixedDirection .....                                                 | 49 |
| 6.2.26   | Codelist CurveRelativeDirection .....                                         | 49 |
| 6.2.27   | Datatype Vector .....                                                         | 50 |
| 6.2.27.1 | Semantics .....                                                               | 50 |
| 6.2.27.2 | Attribute origin .....                                                        | 50 |
| 6.2.27.3 | Attribute offset .....                                                        | 50 |
| 6.2.27.4 | Attribute dimension .....                                                     | 50 |
| 6.2.27.5 | Attribute coordinateSystem .....                                              | 51 |
| 6.2.27.6 | Operations: CrossProduct, DotProduct .....                                    | 51 |
| 6.2.28   | Interface Envelope .....                                                      | 51 |
| 6.2.28.1 | Semantics .....                                                               | 51 |
| 6.2.28.2 | Attribute upperCorner .....                                                   | 52 |
| 6.2.28.3 | Attribute lowerCorner .....                                                   | 52 |
| 6.2.29   | Engineering coordinate systems, Tangent spaces and local interpolations ..... | 52 |
| 6.3      | Requirements Class Coordinate Data .....                                      | 53 |
| 6.4      | Requirements Class Geometry .....                                             | 53 |
| 6.4.1    | Semantics .....                                                               | 53 |
| 6.4.2    | Interface TransfiniteSetOfDirectPositions .....                               | 54 |
| 6.4.3    | CodeList: BoundaryType .....                                                  | 54 |
| 6.4.4    | Interface Geometry .....                                                      | 55 |
| 6.4.4.1  | Semantics .....                                                               | 55 |
| 6.4.4.2  | Attribute isEmpty .....                                                       | 56 |
| 6.4.4.3  | Attribute asBinary .....                                                      | 57 |
| 6.4.4.4  | Attribute asGML: XML .....                                                    | 57 |
| 6.4.4.5  | Attribute asText: CharacterString .....                                       | 57 |
| 6.4.4.6  | Attribute asData: GeometryData .....                                          | 57 |
| 6.4.4.7  | Attribute boundary: Geometry .....                                            | 57 |
| 6.4.4.8  | Attribute centroid .....                                                      | 58 |
| 6.4.4.9  | Attribute closure .....                                                       | 58 |
| 6.4.4.10 | Attribute convexHull .....                                                    | 59 |
| 6.4.4.11 | Attribute coordinateDimension .....                                           | 59 |
| 6.4.4.12 | Attribute envelope .....                                                      | 59 |
| 6.4.4.13 | Attribute is3D: Boolean .....                                                 | 59 |
| 6.4.4.14 | Attribute isCycle .....                                                       | 60 |
| 6.4.4.15 | Attribute isSimple .....                                                      | 60 |
| 6.4.4.16 | Attribute isValid .....                                                       | 61 |
| 6.4.4.17 | Attribute maximalComplex .....                                                | 61 |
| 6.4.4.18 | Attribute metadata: URI[0..*] .....                                           | 61 |
| 6.4.4.19 | Attribute representativePoint: DirectPosition .....                           | 62 |
| 6.4.4.20 | Attribute rsid: RSID [1..*] .....                                             | 62 |
| 6.4.4.21 | Attribute spatialDimension: Integer .....                                     | 62 |
| 6.4.4.22 | Attribute topologicalDimension: Integer .....                                 | 62 |
| 6.4.4.23 | Attribute type: GeometryType [1..*] .....                                     | 63 |
| 6.4.4.24 | Operation buffer (radius: Distance): Geometry .....                           | 63 |
| 6.4.4.25 | Operation: dimension (point: DirectPosition = NULL): Integer .....            | 63 |
| 6.4.4.26 | Operation: distance .....                                                     | 64 |
| 6.4.4.27 | Operation equal .....                                                         | 64 |
| 6.4.4.28 | Operation transform .....                                                     | 64 |
| 6.4.4.29 | Constructors Geometry .....                                                   | 65 |
| 6.4.4.30 | Operations from TransfiniteSet realization .....                              | 67 |
| 6.4.5    | Datatype GeometryData .....                                                   | 68 |
| 6.4.6    | CodeList: GeometryType .....                                                  | 68 |
| 6.4.7    | Interface Encoding .....                                                      | 68 |
| 6.4.8    | Interface Query2D .....                                                       | 69 |

|                                                                                                              |    |
|--------------------------------------------------------------------------------------------------------------|----|
| 6.4.8.1 Semantics .....                                                                                      | 69 |
| 6.4.8.2 Operation: distance .....                                                                            | 69 |
| 6.4.8.3 Operation: buffer (distance: Distance): Geometry .....                                               | 69 |
| 6.4.8.4 Set operation: intersection .....                                                                    | 69 |
| 6.4.8.5 Set operation: difference .....                                                                      | 70 |
| 6.4.8.6 Set operation: symDifference .....                                                                   | 70 |
| 6.4.8.7 Set operation: union .....                                                                           | 70 |
| 6.4.8.8 Map query operations .....                                                                           | 71 |
| 6.4.9 Interface Query3D .....                                                                                | 72 |
| 6.4.10 Interface Empty .....                                                                                 | 73 |
| 6.4.11 Interface Primitive .....                                                                             | 74 |
| 6.4.11.1 Semantics .....                                                                                     | 74 |
| 6.4.11.2 Association: Role segment .....                                                                     | 75 |
| 6.4.12 Datatype PrimitiveData .....                                                                          | 75 |
| 6.4.13 Interface Point .....                                                                                 | 76 |
| 6.4.13.1 Semantics .....                                                                                     | 76 |
| 6.4.13.2 Attribute position: DirectPosition .....                                                            | 76 |
| 6.4.13.3 Attribute boundary: Geometry = Empty .....                                                          | 76 |
| 6.4.13.4 Operation: vectorToPoint (toPoint: DirectPosition): Vector .....                                    | 77 |
| 6.4.13.5 Operation: Bearing (toPoint: DirectPosition): Bearing .....                                         | 77 |
| 6.4.13.6 Operation: pointAtDistance (bearing: Vector): DirectPosition .....                                  | 77 |
| 6.4.13.7 Constructor: Point (pt: DirectPosition): Point Constructor: Point (data:<br>PointData): Point ..... | 77 |
| 6.4.14 Datatype PointData .....                                                                              | 78 |
| 6.4.15 Interface Orientable .....                                                                            | 78 |
| 6.4.15.1 Semantics .....                                                                                     | 78 |
| 6.4.15.2 Attribute orientation: sign .....                                                                   | 78 |
| 6.4.15.3 Association Roles: proxy and primitive operation reverse (): Orientable .....                       | 79 |
| 6.4.16 Datatype OrientableData .....                                                                         | 79 |
| 6.4.17 Datatype Knot .....                                                                                   | 79 |
| 6.4.17.1 Semantics .....                                                                                     | 79 |
| 6.4.17.2 Attribute value: Real .....                                                                         | 80 |
| 6.4.17.3 Attribute multiplicity: Integer .....                                                               | 80 |
| 6.4.18 Interface Curve .....                                                                                 | 80 |
| 6.4.18.1 Semantics .....                                                                                     | 80 |
| 6.4.18.2 Attribute controlPoint .....                                                                        | 83 |
| 6.4.18.3 Attribute dataPoint .....                                                                           | 84 |
| 6.4.18.4 Attribute knot .....                                                                                | 84 |
| 6.4.18.5 Attribute interpolation .....                                                                       | 84 |
| 6.4.18.6 Attribute startPoint .....                                                                          | 84 |
| 6.4.18.7 Attribute endPoint .....                                                                            | 85 |
| 6.4.18.8 Attribute isRing .....                                                                              | 85 |
| 6.4.18.9 Attribute length .....                                                                              | 85 |
| 6.4.18.10 Attribute numDerivativesInterior .....                                                             | 85 |
| 6.4.18.11 Attribute numDerivativesStart .....                                                                | 85 |
| 6.4.18.12 Attribute numDerivativesEnd .....                                                                  | 85 |
| 6.4.18.13 Attribute startParam, endParam .....                                                               | 86 |
| 6.4.18.14 Attribute reverse .....                                                                            | 86 |
| 6.4.18.15 Attribute startConstrParam, endConstrParam .....                                                   | 86 |
| 6.4.18.16 Attribute boundary .....                                                                           | 87 |
| 6.4.18.17 Operation asLine .....                                                                             | 87 |
| 6.4.18.18 Operation constrParam .....                                                                        | 88 |
| 6.4.18.19 Operations length .....                                                                            | 88 |
| 6.4.18.20 Operation param .....                                                                              | 88 |
| 6.4.18.21 Operation paramForPoint .....                                                                      | 88 |

|                  |                                                                                                                                                                             |            |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>6.4.18.22</b> | <b>Operation tangent.....</b>                                                                                                                                               | <b>89</b>  |
| <b>6.4.19</b>    | <b>DataType CurveData .....</b>                                                                                                                                             | <b>89</b>  |
| <b>6.4.20</b>    | <b>Interface OffsetCurve .....</b>                                                                                                                                          | <b>90</b>  |
| <b>6.4.20.1</b>  | <b>Semantics.....</b>                                                                                                                                                       | <b>90</b>  |
| <b>6.4.20.2</b>  | <b>Attribute distance: Length .....</b>                                                                                                                                     | <b>90</b>  |
| <b>6.4.20.3</b>  | <b>Attribute refDirection .....</b>                                                                                                                                         | <b>90</b>  |
| <b>6.4.20.4</b>  | <b>Association role: baseCurve: Curve .....</b>                                                                                                                             | <b>91</b>  |
| <b>6.4.21</b>    | <b>Datatype OffsetCurveData.....</b>                                                                                                                                        | <b>91</b>  |
| <b>6.4.22</b>    | <b>Interface ProductCurve .....</b>                                                                                                                                         | <b>91</b>  |
| <b>6.4.22.1</b>  | <b>Semantics.....</b>                                                                                                                                                       | <b>91</b>  |
| <b>6.4.22.2</b>  | <b>Attribute parameterRange: Interval.....</b>                                                                                                                              | <b>91</b>  |
| <b>6.4.22.3</b>  | <b>Attribute projection: Projection [1..*] .....</b>                                                                                                                        | <b>92</b>  |
| <b>6.4.22.4</b>  | <b>Aggregation role: element .....</b>                                                                                                                                      | <b>92</b>  |
| <b>6.4.22.5</b>  | <b>Constructor: ProductCurve(data: ProductCurveData).....</b>                                                                                                               | <b>92</b>  |
| <b>6.4.23</b>    | <b>DataType ProductCurveData .....</b>                                                                                                                                      | <b>92</b>  |
| <b>6.4.24</b>    | <b>CodeList: CurveInterpolation.....</b>                                                                                                                                    | <b>92</b>  |
| <b>6.4.25</b>    | <b>Interface Surface.....</b>                                                                                                                                               | <b>94</b>  |
| <b>6.4.25.1</b>  | <b>Semantics.....</b>                                                                                                                                                       | <b>94</b>  |
| <b>6.4.25.2</b>  | <b>Attribute boundary: Curve [1..*].....</b>                                                                                                                                | <b>95</b>  |
| <b>6.4.25.3</b>  | <b>Attribute interpolation .....</b>                                                                                                                                        | <b>96</b>  |
| <b>6.4.25.4</b>  | <b>Attribute numDerivativesBoundary: Integer [0..1] .....</b>                                                                                                               | <b>96</b>  |
| <b>6.4.25.5</b>  | <b>Attribute numDerivativeInterior: Integer [0., 1].....</b>                                                                                                                | <b>96</b>  |
| <b>6.4.25.6</b>  | <b>Attribute perimeter: Length.....</b>                                                                                                                                     | <b>96</b>  |
| <b>6.4.25.7</b>  | <b>Attribute area: Area .....</b>                                                                                                                                           | <b>96</b>  |
| <b>6.4.25.8</b>  | <b>Attribute dataPoint: DirectPosition [0..*] .....</b>                                                                                                                     | <b>96</b>  |
| <b>6.4.25.9</b>  | <b>Attribute controlPoint: DirectPosition [0..*].....</b>                                                                                                                   | <b>96</b>  |
| <b>6.4.25.10</b> | <b>Attribute knot: Knot[0..1].....</b>                                                                                                                                      | <b>97</b>  |
| <b>6.4.25.11</b> | <b>Operation: upNormal (point: DirectPosition): Vector .....</b>                                                                                                            | <b>97</b>  |
| <b>6.4.25.12</b> | <b>Operation: Surface (constructors) .....</b>                                                                                                                              | <b>97</b>  |
| <b>6.4.26</b>    | <b>Datatype SurfaceData .....</b>                                                                                                                                           | <b>98</b>  |
| <b>6.4.27</b>    | <b>CodeList: SurfaceInterpolation .....</b>                                                                                                                                 | <b>98</b>  |
| <b>6.4.28</b>    | <b>Interface Solid.....</b>                                                                                                                                                 | <b>99</b>  |
| <b>6.4.28.1</b>  | <b>Semantics.....</b>                                                                                                                                                       | <b>99</b>  |
| <b>6.4.28.2</b>  | <b>Attribute boundary: Surface [1..*].....</b>                                                                                                                              | <b>99</b>  |
| <b>6.4.28.3</b>  | <b>Attribute area: Area .....</b>                                                                                                                                           | <b>100</b> |
| <b>6.4.28.4</b>  | <b>Attribute volume: Volume.....</b>                                                                                                                                        | <b>100</b> |
| <b>6.4.28.5</b>  | <b>Attribute dataPoint: DirectPosition [0..*] .....</b>                                                                                                                     | <b>100</b> |
| <b>6.4.28.6</b>  | <b>Attribute controlPoint: DirectPosition [0..*].....</b>                                                                                                                   | <b>101</b> |
| <b>6.4.28.7</b>  | <b>Attribute interpolation: SolidInterpolation [0..1].....</b>                                                                                                              | <b>101</b> |
| <b>6.4.28.8</b>  | <b>Attribute knot: Knot[0..3] .....</b>                                                                                                                                     | <b>101</b> |
| <b>6.4.28.9</b>  | <b>Operation: Solid (constructor) .....</b>                                                                                                                                 | <b>101</b> |
| <b>6.4.29</b>    | <b>Datatype SolidData .....</b>                                                                                                                                             | <b>101</b> |
| <b>6.4.29.1</b>  | <b>Attribute boundary: CurveData*[1..*] .....</b>                                                                                                                           | <b>101</b> |
| <b>6.4.29.2</b>  | <b>Attribute segment: SolidData*[0..*].....</b>                                                                                                                             | <b>101</b> |
| <b>6.4.30</b>    | <b>CodeList: SolidInterpolation .....</b>                                                                                                                                   | <b>101</b> |
| <b>6.4.31</b>    | <b>Interface Collection.....</b>                                                                                                                                            | <b>102</b> |
| <b>6.4.31.1</b>  | <b>Semantics.....</b>                                                                                                                                                       | <b>102</b> |
| <b>6.4.31.2</b>  | <b>Attribute elementType: GeometryType [0..*].....</b>                                                                                                                      | <b>102</b> |
| <b>6.4.31.3</b>  | <b>Attribute numElement: Integer .....</b>                                                                                                                                  | <b>103</b> |
| <b>6.4.32</b>    | <b>Role element: Geometry.....</b>                                                                                                                                          | <b>103</b> |
| <b>6.4.32.1</b>  | <b>The role element is an alternate access point to the elements in the Geometric object. The two logical arrays Attribute element and role: element are identical.....</b> | <b>103</b> |
| <b>6.4.32.2</b>  | <b>Operation: add (geo: Geometry): Boolean .....</b>                                                                                                                        | <b>103</b> |
| <b>6.4.32.3</b>  | <b>Operation: remove (geo: Geometry): Boolean .....</b>                                                                                                                     | <b>103</b> |
| <b>6.4.32.4</b>  | <b>Constructor: Collection (data: CollectionData): Collection.....</b>                                                                                                      | <b>103</b> |

|          |                                                                   |     |
|----------|-------------------------------------------------------------------|-----|
| 6.4.32.5 | Association: element: Geometry.....                               | 103 |
| 6.4.33   | DataType CollectionData .....                                     | 103 |
| 6.4.34   | Interface Complex .....                                           | 104 |
| 6.4.34.1 | Semantics .....                                                   | 104 |
| 6.4.35   | Role Complex: generator: Primitive .....                          | 106 |
| 6.4.36   | Role Complex: superComplex and subComplex .....                   | 106 |
| 6.4.36.1 | Cellular complexes (informative example for use in topology)..... | 106 |
| 6.4.36.2 | Operation: Complex: isMaximal.....                                | 107 |
| 6.4.36.3 | Contains association .....                                        | 107 |
| 6.4.36.4 | Association role element.....                                     | 107 |
| 6.5      | Requirements Class Geometry Data .....                            | 107 |
| 7        | Interpolations for Curves .....                                   | 107 |
| 7.1      | Requirements Class Line Curve.....                                | 107 |
| 7.1.1    | Semantics .....                                                   | 107 |
| 7.1.2    | Interface Line.....                                               | 108 |
| 7.1.2.1  | Semantics .....                                                   | 108 |
| 7.1.2.2  | Attribute interpolation: CurveInterpolation=linear .....          | 109 |
| 7.1.2.3  | Operation: Line (constructor).....                                | 110 |
| 7.1.3    | DataType LineData .....                                           | 110 |
| 7.2      | Requirements Class Line Data .....                                | 110 |
| 7.3      | Requirements Class Geodesic Curve.....                            | 110 |
| 7.3.1    | Semantics .....                                                   | 110 |
| 7.3.2    | Interface Geodesic.....                                           | 110 |
| 7.3.2.1  | Semantics .....                                                   | 110 |
| 7.3.2.2  | Attribute interpolation.....                                      | 111 |
| 7.3.2.3  | Operation: Geodesic (constructor) .....                           | 111 |
| 7.3.3    | DataType GeodesicData .....                                       | 111 |
| 7.4      | Requirements Class Geodesic Curve Data .....                      | 111 |
| 7.5      | Requirements Class Rhumb.....                                     | 111 |
| 7.5.1    | Interface Rhumb .....                                             | 111 |
| 7.5.1.1  | Semantics .....                                                   | 111 |
| 7.5.1.2  | Attribute interpolation.....                                      | 112 |
| 7.5.1.3  | Rhumb (constructor) .....                                         | 112 |
| 7.5.2    | DataType RhumbData.....                                           | 112 |
| 7.6      | Requirements Class Rhumb Curve Data.....                          | 112 |
| 7.7      | Requirements Class Polynomial Curves.....                         | 112 |
| 7.7.1    | Semantics .....                                                   | 112 |
| 7.7.2    | Interface RealFunction .....                                      | 113 |
| 7.7.2.1  | Semantics .....                                                   | 113 |
| 7.7.2.2  | Attribute name .....                                              | 113 |
| 7.7.2.3  | Attribute domain .....                                            | 113 |
| 7.7.2.4  | Attribute metadata .....                                          | 113 |
| 7.7.2.5  | Operation: value .....                                            | 113 |
| 7.7.3    | Interface FunctionArc.....                                        | 113 |
| 7.7.3.1  | Semantics .....                                                   | 113 |
| 7.7.3.2  | Attribute domain .....                                            | 113 |
| 7.7.4    | Association Role function.....                                    | 114 |
| 7.7.5    | Interface FunctionCurve .....                                     | 114 |
| 7.7.5.1  | Semantics .....                                                   | 114 |
| 7.7.5.2  | Attribute FunctionCurve: numArc: Integer.....                     | 114 |
| 7.7.5.3  | Attribute FunctionCurve: metadata: URL .....                      | 114 |
| 7.7.5.4  | Association Role segment: FunctionArc.....                        | 114 |
| 7.7.6    | Interface RealPolynomial.....                                     | 114 |
| 7.7.6.1  | Semantics .....                                                   | 114 |
| 7.7.6.2  | Attribute name, domain, metadata.....                             | 114 |

|                                                                                                                                                               |     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 7.7.6.3 Attribute degree.....                                                                                                                                 | 115 |
| 7.7.6.4 Attribute c: Real .....                                                                                                                               | 115 |
| 7.7.6.5 Operation: value .....                                                                                                                                | 115 |
| 7.7.7 Interface PolynomialArc.....                                                                                                                            | 115 |
| 7.7.7.1 Attribute degree: Integer.....                                                                                                                        | 115 |
| 7.7.7.2 Constructor: PolynomialArc.....                                                                                                                       | 115 |
| 7.7.7.3 Association Role function (coordinateOffset: Integer): RealPolynomial.....                                                                            | 115 |
| 7.7.8 Datatype PolynomialArcData.....                                                                                                                         | 116 |
| 7.7.8.1 Semantics .....                                                                                                                                       | 116 |
| 7.7.8.2 Attribute CoordinateOffset .....                                                                                                                      | 116 |
| 7.7.9 Interface PolynomialCurve .....                                                                                                                         | 116 |
| 7.7.9.1 Semantics .....                                                                                                                                       | 116 |
| 7.7.9.2 Attribute degree.....                                                                                                                                 | 116 |
| 7.7.10 DataType PolynomialCurveData.....                                                                                                                      | 117 |
| 7.8 Requirements Class Polynomial Curve Data .....                                                                                                            | 117 |
| 7.9 Requirements Class Conic Curves .....                                                                                                                     | 117 |
| 7.9.1 Semantics .....                                                                                                                                         | 117 |
| 7.9.2 Interface Arc.....                                                                                                                                      | 118 |
| 7.9.2.1 Semantics .....                                                                                                                                       | 118 |
| 7.9.2.2 Attribute numArc .....                                                                                                                                | 118 |
| 7.9.2.3 Attribute controlPoints .....                                                                                                                         | 118 |
| 7.9.2.4 Attribute dataPoints.....                                                                                                                             | 119 |
| 7.9.2.5 Attribute radius .....                                                                                                                                | 119 |
| 7.9.2.6 Constructor Arc (data:ArcData): ArcString.....                                                                                                        | 119 |
| 7.9.3 Datatype ArcData .....                                                                                                                                  | 119 |
| 7.9.4 Interface Circle .....                                                                                                                                  | 120 |
| 7.9.4.1 Semantics .....                                                                                                                                       | 120 |
| 7.9.4.2 Constructor Circle .....                                                                                                                              | 120 |
| 7.9.5 Interface Conic.....                                                                                                                                    | 120 |
| 7.9.5.1 Semantics .....                                                                                                                                       | 120 |
| 7.9.5.2 Attribute dataPoint and controlPoints.....                                                                                                            | 122 |
| 7.9.6 Interface EllipticArc, Datatype EllipticArcData .....                                                                                                   | 122 |
| 7.10 Requirements Class Conic Curve Data .....                                                                                                                | 123 |
| 7.11 Requirements Class Spiral Curve.....                                                                                                                     | 123 |
| 7.11.1 Semantics, Mathematical background: curves and curvature .....                                                                                         | 123 |
| 7.11.1.1 A spiral curve is a C3 curve with strictly monotonic curvature. Why this is important is traced to the formula in physics for centrifugal force..... | 123 |
| 7.11.1.2 Fundamental Theorem of Plane Curves (in $E^2$ ):.....                                                                                                | 123 |
| 7.11.1.3 Fundamental Theorem of Space Curves (in $E^3$ ): .....                                                                                               | 124 |
| 7.11.1.4 Creating a spiral from a curvature function .....                                                                                                    | 126 |
| 7.11.2 Interface Spiral Curves.....                                                                                                                           | 128 |
| 7.11.2.1 Semantics.....                                                                                                                                       | 128 |
| 7.11.2.2 Attribute curvature .....                                                                                                                            | 128 |
| 7.11.2.3 Attribute torsion: RealFunction.....                                                                                                                 | 128 |
| 7.11.2.4 Attribute startFrame.....                                                                                                                            | 128 |
| 7.11.2.5 Constructor Spiral.....                                                                                                                              | 128 |
| 7.11.3 Interface Clothoid Curve .....                                                                                                                         | 129 |
| 7.11.4 Datatype SpiralData .....                                                                                                                              | 129 |
| 7.12 Requirements Class Spiral Curve Data .....                                                                                                               | 129 |
| 7.13 Requirements Class Spline Curve .....                                                                                                                    | 129 |
| 7.13.1 Semantics .....                                                                                                                                        | 129 |
| 7.13.2 CodeList: KnotType.....                                                                                                                                | 130 |
| 7.13.3 CodeList: SplineCurveForm .....                                                                                                                        | 131 |
| 7.13.4 Interface SplineCurve .....                                                                                                                            | 131 |
| 7.13.4.1 Semantics.....                                                                                                                                       | 131 |

|          |                                                                |     |
|----------|----------------------------------------------------------------|-----|
| 7.13.4.2 | Attribute curveForm .....                                      | 131 |
| 7.13.4.3 | Attribute knot.....                                            | 131 |
| 7.13.4.4 | Attribute degree .....                                         | 132 |
| 7.13.4.5 | Attribute knotSpec.....                                        | 132 |
| 7.13.4.6 | Attribute isRational.....                                      | 133 |
| 7.13.5   | Interface PolynomialSpline .....                               | 134 |
| 7.13.5.1 | Semantics .....                                                | 134 |
| 7.13.5.2 | Attribute derivativeAtStart .....                              | 135 |
| 7.13.6   | Interface CubicSpline .....                                    | 135 |
| 7.13.7   | Interface Bezier.....                                          | 136 |
| 7.13.8   | Interface BSplineCurve (and NURBS).....                        | 137 |
| 7.13.8.1 | Semantics .....                                                | 137 |
| 7.13.8.2 | Operation: BSplineCurve.....                                   | 138 |
| 7.13.9   | DataType BsplineData .....                                     | 138 |
| 7.14     | Requirements Class Spline Curve Data.....                      | 138 |
| 8        | Interpolations for Surfaces .....                              | 138 |
| 8.1      | Requirements Class Polygon Surface .....                       | 138 |
| 8.1.1    | Semantics .....                                                | 138 |
| 8.1.2    | Interface Polygon .....                                        | 139 |
| 8.1.2.1  | Semantics .....                                                | 139 |
| 8.1.2.2  | Attribute Polygon: boundary: Curve [1..*].....                 | 139 |
| 8.1.2.3  | Association role: spanningSurface: GriddedSurface [0..1] ..... | 139 |
| 8.1.2.4  | Operation: Polygon (constructor) .....                         | 139 |
| 8.1.3    | Datatype PolygonData .....                                     | 140 |
| 8.1.4    | Interface PolyhedralSurface.....                               | 140 |
| 8.1.4.1  | Semantics .....                                                | 140 |
| 8.1.4.2  | Operation: PolyhedralSurface (constructor) .....               | 140 |
| 8.1.4.3  | Association role: segment: Polygon [1..*] .....                | 140 |
| 8.1.5    | Datatype PolyhedralSurfaceData .....                           | 140 |
| 8.1.6    | Interface Triangle.....                                        | 140 |
| 8.1.7    | Datatype TriangleData .....                                    | 141 |
| 8.1.8    | Interface TriangulatedSurface .....                            | 141 |
| 8.1.9    | Datatype TriangulatedSurfaceData .....                         | 141 |
| 8.2      | Requirements Class Polygon Surface Data.....                   | 141 |
| 8.3      | Requirements Class Parametric Curve Surface .....              | 141 |
| 8.3.1    | Semantics .....                                                | 141 |
| 8.3.2    | Interface ParametricCurveSurface .....                         | 142 |
| 8.3.2.1  | Semantics .....                                                | 142 |
| 8.3.2.2  | Attribute horizontalCurveType, verticalCurveType.....          | 143 |
| 8.3.2.3  | Attribute rows .....                                           | 144 |
| 8.3.2.4  | Attribute columns .....                                        | 144 |
| 8.3.2.5  | Attribute dataPoints.....                                      | 144 |
| 8.3.2.6  | Attribute controlPoints .....                                  | 144 |
| 8.3.2.7  | Attribute horizontalCurveType.....                             | 144 |
| 8.3.2.8  | Attribute verticalCurveType .....                              | 144 |
| 8.3.2.9  | Operation: horizontalCurve: Curve.....                         | 144 |
| 8.3.2.10 | Operation: verticalCurve: Curve.....                           | 145 |
| 8.3.2.11 | Operation: surface: DirectPosition.....                        | 145 |
| 8.3.3    | Datatype ParametricCurveSurfaceData .....                      | 145 |
| 8.3.4    | Interface BilinearGrid.....                                    | 145 |
| 8.3.5    | Extensions of ParametricCurveSurface.....                      | 146 |
| 8.4      | Requirements Class Parametric Curve Surface Data.....          | 146 |
| 8.5      | Requirements Class Conic Surface .....                         | 146 |
| 8.5.1    | Semantics .....                                                | 146 |
| 8.5.2    | Interface Sphere.....                                          | 147 |

|          |                                                                        |     |
|----------|------------------------------------------------------------------------|-----|
| 8.5.3    | Interface Cone.....                                                    | 148 |
| 8.5.4    | Interface Cylinder.....                                                | 148 |
| 8.6      | Requirements Class Conic Surface Data.....                             | 148 |
| 8.7      | Requirements Class Spline Surface.....                                 | 148 |
| 8.7.1    | Semantics .....                                                        | 148 |
| 8.7.2    | Interface BSplineSurface (and NURBS) .....                             | 149 |
| 8.7.2.1  | Semantics .....                                                        | 149 |
| 8.7.2.2  | Attribute degree: Integer.....                                         | 149 |
| 8.7.2.3  | Attribute knot: Knot [2] .....                                         | 150 |
| 8.7.2.4  | Attribute surfaceForm: BSplineSurfaceForm .....                        | 150 |
| 8.7.2.5  | Attribute knotSpec: KnotType.....                                      | 150 |
| 8.7.2.6  | Attribute isPolynomial: Boolean.....                                   | 150 |
| 8.7.2.7  | BSplineSurfaceData (default constructor datatype) .....                | 150 |
| 8.7.3    | Codelist BSplineSurfaceForm.....                                       | 150 |
| 8.8      | Requirements Class Spline Surface Data .....                           | 151 |
| 9        | Interpolations for Solids .....                                        | 151 |
| 9.1      | Requirements Class Boundary Representation Solid .....                 | 151 |
| 9.2      | Requirements Class Boundary Representation Solid Data .....            | 151 |
| 9.3      | Requirements Class Parametric Curve Solid .....                        | 151 |
| 9.3.1    | Interface ParametricCurveSolid .....                                   | 151 |
| 9.3.1.1  | Semantics .....                                                        | 151 |
| 9.3.1.2  | Attribute: horizontalCurveType, verticalCurveType, depthCurveType..... | 152 |
| 9.3.1.3  | Attribute: dataPoint, controlPoint .....                               | 153 |
| 9.3.2    | Interface BSolidSpline .....                                           | 153 |
| 9.3.3    | Other interpolations.....                                              | 153 |
| 9.4      | Requirements Class Parametric Curve Solid Data.....                    | 153 |
| 10       | Topology .....                                                         | 153 |
| 10.1     | Requirements Class Topology root.....                                  | 153 |
| 10.1.1   | Semantics .....                                                        | 153 |
| 10.1.2   | Interface Topology .....                                               | 155 |
| 10.1.2.1 | Semantics .....                                                        | 155 |
| 10.1.2.2 | Attribute isEmpty.....                                                 | 155 |
| 10.1.2.3 | Attribute dimension.....                                               | 155 |
| 10.1.2.4 | Attribute boundary .....                                               | 156 |
| 10.1.2.5 | Attribute coBoundary.....                                              | 157 |
| 10.1.2.6 | Attribute interior .....                                               | 158 |
| 10.1.2.7 | Attribute exterior.....                                                | 158 |
| 10.1.2.8 | Attribute closure .....                                                | 158 |
| 10.1.2.9 | Attribute maximalComplex .....                                         | 158 |
| 10.1.3   | Interface Primitive.....                                               | 158 |
| 10.1.3.1 | Semantics .....                                                        | 158 |
| 10.1.3.2 | Association: Realization .....                                         | 158 |
| 10.1.3.3 | Association: Complex.....                                              | 159 |
| 10.1.3.4 | Association: Isolated In.....                                          | 159 |
| 10.1.4   | Interface DirectedTopo .....                                           | 161 |
| 10.1.4.1 | Semantics .....                                                        | 161 |
| 10.1.4.2 | Attribute orientation .....                                            | 162 |
| 10.1.4.3 | Attribute negate.....                                                  | 162 |
| 10.1.4.4 | Attribute expression .....                                             | 162 |
| 10.1.4.5 | Association: Center.....                                               | 162 |
| 10.1.4.6 | Constraints .....                                                      | 162 |
| 10.1.5   | Datatype TopologyData.....                                             | 163 |
| 10.1.5.1 | Semantics .....                                                        | 163 |
| 10.1.5.2 | Attribute id: GenericName .....                                        | 163 |
| 10.1.5.3 | Attribute boundaryData .....                                           | 163 |

|          |                                                     |     |
|----------|-----------------------------------------------------|-----|
| 10.1.5.4 | Attribute geometryData .....                        | 163 |
| 10.1.6   | Data Type PrimitiveData .....                       | 163 |
| 10.1.7   | Data Type ComplexData .....                         | 163 |
| 10.1.7.1 | Semantics .....                                     | 163 |
| 10.1.7.2 | Attribute primitive .....                           | 163 |
| 10.1.8   | Datatype Expression .....                           | 164 |
| 10.1.8.1 | Semantics .....                                     | 164 |
| 10.1.8.2 | Attribute negate .....                              | 164 |
| 10.1.8.3 | Attribute boundary .....                            | 164 |
| 10.1.8.4 | Attribute coBoundary .....                          | 164 |
| 10.1.8.5 | Attribute support .....                             | 165 |
| 10.1.8.6 | Attribute asSet .....                               | 165 |
| 10.1.8.7 | Operation: plus(expression) .....                   | 165 |
| 10.1.8.8 | Operation: minus .....                              | 165 |
| 10.1.8.9 | Operation: equals .....                             | 165 |
| 10.1.9   | Datatype ExpressionTerm .....                       | 166 |
| 10.2     | Requirements Class Topology Root Data .....         | 166 |
| 10.3     | Requirements Class Node .....                       | 166 |
| 10.3.1   | Semantics .....                                     | 166 |
| 10.3.2   | Interface Node .....                                | 166 |
| 10.3.2.1 | Semantics .....                                     | 166 |
| 10.3.2.2 | Association: Center .....                           | 166 |
| 10.3.2.3 | Attribute boundary .....                            | 167 |
| 10.3.2.4 | Constraints .....                                   | 167 |
| 10.3.3   | Interface DirectedNode .....                        | 167 |
| 10.4     | Requirements Class Edge .....                       | 167 |
| 10.4.1   | Interface Edge .....                                | 167 |
| 10.4.1.1 | Semantics .....                                     | 167 |
| 10.4.1.2 | Attribute coBoundary .....                          | 167 |
| 10.4.1.3 | Attribute boundary .....                            | 168 |
| 10.4.1.4 | Association: Center .....                           | 168 |
| 10.4.1.5 | Constraints .....                                   | 168 |
| 10.4.2   | Interface DirectedEdge .....                        | 168 |
| 10.5     | Requirements Class Face .....                       | 168 |
| 10.5.1   | Semantics .....                                     | 168 |
| 10.5.2   | Interface Face .....                                | 168 |
| 10.5.2.1 | Attribute boundary .....                            | 168 |
| 10.5.2.2 | Attribute coBoundary .....                          | 169 |
| 10.5.2.3 | Association Center roles .....                      | 169 |
| 10.5.2.4 | Constraints .....                                   | 169 |
| 10.5.3   | Interface DirectedFace .....                        | 169 |
| 10.6     | Requirements Class Topology Solid .....             | 169 |
| 10.6.1   | Interface Solid .....                               | 169 |
| 10.6.1.1 | Semantics .....                                     | 169 |
| 10.6.1.2 | Attribute boundary .....                            | 169 |
| 10.6.1.3 | Attribute coBoundary .....                          | 170 |
| 10.6.1.4 | Association role: Center .....                      | 170 |
| 10.6.2   | Interface DirectedSolid .....                       | 170 |
| 10.7     | Requirements Class Topological Complex .....        | 170 |
| 10.7.1   | Semantics .....                                     | 170 |
| 10.7.2   | Interface Complex .....                             | 171 |
| 10.7.2.1 | Semantics .....                                     | 171 |
| 10.7.2.2 | Complex: constructor of a topological complex ..... | 171 |
| 10.7.2.3 | Association role: maximalComplex .....              | 172 |
| 10.7.2.4 | Attribute isMaximal .....                           | 172 |

|                     |                                                                       |     |
|---------------------|-----------------------------------------------------------------------|-----|
| 10.7.2.5            | Attribute isConnected.....                                            | 172 |
| 10.7.2.6            | Association: Contains.....                                            | 173 |
| 10.7.2.7            | Association: Complex.....                                             | 173 |
| 10.7.2.8            | Association: Realization .....                                        | 173 |
| 10.8                | Requirements Class Derived Topological Relations.....                 | 173 |
| 10.8.1              | Introduction .....                                                    | 173 |
| 10.8.2              | Canonical form for Geometry.....                                      | 174 |
| 10.8.3              | Boundary operators for aggregate objects .....                        | 175 |
| 10.8.3.1            | Common semantics .....                                                | 175 |
| 10.8.3.2            | "Mod 2" boundaries.....                                               | 175 |
| 10.8.3.3            | "At least 2" boundaries .....                                         | 176 |
| 10.8.4              | Boolean or set operators .....                                        | 176 |
| 10.8.4.1            | Form of the Boolean operators (originally Egenhofer 4 matrix).....    | 176 |
| 10.8.4.2            | Relate Reference Operation .....                                      | 177 |
| 10.8.4.3            | Relation to set operations.....                                       | 177 |
| 10.8.5              | Egenhofer operators .....                                             | 177 |
| 10.8.5.1            | Form of the Egenhofer operators (originally Egenhofer 9 matrix) ..... | 177 |
| 10.8.5.2            | Egenhofer relate .....                                                | 178 |
| 10.8.5.3            | Relation to set operations.....                                       | 178 |
| 10.8.6              | Full topological operators .....                                      | 178 |
| 10.8.6.1            | Form of the full topological operators.....                           | 178 |
| 10.8.6.2            | Full topological relate.....                                          | 179 |
| 10.8.6.3            | Named Operators.....                                                  | 180 |
| 10.8.6.3.1          | Equals .....                                                          | 180 |
| 10.8.6.3.2          | Contains.....                                                         | 180 |
| 10.8.6.3.3          | Disjoint.....                                                         | 180 |
| 10.8.6.3.4          | Intersects.....                                                       | 180 |
| 10.8.6.3.5          | Touches.....                                                          | 180 |
| 10.8.6.3.6          | Crosses .....                                                         | 181 |
| 10.8.6.3.7          | Within.....                                                           | 181 |
| 10.8.6.3.8          | Overlaps.....                                                         | 182 |
| 10.8.7              | Combinations.....                                                     | 182 |
| 11                  | Special Requirements Classes.....                                     | 182 |
| 11.1                | Requirements Class Simplicial geometry .....                          | 182 |
| 11.1.1              | Semantics .....                                                       | 182 |
| 11.1.2              | Datatype Simplex.....                                                 | 183 |
| 11.1.2.1            | Semantics .....                                                       | 183 |
| 11.1.2.2            | Attribute dimension.....                                              | 184 |
| 11.1.2.3            | Attribute numVertex.....                                              | 184 |
| 11.1.2.4            | Attribute vertex .....                                                | 184 |
| 11.1.2.5            | Operation BoundarySimplex .....                                       | 184 |
| 11.1.2.6            | Operation Boundary .....                                              | 185 |
| 11.1.3              | Data Type SimplicialTerm.....                                         | 185 |
| 11.1.4              | Data Type::SimplicialPolynomial .....                                 | 185 |
| 11.1.5              | Data Type::SimplicialComplex.....                                     | 185 |
| 11.2                | Requirements Class Point Clouds .....                                 | 185 |
| 11.2.1              | Semantics .....                                                       | 185 |
| 11.2.2              | Interface PointCloud .....                                            | 186 |
| 11.2.2.1            | Semantics .....                                                       | 186 |
| 11.2.2.2            | Attribute range: Range.....                                           | 186 |
| 11.2.2.3            | Attribute point .....                                                 | 186 |
| 11.2.2.4            | Operation: SubCloud .....                                             | 186 |
| 11.2.2.5            | Operation: SimplicialComplex.....                                     | 187 |
| Annex A (normative) | Abstract test suite .....                                             | 188 |

|                                                                            |            |
|----------------------------------------------------------------------------|------------|
| <b>Annex B (informative) Abstract test suite .....</b>                     | <b>202</b> |
| <b>Annex C (informative) MiniTopo .....</b>                                | <b>207</b> |
| <b>Annex D (informative) Crosswalk 19107:2003 to Current version .....</b> | <b>213</b> |
| <b>Bibliography .....</b>                                                  | <b>218</b> |

## Foreword

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

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

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

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

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

The committee responsible for this document is Technical Committee ISO/TC 211, *Geographic information/Geomatics*".

This second edition cancels and replaces the first edition (ISO 19107:2003), which has been technically revised and now forms a logical subset of this second edition. In other words, this International Standard is 100 % backwardly compatible with its previous version, ISO 19107:2003, except in a few areas (in NURBS) where the previous version contained technical errors that are corrected in this revision.

## Introduction

This International Standard provides conceptual schemas for describing, representing and manipulating the spatial characteristics of geographic entities. Standardization in this area is the cornerstone for other geographic information design, specification and standardization.

"Vector" data consists of geometric primitives used to construct expressions of 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. This International Standard deals only with vector data.

There is a hierarchy of complexity in the "geometry" of the underlying object used in various coordinate systems. These may use reference planes (map geometry – Euclidean), reference spheres (spherical geometry — using spherical trigonometry), reference ellipsoids (ellipsoidal geometry using Gaussian or Riemannian metrics) or more complex surfaces (usually using numeric approximations for calculation). The coordinates of a point locate it on, or in relation to, the reference geometry. With the exception of the "map geometry," the usual Euclidean formulae for distance and area do not apply directly in the coordinate system.

Topology expressions provide qualitative descriptions of the spatial relations between geometry objects. Topology deals with the characteristics of geometric figures that remain invariant if the space is deformed elastically. Topological properties do not change when information is transformed from one coordinate system to another, usually including the coordinate function that map from  $\mathbb{R}^2$  or  $\mathbb{R}^3$  to the reference geometry. Topological properties in the domain of the coordinate system will be identical to those on the geographic surface; but the metric properties may change significantly (e.g. distance, area, direction).

Spatial operators are functions and procedures that use, query, create, modify or delete spatial objects. This International Standard defines the taxonomy of some of the more important operators, their definitions and implementations. The goals are to:

- Define spatial operators unambiguously, so that different implementations will yield comparable results within the limitations of accuracy and resolution.
- 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.
- Define an operator algebra that will allow combinations of the base operators to be used predictably in the query and manipulation of geographic feature data.

Standardized conceptual schemas for spatial characteristics will increase the ability to share geographic information between 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 and functions.

This standard is technical because geometry is a technical topic. Euclid was speaking of a simpler form of geometry to the most powerful man in his world when he said:

*There is no royal road to geometry (μή εἶναι βασιλικὴν ἀτραπὸν ἐπὶ γεωμετρίαν).*

*Euclid to Ptolemy I Soter (General with Alexander the Great, Pharaoh of Egypt) —  
Attributed by Proclus (412–485 AD) in Commentary on the First Book of Euclid's Elements*

# Geographic information — Spatial schema

## 1 Scope

This International Standard specifies conceptual schemas for describing the spatial characteristics of geographic entities, and a set of spatial operations consistent with these schemas. It treats "vector" geometry and topology. It defines standard spatial operations for use in access, query, management, processing and data exchange of geographic information for spatial (geometric and topological) objects. Because of the nature of geographic information, these geometric coordinate spaces will normally have up to three spatial dimensions, one temporal dimension and any number of other spatially dependent parameters as needed by the applications. In general, the topological dimension of the spatial projections of the geometric objects will be at most three.

## 2 Conformance

### 2.1 Requirements class conformance targets

This International Standard presents conceptual schemas for describing the spatial characteristics of geographic features. This schema defines conceptual interfaces that may be realized 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.

Standardization targets (things which are testable against the Conformance Classes) determine the depth to which test are made. This International Standard determines tests for three target classifications of target types. These are:

- Data structures for data interchange, messaging or transfers using encoding similar to those defined in ISO 19118 [2]. These formats may be text or binary based. Text formats include any use or well-known text formats, or textual encodings such as XML, or JSON. Binary transfer can be associated to the in-core storage formats of any sufficiently robust programming language;
- Interfaces to software libraries and services; or
- Interfaces to databases using query languages such as SQL [8] or GeoSPARQL [36]

Each interface will have a default constructor with an associated data type that describes one common creation method based on a minimal and logically complete representation of that interface's datatype implementation. Interchange and encoding will be based on these default constructors and data types.

Compliance to a conformance class by a "data type" structure is a logical conformance, not a specific structure. As long as the data types implemented correspond in its informational content to the default constructors, they are deemed compliant. The following requirement defines the test more concisely.

**REQ. 1**      **Conformant data structures shall be logically equivalent to the default constructor for those interfaces supported by a requirements class.**

**Rec. 1**      **Equivalence should be demonstrated by a one to one bidirectional association between the values of the conformant datatype values and the possible values of the classifiers in the corresponding requirements class in this International Standard.**

Two structures are logically equivalent ( $\cong$ ) if each can be determined ( $\Rightarrow$ ) from the other.

$$[A \cong B] \Leftrightarrow [[A \Rightarrow B] \wedge [B \Rightarrow A]]$$

(1)

The mechanism defined in the above recommendation essentially makes all implementations compliant to ISO 19107:2003 also compliant as a data application to this International Standard. For example, all arcs constructors in ISO 19107:2003 can be used to calculate the parameters of the default arc data constructor (see Clause 7.9.3) which is the arc centers and a set of arc endpoints.

Few applications will require the full range of capabilities described by this conceptual schema. This International Standard, 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. The conformance classes are arranged as a core set of functionality with an orderly progression of extensions, so that a partial implementation of this International Standard has a logical path to the functionality of the extensions.

Implementations that define full object functionality will allow the implementation of all operations and support logical access to the conceptual attributes and associations defined by the classifiers of the chosen conformance classes.

These implementations will normally be programming language dependent, but there are no specific language restrictions. UML is generally used for object-oriented languages with class systems, but as long as instances can be created that support the functions required, the use of a class-system is an implementation decision. For example, C++ or JAVA may directly implement the UML model in this International Standard, but JavaScript (which does not have a class-based system) is still capable of implementing object instances that can support the class-contracts in the conformance classes in this document.

Implementations that choose to depend on external "free functions" for some or all operations, or forgo them altogether, will support data types sufficient to record the state of each of the chosen classifier as defined by its member variables and operations.

The support of sufficient data implies that all functions defined here can be implemented based solely on these data descriptions. Common encodings may include XML, Well-Known Text, marshalling languages such as JSON (JavaScript Object Notation) and any other format/language needed by an application environment.

Common names for "metaphorically identical" but technically different entities are necessary to link knowledge common to different domains or applications. The UML model in this International Standard defines abstract interfaces and data structures, application schemas define conceptual classes, various software systems define implementation classes or data structures, and the encoding standard (e.g. ISO 19118) defines entity tags. All of these may 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" named types defined in the UML model to be used directly under the same or similar names in application schemas. If distinctions need to be made, package name or namespace should be supplied. For example, this International Standard defines a Geometry interface logically equivalent to the Geometry defined in ISO 19107:2003, and gml: AbstractGeometry defined in ISO 19136:2007 Geography Mark-up Language (GML).

Because ISO 19107:2003 is a logical subset of this International Standard, all implementations conformant to ISO 19107:2003 that have corrected the pertinent errors in that document are conformant to some conformance class of this International Standard. Each Requirements Class described in this International Standard will have two associated conformance classes:

- a) based on data, operations and behaviour; or

b) based on data alone.

The purpose of “a-type” conformance is to identify the minimal functionality for the active interoperation of geometric data applications. The purpose of “b-type” conformance is to enable universal communication of spatial information between applications consistent with this International Standard and its predecessor ISO 19107:2003 and thereby defining a consistent and universal information community for geometry in spatial applications. For example, all versions of GML would be in “b-type” conformance.

**REQ. 2** An “a-type” or “behavioral” conformant implementation of any requirements class in this international standard shall be able to satisfy all requirements in that Requirements Class.

**REQ. 3** A “b-type” or “data” conformant implementation of any requirements class in this International standard shall be able to communicate spatial information (coordinate, geometry or topology as appropriate) based on the default constructors defined in that requirements class or logically informational-equivalent data structures whose transformation to the default constructor data types by a well-known and well-defined logical mechanism.

There are many conformant options for schemas that define types for geometric and topological objects; they are differentiated based on differences in the following criteria:

- Geometry metrics (geodesy);
- Topological dimensionality;
- Interpolation schemes;
- Structural complexity;
- Functional completeness.

### 2.1.1 Geometry metrics (geodesy)

In defining the geometry metrics, the application schema will be required to specify which underlying surfaces will be supported in calculating geometric measures. In general, the underlying surfaces (called “GeometricReferenceSurface”) are:

- The 2D plane or map geometry uses standard Cartesian geometry and Pythagorean “flat” metrics, and has inherent inaccuracies associated with the curvature of the Earth dependent on the size of the area covered and the characteristics of the chosen projections (coordinate reference systems or CRS);
- A 2D sphere uses classical spherical surfaces (a perfectly round Earth of constant curvature). These systems use great circles as geodesics and do not adjust for the eccentricity of the Earth's surface. Depending on the CRS, slightly different spheres may be chosen for a “best fit” for a limited locale. While theoretically more accurate than planar, as the area of coverage increases accuracy of measurements do degrade but at a slower rate;
- A 2D ellipsoid uses a surface generated by the rotation of an ellipse about its shorter axis (polar) creating a circular equator and other lines of constant latitude, with orthogonal, elliptical lines of constant longitude. These systems adjust for the eccentricity of the ellipsoid, but do not factor in local gravitational anomalies. Again, varieties of “best fit” ellipsoids are used;

- A continuous 2D surface of an equipotential surface of gravity (such as the geoid or quasi-geoid) or a tidal Reference Surface (such as mean sea level or lowest astronomic tide); Potential (or height) differences are referenced to an ellipsoid.

NOTE Although generally smooth and spatially continuous in two dimensions, such a surface is irregular and there is insufficient information to compute across it. It is usually represented through an analytical approximation using spherical harmonics where the spatial context is referenced to an ellipsoid, such as the Earth Gravity Model 2008 (EGM 2008) referenced to the WGS 84 ellipsoid, or by a digital elevation model referenced to a 2D plane.

Three-dimensional geometry models will also be supported. There are essentially two approaches:

- Geocentric Cartesian 3D, using a 'flat' 3D Euclidean space: These include geocentric (origin at the "center" of the Earth) and projective (origin on the surface of the Earth);
- Using one of the surfaces from the four classes above, and adding a measure of the distance in 3D either above or below the "GeometricReferenceSurface": The third dimension is measured parallel to a near-normal or normal line (orthogonal to the surface, in the general direction of gravitational up).

### 2.1.2 Topological dimensionality

In defining the topological dimensionality of object types to be implemented, the application schema will be required to specify which of the interpolation types for curves or surfaces they wish to implement.

There are four levels for simple spatial geometry:

- 0-dimensional objects (points and locations in a well-defined coordinate system);
- 0-dimensional and 1-dimensional objects (adding curves of various types);
- 0-, 1-, and 2-dimensional objects (adding surfaces or areas of various types);
- 0-, 1-, 2- and 3-dimensional objects (adding solids).

### 2.1.3 Interpolation schemes

Curve implementations, for those application schemas including topological 1-dimensional objects, will always include "linear" or "geodesic" interpolation. Application schema including 1-dimensional objects should always include a mechanism to approximate any curve as a line or geodesic string to allow for transfer of data into simpler schemas when needed.

Surface implementations, for those application schemas including 2-dimensional objects, will always include a "planar" or "reference surface" interpolation technique. The most common such interpolations are boundary representations, where boundary curves lie on a single surface. These include triangular irregular networks (TIN) and polyhedral surfaces. Obviously, in 2D coordinate reference systems these polygons (with any curves as boundary) are the only surface types. "Non-flat" surfaces require 3D coordinate reference systems.

**Rec. 2** Application schema should always include a mechanism to approximate any surface as collections of planar surface segments to allow for transfer of data into simpler schemas when needed.

Additional curve and surface interpolation mechanisms are optional, but if implemented, they will follow the definition included in this International Standard.

There are no restrictions on extensions of this International Standard. The most common extension themes may include, but are not limited to the following:

- Mechanisms to improve geometric measures such as Gaussian and Riemannian metric techniques and algorithms for the improvement of accuracy or efficiency in measures for length, area and volume for geometric objects;
- Interpolation techniques that extend the point, curve, surface and solid techniques established in this international standard, e.g. triangulated networks, tetrahedral networks, specific spline interpolation methods. These may be targeted to specific applications;
- Indexing or tiling algorithms that improve performance or efficiencies for geometries in important geographic coordinate reference systems or datums;
- Encoding techniques for geographic geometry, such as WKT, JSON or XML;
- Algorithmic improvements to the maintenance and editing of geometry and topology structures both cellular and simplicial topology;
- Dealing with spatio-temporal (4D) geometries and topologies.

#### 2.1.4 Structural complexity

Four levels of structural complexity are identified:

- Geometric primitives;
- Geometric complexes;
- Topological primitives and complexes;
- Topological primitives and complexes with geometric realization.

**NOTE** 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 most obvious implementation would have every geometry object also implement the topological object interface. The types of object included in a complex are controlled by the dimension of that complex. What is commonly called "chain-node" or "link-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 spatial coordinate system. A full topology in a 3D environment would be a 3-dimension topological complex realized by geometric objects in a 3D spatial coordinate system.

The third criterion is the level of functional complexity or completeness that determines the member elements (attributes, association roles and operations) of those types that are implemented.

#### 2.1.5 Functional complexity

There are three levels of functional completeness as follows:

- Data types only;
- Simple data access operations;
- Complete operations.

Clause 10.8, Requirements Class Derived Topological Relations, of this International Standard defines Boolean operators that may be used to investigate topological and metric relations between geometric objects.

The normative content of this International Standard is divided into the "requirements class" packages listed in the UML model and normative Clauses. Each of these sets of requirements will correspond to a conformance class in the Abstract Test suite. Each requirements class Dependency: will be realized as a conformance class extension.

## 2.2 Conformance classes

To conform to this International Standard, an implementation will satisfy the requirements of the Abstract test suite (ATS) in Annex A for a specified conformance class. In general, these conformance classes can be given by "description" parameters. These parameters include the following identifiers and their list of possible values:

Geometric Reference Surface (coordinate surface representing 0-height):

- Planes;
- spheres and planes;
- ellipsoids (biaxial rotated on the polar axis), spheres and planes; or
- geosurfaces and geoids, ellipsoids, spheres and planes.

Mechanism for height:

- none (2D) ;
- distance from the GeometricReferenceSurface (3D or compound 2+1D).

Dimension of non-empty geometry supported (empty is "-1" dimensional):

- 0-dimension, only points or direct positions;
- 1-dimension, curves and points;
- 2-dimension, surfaces, curves and points; or
- 3-dimension, solids, surfaces, curves and points.

Support of topological structures up to dimension<sup>1)</sup>

- no support;
- 1-dimensional network graphs, e.g. links and nodes;
- 2-dimensional graphs, e.g. networks spanned by surfaces; or

---

<sup>1)</sup> Higher dimension topological complexes are acceptable, but probably restricted to cellular or simplicial topological complexes, where each element is an isomorphic copy of a disk of appropriate dimensions ( $D_n = \{p \in \mathbb{R}^n \mid \|p\| \leq 1\}$ ). This will allow academic theorem from algebraic topology (specifically CW or simplicial complexes) to be applied appropriately.

- 3-dimensional, e.g. as above with surfaces spanned by solids;
- Higher dimension based on application domains, see ISO 19111-2, usually parametric extension of a base spatial representation of at most 3D.

Support for Interpolations:

- Linear including small circle along a parallel of latitude or great circle along a meridian;
- planar;
- bilinear and trilinear;
- polynomials, such as bicubic;
- splines which use the vector algebra in the domain ( $\mathbb{R}^n$ ) of the coordinate reference system used in the direct positions;
- rhumb line curves of constant navigational bearing, crossing all meridians at the same angle;
- and others.

Euclidean geometry in local spaces (using the tangent plane) which is then mapped to the surface supported using the mapping defined by geodesics (see exponential map in the definitions below (includes geodesics, geodesic circles and other conic sections including spirals).

This means that there will be one core conformance class containing general abstract classes but implementations for point only. Each extension adds new implementations classes, clustered into packages as needed to create a logical progression of valuable extensions.

Each implementation will have a set of "codelists" that specify support for geometry classes, coordinate systems and underlying geometric surfaces (see Clause 6.2.1).

### 2.3 Requirements classes

Each Conformance Class corresponds to a single Requirements Class listed by clause number in the List of Requirements. Each such pair or conformance plus requirements corresponds to a package with stereotype «requirementsclass» in the UML model. Dependencies between these packages (see Figure 1) correspond to dependencies between the classes. For example, the core requirements class is "Coordinates" in the UML. All other requirements classes are dependent on this core. The second layer is rooted at "Geometry" containing primitive such as Point, Curve, Surface, Solid or Topology. Directly below are the extensions for each of the variations of the primitive types (curve, surface, solids and topology). "Point" does not need a separate requirements class since there is only one Point interface that is in the core "Geometry" package. For Curves, Surfaces and Solids there are more conformance extensions based on interpolation type.

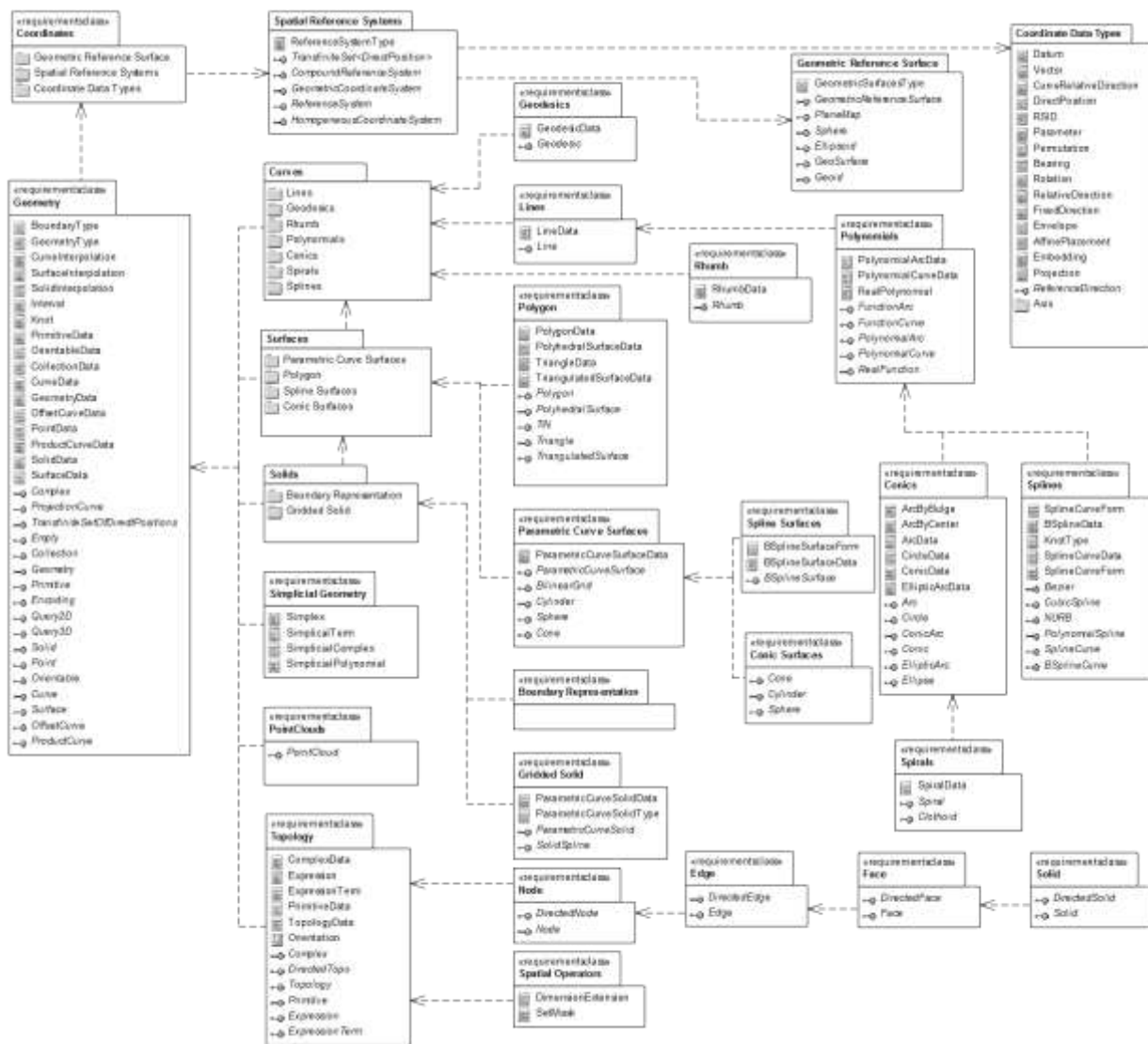


Figure 1 — Geometry Packages and Requirements Classes

### 3 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 19101-1, *Geographic information — Reference model*

ISO 19103, *Geographic information — Conceptual schema language*

ISO 19108, *Geographic information — Temporal Schema*

ISO 19109, *Geographic information — Rules for application schema*

ISO 19111, *Geographic information — Spatial referencing by coordinates*

ISO 19111-2, *Geographic information — Spatial referencing by coordinates — Part 2: Extension for parametric values*

ISO 19125-1, *Geographic information — Simple feature access — Part 1: Common architecture*

ISO 19125-2 *Geographic information — Simple feature access — Part 2: SQL option*

ISO 19162:2015, *Geographic information — Well-known text representation of coordinate reference systems*

ISO/IEC 11404:2007, *Information technology — Programming languages, their environments and system software interfaces — Language-independent datatypes*

ISO/IEC 13249-3:2011, *Information technology — SQL Multimedia and Application Packages — Part 3: Spatial*

ISO 80000-2:2009, *Quantities and units — Part 2: Mathematical signs and symbols to be used in the natural sciences and technology*

OMG Unified Modeling Language™ (OMG-UML), Infrastructure, Version 2.4.1,  
<http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/>

OMG Unified Modeling Language™ (OMG-UML), Superstructure, Version 2.4.1,  
<http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF>

## 4 Terms and definitions

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

All terms in any normative reference also apply when used in the context of that reference. Definitions of common data types are taken from ISO 19103, ISO/IEC 11404 or OMG-UML. Any term in the normative references in Clause 3 may be used.

Common words from geometry, such as point, curve, line, surface solid, etc., take the common meanings unless they are used as classifier names (usually interfaces), in which case they are a digital representation of the geometric concept. Common mathematical terms that are not defined here take on their common meanings in mathematics (see [11], ISO 80000-2, ISO/IEC 11404 or a standard text on the topic, such as the "N. Bourbaki"<sup>2)</sup> series currently published by Springer Verlag, in French, English and German). Care should be taken since mathematical terms can be context sensitive, and may easily be confused with common words. For example, "open" set, "closed" curve, "closed" set, "rational" function, "boundary," "interior," "closure", "exterior", "function" and others from common language but have very specific meanings in mathematics and in this International Standard. Where necessary to prevent confusion, existing definitions have been elaborated to make their intent in this International Standard explicit. Mathematical terms include common vocabulary from geometry, topology, calculus, geodesy and differential geometry. Many of these terms may be sufficiently common that inclusion is not necessary. They are included here to prevent confusion especially for terms like the ones listed above that have another meaning in another context.

---

<sup>2)</sup> N. Bourbaki is the pen-name for the "Association des collaborateurs de Nicolas Bourbaki" (Association of Collaborators of Nicolas Bourbaki) of mathematicians first published in 1935 and dedicated to "formalizing" mathematics. The group has an office at the "École Normale Supérieure" in Paris. See [https://en.wikipedia.org/wiki/Nicolas\\_Bourbaki](https://en.wikipedia.org/wiki/Nicolas_Bourbaki). Their books are held in high regard by the mathematical community.

#### 4.1

##### abstract root

<programming> common root classifier of a category which is a superclass of any other classifier in the category

Note 1 to entry: The class Any in some programming languages is the abstract root of all classes. Thus, it is the *de facto* union of all classes. In this International Standard, Geometry is the (named and explicit) abstract root for all geometry objects. In the package Geometry and any of its subpackages (including those in its Requirements Classes), any interface will be a subtype of Geometry either directly or transitively.

#### 4.2

##### arc

<geometry> segment (4.80) of a curve

#### 4.3

##### barycentric coordinates

<coordinate geometry> point in a **n-dimension** coordinate system using  $n+1$  numbers,  $[u_0, u_1, u_2, u_3, \dots, u_n]$ ,  $0 \leq u_i \leq 1$ ,  $\sum u_i = 1.0$ , in which the location of a point of an **Error! Reference source not found.** (of any dimension) is specified by a weighted center of mass of equal masses placed at its vertices using vector algebra of the  $\mathbb{R}^n$  used in the **Error! Reference source not found.**

Note 1 to entry: Even though there are  $n+1$  coordinates in a barycentric coordinate system, the topological dimension is  $n$ , since the restriction (sums to 1.0) loses 1 degree of freedom (once you have  $n$  ordinates, the remaining one is determined such as  $u_n = 1.0 - \sum_{i=0}^{n-1} u_i$ ). The coordinates for the simplex are all non-negative, but the system can be extended outside of the simplex by using negative numbers. If the ordinates are all positive, then the point is inside (interior to) the  $n$ -simplex. If one of them is 1.0 and the other 0, this is a corner of the simplex. If one of them is zero and the others still each greater than or equal to zero, the point is on the  $n-1$ -simplex opposite the vertex zeroed out. If any are negative, the point is outside of the simplex. The coordinates are dependent on the underlying coordinate reference system of the source data.

#### 4.4

##### bearing

horizontal angle, tangent or direction at a point

Note 1 to entry: This definition (as opposed to the one in ISO 19162:2015) is required for this International Standard because the concept is used in other definitions, such as first geodetic problem and second geodetic problem. The two definitions are nearly equivalent because the tangent of a curve on a surface is a tangent to the surface, and does specify a direction. Usual 2D measure of bearing can be an angle equivalently measured from North clockwise, or a unit tangent vector. If the coordinate system is spatially 3D, the horizontal bearing angle may also need to a vertical altitude angle to be complete. If a reference curve (as used in ISO 19162) is parameterized by arc length, then the "derivative" is a unit vector. If another parameterization " $t$ " is used, then the derivative should be normalized ( $\vec{\tau}/\|\vec{\tau}\|$ ;  $\dot{c}(t) = \vec{\tau}$ ). This is useful, since parameterization by arc length can be computationally difficult. The numeric representation of a vector depends on the coordinate system. The bearing is not dependent on a coordinate system but it can be represented in any reasonable system. The bearing is not dependent on its various representations.

#### 4.5

##### bicontinuous

<mathematics> invertible, continuous and with a continuous inverse

#### 4.6

##### boundary

set that represents the limit of an entity

Note 1 to entry: 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.

#### 4.7

##### **buffer**

geometric object containing all points and only those points whose distance from a specified geometric object is less than or equal to a given distance use in its construction

#### 4.8

##### **closure**

union of the interior and boundary of a topological object 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 1 to entry: If a node is on the boundary of an edge, that edge is on the coboundary of that node. Any orientation parameter associated with one of these relations would also be associated with the other. The coboundary of a node can be called a "node star".

#### 4.10

##### **conformal**

angle-preserving

Note 1 to entry: Some projections are conformal. For example, a Mercator preserves the angle between curves, so that if two curves in a Mercator projected plane cross at a 90°, then the preimage curves on the ellipsoid also cross at 90°, such as lines of constant latitude and lines of constant longitude.

#### 4.11

##### **connected**

property of a topological space implying that only the entire space or the empty set are the only subsets which are both open and closed

Note 1 to entry: The formal definition of connected is that any pair of locally open sets whose union is the entire space must have a non-empty intersection.

a topological space  $T$  is connected if and only if

$$[X, Y \text{ are open}] \Rightarrow [X, Y \subset T, X \cup Y = T \Rightarrow X \cap Y \neq \emptyset] \quad (2)$$

This formal definition is difficult to use. The term path connected, defined below is equivalent for the purposes of this International Standard. The use of "finite precision" coordinates makes sets which are connected but not path connected impossible to represent. In all cases "connected" is used, but "path connected" is easier to test and to visualize.

#### 4.12

##### **connected node**

<topology> node that starts or ends one or more edges

#### 4.13

##### **control point**

<coordinate geometry> point used in the construction of a geometry that partially controls its shape but does not necessarily lie on the geometry

Note 1 to entry: A center of an arc is a control point; poles in b-spline curves are control points.

#### 4.14

##### **convex**

<geometry> containing all points on a "line" joining two interior points

Note 1 to entry: The definition of convex requires a definition of line. For coordinate systems, this is the usual linear interpolated arc, but in context, the "line" on a geometric reference surface will be a "geodesic arc". The default in this document is the linear interpolate.

#### 4.15

##### **convex hull**

<geometry> smallest convex set containing a given geometric object

Note 1 to entry: "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". Another definition in a Euclidean space  $\mathbb{E}^n$  is the union of all lines with both end points in the given geometric object.

$$\begin{aligned} C &= A.\text{convexHull} \Leftrightarrow \\ &[C.\text{convex} = \text{TRUE}] \wedge [A \subset C] \wedge \\ &[[B.\text{convex} = \text{true}, A \subset B] \Rightarrow [A \subseteq C \subseteq B]] \end{aligned} \quad (3)$$

#### 4.16

##### **coordinate**

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

Note 1 to entry: In a coordinate reference system, the numbers are qualified by units. The number of offsets (generally called "ordinates") in a coordinate is not the dimension. If there is a constraint, the dimension can be smaller. See "coordinate dimension".

#### 4.17

##### **coordinate dimension**

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

Note 1 to entry: The coordinate dimension represents the number of choices made, and constraints can restrict choices. A barycentric coordinate which has (n+1)-offsets, but the underlying space is dimension n. Homogeneous coordinates (wx, wy, wz, w) are actually 3 dimensional because the choice of "w" does not affect the position, i.e. (wx, wy, wz, w) = (x, y, z, 1) → (x, y, z) which is not affected by w. The dimension will be at most the count of the numbers in the coordinate, but it can be less if the coordinates are constrained in some manner.

#### 4.18

##### **coordinate reference system**

<differential geometry, geodesy> coordinate system that is related to an object by a datum

[SOURCE: ISO 19111:2007, 4.8, modified – Note 1 and 2 to entry has been added]

Note 1 to entry: The original definition of coordinate reference system (CRS) uses a geometric object (a geodetic datum) which is referenced to the real world by a Datum. ISO 19111-2 extends this to any "parameter" which essentially can be represented as the graph of the parameter relation. This graph is in the cross product of the Datum, the domain space of the parameter function, and the parameter space (which in turn may be multi-dimensional). Thus, every coordinate system used in this International Standards is logically a CRS. Since this was not the original intent of ISO 19111-2, this International Standards will use CRS for coordinate systems associated to a geodetic datum, and use the more general term "coordinate system" for anything else that is either more or less. Here, CRS means spatial coordinates. In every case where the underlying datum (surface) is not flat, the coordinate system is not Euclidean, and the metric is not Pythagorean.

Note 2 to entry: In the event that later versions of ISO 19111 change the definition as copied above, this standard should not be affected. In the case that dynamic datums are used, the measurements made by the geometry operations of the objects in this standard would only be valid at the time used for the datum and the associated coordinates. If the datum varies, the measure may also vary, but the definitions of the measures remains the same, but the actual values may be a function of time.