Basismodul 3D

für

INTERLIS 2.4

Inhaltsverzeichnis

1. Einleitung

Die hier vorliegende Dokumentation beschreibt das Basismodul 3D des Bundes. Das Basismodul 3D ergänzt die bestehenden Basismodule des Bundes [1] um 3D-Geometrietypen.

Die Dokumentation ist wie folgt aufgebaut:

- In Kapitel 1 werden die Entwurfsziele und möglichen Anwendungen aufgezeigt.
- In Kapitel 2 werden die 3D-Geometrietypen des Basismoduls erläutert.
- Der Anhang enthält die Beschreibung des Datenmodells in INTERLIS 2.4 [2].

1.1 Entwurfsziele

Das Gebiet der 3D-Geometrietypen ist sehr umfangreich und es existieren dazu bereits verschiedene Standards (z.B. ISO 19107 Spatial Schema, ISO 16739 IFC, GML, etc.). Es ist nicht Absicht des 3D-Basismoduls, diese Standards zu ersetzen. Es geht vielmehr darum, dem INTERLIS-Modellierer einen einfachen aber praxisgerechten Werkzeugkasten zur Verfügung zu stellen, um den Einstieg in die 3D-Datenmodellierung zu erleichtern und den Wildwuchs von 3D-Datentypen in INTERLIS-Modellen zu verhindern. Auf die Erweiterungen der aktuellen Sprache INTERLIS 2.4 um zusätzliche 3D Typen wurde bewusst verzichtet.

Dazu konzentriert sich das 3D-Basismodul auf die Definition von allgemein verwendbaren Basistypen (Punk, Linie, Fläche, Körper). Die Basistypen wurden so angelegt, dass sie in Zukunft mit Mitteln der Sprache INTERLIS leicht erweitert werden können. Aktuell werden z.B. nur ebene 3D-Flächen angeboten, in Zukunft wären aber z.B. auch 3D-Freiformflächen (Splines) denkbar.

Um die Komplexität des 3D-Basismoduls nicht unnötig zu erhöhen, wurde generell auf die Einführung von «Constructive Solid Geometry» (CSG) verzichtet. Im aktuellen 3D-Basismodul gibt es daher keine Basiskörper (Kugel, Kegel, Zylinder, Quader, etc.) welche sich via Addition oder Subtraktion zu neuen Körperformen kombinieren lassen.

1.2 Mögliche Anwendungen

Bei der Auswahl der Basistypen wurde darauf geachtet, dass damit konkrete Anwendungsgebiete abgedeckt werden können. Beispiele hierfür sind:

- Stockwerkeigentum: Der Typ `Solid3D` kann direkt für die Modellierung von Stockwerkeigentum verwendet werden.
- Leitungskataster: Mit dem Typ `Pipe3D` steht ein einfacher Typ für die Definition von Leitungen zur Verfügung. Der Typ `Solid3D` kann in einer 3D-Symbolbibliothek z.B. für die Definition von Schächten verwendet werden.
- Geologie: Der Typ `Tin3D` kann für Trennflächen zwischen geologischen Schichten und der Typ `Solid3D` für die Modellierung der Schichten verwendet werden.
- Verkehr: Strassenachsen können mit dem Typ `CompositeCurve3D` und die Oberfläche einer Strasse als `CompositeSurface3D` modelliert werden.
- etc.

1.3 Konventionen

Verwendete Konventionen:

fett	Definitionen, neue Begriffe.
`courier`	Definitionen aus dem INTERLIS Datenmodell des Anhangs.
[1]	Verweis auf das Literaturverzeichnis im Anhang.

Die Begriffe, welche zur Beschreibung der Kurven und Flächen in [2] (Kapitel 2.8.12 und 2.8.13) eingeführt werden, gelten auch hier zur Beschreibung der 3D-Datentypen. Statt „Kurve“ oder „Linie“ werden wir daher wie im 2D-Fall den Begriff „Linienzug“ brauchen.

2. Beschreibung der 3D-Datentypen

2.1 Verwendung des 3D-Basismoduls

Damit in einem Anwendungsmodell mit den hier definierten 3D-Datentypen konkrete Koordinatensysteme verwendet werden können, muss zusätzlich zum Datenmodell `Geometry3D_V2` ein geeigneter INTERLIS 2.4 Kontext importiert werden. Beispiel für LV95:

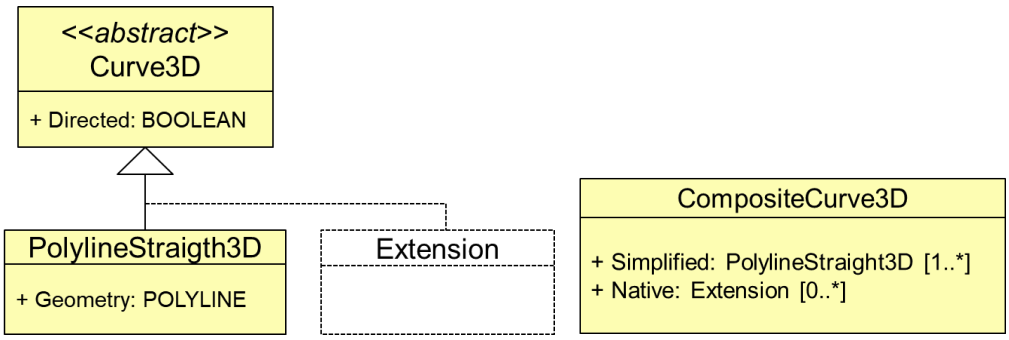
```
MODEL MeineAnwendung =

     IMPORTS Geometry3D_V2, ContextCHLV95_V2;
     ....

END MeineAnwendung.
```

2.2 Punkt-Typen

Der Raum ist eine (unendliche) Menge von Punkten. Ein Punkt als Element des Raumes ist durch das Zahlentripel seiner Koordinate im entsprechenden Referenzsystem eindeutig definiert. Der generische Datentyp `Coord3` wird für die Definition von Punkten direkt aus dem Geometrie-Basismodul `Geometry_V2` übernommen.

2.3 Linienzug-Typen

UML-Diagramm der 3D-Linienzug-Typen:



Curve3D

Mit dem Datentyp `Curve3D` werden (allgemeine) Linienzüge im Raum beschrieben. Der Datentyp `Curve3D` ist selber abstrakt. Alle einfachen Linienzug-Typen sind Spezialisierungen davon. Aktuell gibt es nur eine einzige konkrete Spezialisierung (`PolylineSraight3D`).

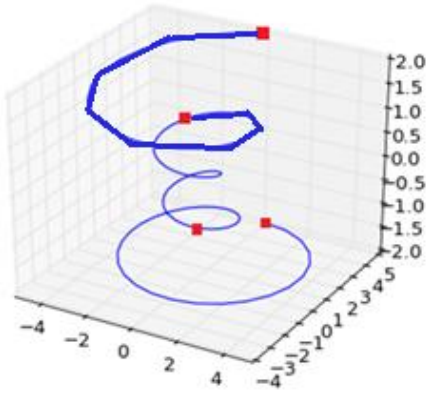
In Zukunft können gemäss dem Entwurfsmuster `Extension` weitere konkrete Linienzug-Typen (z.B. 3D-Splines) definiert werden. Linienzüge gemäss `Curve3D` können gerichtet sein oder auch nicht (Attribut `Directed`). Bei gerichteten `Curve3D` darf die Reihenfolge der Stützpunkte im Transfer nicht mehr verändert werden.

PolylineStraight3D

Beschreibt einen Linienzug im Raum, dessen Teilstücke ausschliesslich Geradenstücke sind.

CompositeCurve3D

Beschreibt eine Liste (d.h. eine endliche Folge) von Linienzügen. Die Linienzüge sind verbunden, d.h. ausser beim ersten und beim letzten Linienzug stimmt der Anfangspunkt eines Linienzugs der Liste jeweils mit dem Endpunkt des Vorgänger-Linienzuges überein. Die Linienzüge der `CompositeCurve3D` dürfen sich in der Projektion oder im Raum schneiden.



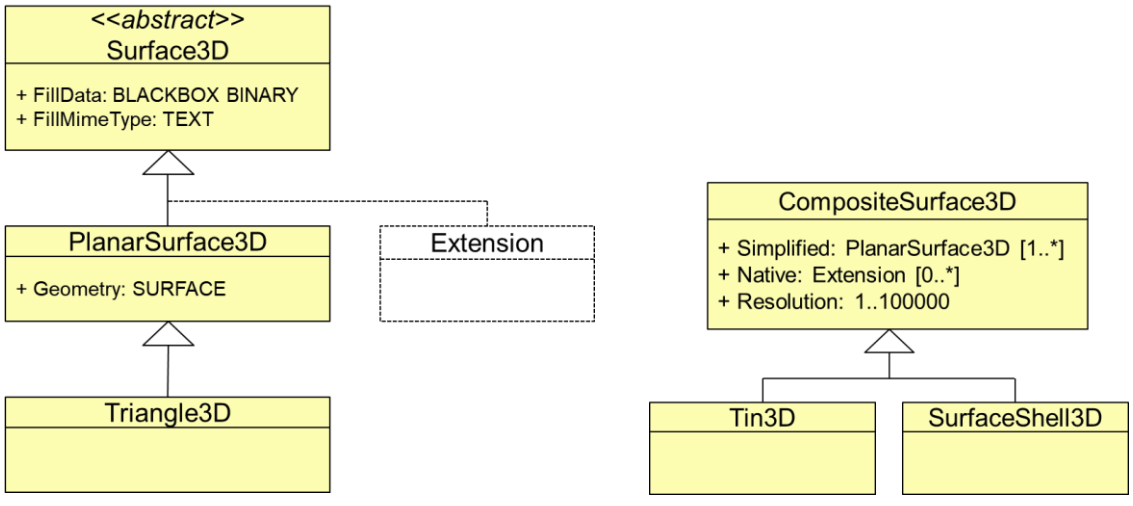
Beispiel für einen Linienzug vom Typ `CompositeCurve3D`

In diesem Beispiel sieht man einen Linienzug vom Typ `CompositeCurve3D` bestehend aus drei einfachen Linienzügen. Die Anfangs- und Endpunkte der einfachen Linienzüge sind rot markiert. Der erste Linienzug (von oben) ist eine `PolylineStraight3D`, die weiteren Linienzüge sind beliebige Spezialisierungen von `CompositeCurve3D`.

Wie bereits erläutert, ist aktuell nur die konkrete Kurvenform `PolylineStraight3D` für `CompsiteCurve3D` definiert. Um die Kurvenformen in Zukunft erweitern zu können (z.B. durch Splines) und trotzdem die Interoperabilität zwischen den Systemen zu gewährleisten, müssen alle Linienzüge immer als `PolylineStraight3D` im Attribut `Simplified` übertragen werden. Bei anderen Kurvenformen *muss zusätzlich* das Attribut `Native` mit der Originalgeometrie übertragen werden. Die Abweichung zwischen `Native` und `Simplified` darf nicht mehr als Resolution mal die geometrische Auflösung betragen (s.a. Kapitel 3 Interoperabilität).

2.4 Flächen-Typen

UML-Diagramm der 3D Flächen-Typen:



Surface3D

Alle einfachen 3D-Flächen-Typen sind Spezialisierungen des abstrakten Typs `Surface3D`. Aktuell gibt es nur die konkreten Spezialisierungen `PlanarSurface3D` und `Triangle3D`. In Zukunft können weitere konkrete Spezialisierungen (z.B. 3D-Spline Flächen) gemäss Entwurfsmuster `Extension` definiert werden. Der Typ `Surface3D` entspricht den bei 2.8.13.1 in [2] definierten „Flächen“. Über deren Rand kann nur gesagt werden, dass er aus endlich vielen Kurvenstücken besteht, die nur Endpunkte gemeinsam haben. Innere und äussere Ränder können vorerst nur für ebene Flächen definiert werden. Informationen zum Inneren der Fläche (z.B. Rendering-Daten) können mit `FillMimeType` / `FillData` übertragen werden. Die Struktur dieser Daten wird hier aber nicht weiter formalisiert.

PlanarSurface3D

Beschreibt eine **ebene Fläche**, d.h. eine Fläche, deren Punkte alle in einer Ebene E liegen. Eine ebene Fläche vom Typ `PlanarSurface3D` wird von einem äusseren und keinem, einem oder mehreren inneren Rändern begrenzt. Der äussere Rand ist im Gegenuhrzeigersinn, die inneren Ränder sind im Uhrzeigersinn orientiert. Läuft man im entsprechenden Drehsinn entlang den Rändern, liegt daher das Innere der Fläche immer links vom Betrachter. Die Rand-Kurven der inneren und äusseren Ränder dürfen nur aus Geradenstücken bestehen, müssen also geschlossene Linienzüge vom Typ `PolylineStraight3D` sein. Die Projektionsdistanz der Randpunkte auf die Ebene E darf höchstens so gross sein wie die geometrische Auflösung im Modell. Die Ebene E muss im Allgemeinen durch eine Ausgleichsrechnung der Eckpunkte der ebenen Fläche bestimmt werden (ausser bei Dreiecken).

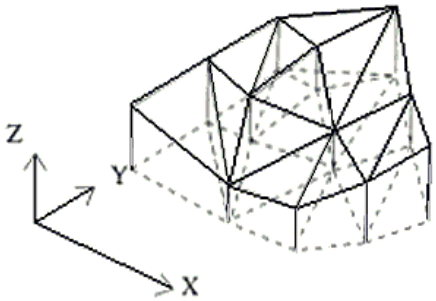
Triangle3D

Beschreibt ein **Dreieck**, d.h. ist eine Einschränkung des Typs `PlanarSurface3D`. Der Rand der Fläche besteht aus genau drei Stützpunkten.

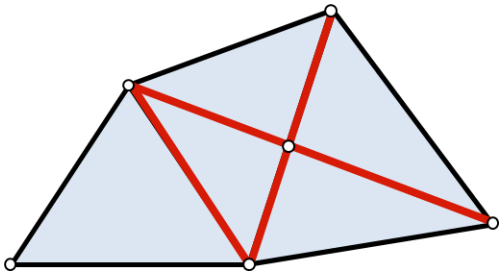
Mit der Funktion `getEdgePointCount()` wird die Anzahl der Stützpunkte in einem Constraint erzwungen. `getEdgePointCount()` zählt alle Stützpunkte des äusseren Rands / der inneren Ränder einer Fläche. Anfang- und Endpunkt der Ränder werden dabei nur einfach gezählt.

CompositeSurface3D

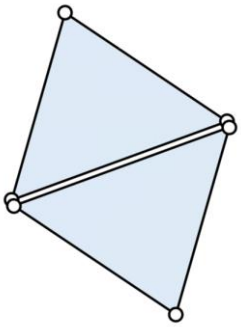
Beschreibt ein **Flächennetz**, d.h. die Vereinigung einer endlichen Menge von Flächen, die Spezialisierungen des Typs `Surface3D` sind und die höchstens Randpunkte gemeinsam haben. Anschaulich: Die (Teil-)Flächen eines Flächennetzes werden an den gemeinsamen Rändern miteinander „verklebt“. D.h. jeder Rand-Linienzug im Inneren einer Fläche vom Typ `CompositeSurface3D` wird jeweils von zwei (Teil-)Flächen des Flächennetzes verwendet. Die übrigen Rand-Linienzüge bilden die äusseren bzw. inneren Begrenzungen des Flächennetzes. Das Innere des Flächennetzes muss (evtl. mehrfach) zusammenhängend sein, d.h. jeder Punkt im Inneren des Flächennetzes vom Typ `CompositeSurface3D` ist mit jedem anderen inneren Punkt über (mindestens) einen Weg im Inneren verbunden, der weder einen Rand des Flächennetzes schneidet noch Stützpunkte von äusseren (oder inneren) Rändern des Flächennetzes enthält. Wenn das Innere des Flächennetzes zusammenhängend ist, dürfen sich Ränder des Flächennetzes in Stützpunkten berühren. Die Flächen dürfen sich gegenseitig durchdringen.



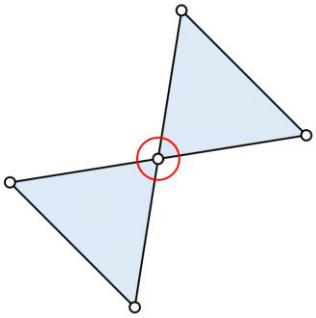
Beispiel für ein Flächennetz vom Typ `CompositeSurface3D` aus Dreiecken



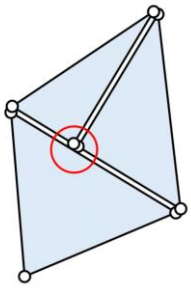
äussere Linienzüge (schwarz) und innere Linienzüge (rot)



Gültiges Flächennetz vom Typ `CompositeSurface3D`



Ungültig da Inneres nicht zusammenhängend



Ungültig da nicht alle inneren Linienzüge doppelt

Hinweise zur Interoperabilität von `CompositeSurface3D`, s.a. Kapitel 3.

Tin3D

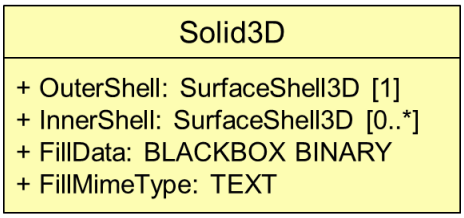
Ist eine Spezialisierung von `CompositeSurface3D`. Ein Flächennetz vom Typ `Tin3D` besteht nur aus Dreiecken vom Typ `Triangle3D`. Der Typ `Tin3D` eignet sich daher besonders für digitale Terrainmodelle (DTM).

SurfaceShell3D

Ist eine Spezialisierung von `CompositeSurface3D`, Das Flächennetz vom Typ `SurfaceShell3D` beschreibt eine geschlossene Hülle im Raum. *Jeder* Rand-Linienzug wird von zwei Teil-Flächen verwendet und die Teil-Flächen dürfen sich nicht gegenseitig durchdringen.

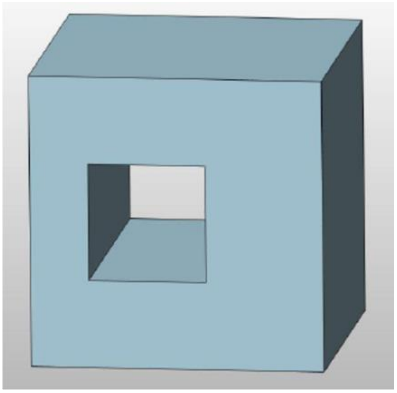
2.5 Körper-Typen

UML-Diagramm der 3D Körper-Typen:

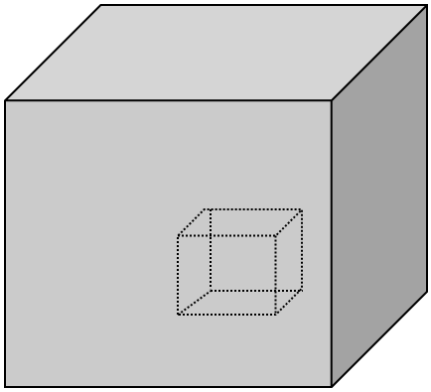


Solid3D

Ein Körper vom Typ `Solid3D` wird begrenzt von einer äusseren Hülle (`OuterShell`) und optional beliebig vielen inneren Hüllen (`InnerShells`) vom Typ `SurfaceShell3D`. Innere und äussere Hüllen dürfen sich höchstens in einer gemeinsamen Kante oder in einem gemeinsamen Eckpunkt berühren. Die inneren Hüllen müssen vollständig in der äusseren Hülle liegen. Das Innere des Körpers muss (evtl. mehrfach) zusammenhängend sein. D.h. jeder Punkt im Inneren des Körpers ist durch (mindestens) einen Weg (genauer: eine Klasse äquivalenter Wege) im Inneren des Körpers mit jedem anderen inneren Punkt verbunden. Dieser Weg darf weder Hüllen schneiden noch Stützpunkte von Hüllen enthalten. Sofern diese Bedingung erfüllt ist, dürfen sich Hüllen in gemeinsamen Eckpunkten oder in gemeinsamen Linienzügen berühren, nicht aber in Flächen.

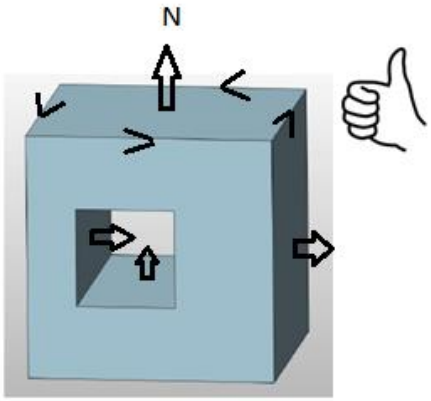


Beispiel mit einer OuterShell



Beispiel mit je einer Outer- und InnerShell

Der Normalenvektor der Teil-Flächen der Hüllen vom Typ `SurfaceShell3D` zeigt immer vom Inneren gegen die äussere Umgebung des Körpers. Die Richtung des Normalenvektor ist wie folgt definiert: Folgen die Finger der rechten Hand dem Umlaufsinn des äusseren Rands der Teil-Fläche, dann zeigt der Daumen in Richtung des Normalenvektors. Bemerkung: Während der INTERLIS Typ `SURFACE` keinen Umlaufsinn kennt, ist der Umlaufsinn von Teil-Flächen des Typs `SurfaceShell3D` wesentlich.



Normalenvektor N und Umlaufsinn

Nur die Hülle des Körpers wird in `Solid3D` explizit beschrieben. Das Innere des Körpers (z.B. Voxel-Daten) kann mit den Attributen `FillMimeType` / `FillData` übertragen werden. Die Struktur dieser Daten wird hier aber nicht weiter formalisiert.

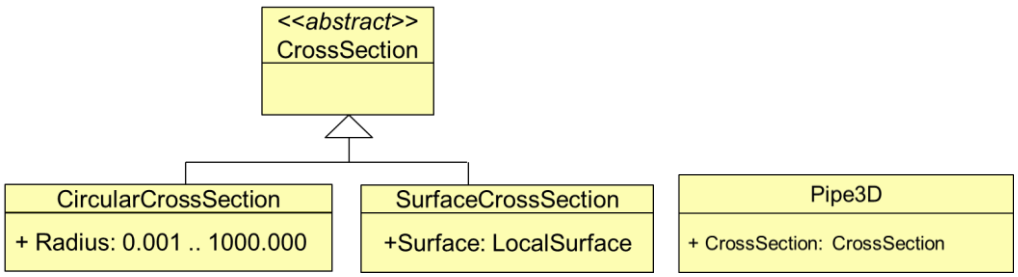
Funktion isCompositeSolid3D()

Mit der Funktion `isCompositeSolid()` kann in einem (`SET`) `CONSTRAINT` verlangt werden, dass eine Menge von `Solid3D` einen zusammengesetzten `Solid3D` bilden müssen. Dabei dürfen sich die Teil-Solids der Menge nicht gegenseitig durchdringen. Ausserdem muss die Menge zusammenhängend sein. D.h. jeder Punkt im Inneren des Körpers ist mit jedem anderen Punkt im Inneren durch einen Weg verbunden, welcher nur innere aber keine äusseren Flächen kreuzt.

2.6 Parameterische Typen

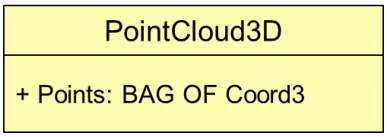
Pipe3D

Um Röhren (z.B. im Leitungskataster) nicht mit `Solid3D` aufwändig approximieren zu müssen, wird der parametrische Typ `Pipe3D` angeboten. Eine `Pipe3D` ist eine Spezialisierung des Typ `CompositeCurve3D` mit zugeordnetem Querprofil (`CrossSection`). Als Querprofil kann ein Kreis (`CircularCrosssection`) oder eine beliebige Fläche (`SurfaceCrossection`) angegeben werden.



2.7 Multi-Typen

Als einziger Multi-Typ wird für Lidardaten der Typ `PointCloud3D` angeboten. Damit wird es in Zukunft möglich für `PointCloud3D` effizientere Codierungsformen vorzusehen, als dies im Moment mit INTERLIS 2.4 möglich ist.



3. Interoperabilität

Um die Interoperabilität zwischen den Systemen zu erleichtern, wird für `CompositeCurve3D` und `CompositeSurface3D` verlangt, bei spezialisierten Geometrietypen gemäss Entwurfsmuster `Extension` die Geometrie in den Attributen `Simplified` und `Native` zu übertragen. `Native` enthält die originäre Geometriedefinition während `Simplified` aus in Geradenstücke bzw. planare Flächen aufgelöste Darstellung der originären Geometrie besteht. Programme welche `Extension` (noch) nicht unterstützen, können so die Geometrie aus `Simplified` verwenden.

Die maximale Abweichung zwischen `Simplified` und `Native` darf nicht mehr als `Resolution` mal die geometrische Auflösung im Modell betragen, also z.B. „1 mal ein Millimeter“. Mit `Resolution` kann daher bestimmt werden, wie genau `Native` durch `Simplified` approximiert wird.

Anhang

A. Basismodul 3D in INTERLIS 2.4

INTERLIS Modelldatei-Name: CHBase_Part8_GEOMETRY3D_20191218.ili

```
/* ######################################################################
   CHBASE - BASE MODULES OF THE SWISS FEDERATION FOR MINIMAL GEODATA
     MODELS
   ======
   BASISMODULE DES BUNDES           MODULES DE BASE DE LA
     CONFEDERATION
   FÜR MINIMALE GEODATENMODELLE     POUR LES MODELES DE GEODONNEES
     MINIMAUX

   PROVIDER: GKG/KOGIS - GCS/COSIG            CONTACT:
     models@geo.admin.ch
   DRAFT: 2023-03-06
   ######################################################################
*/

INTERLIS 2.4;

/* ######################################################################
   ######################################################################
   PART VIII -- GEOMETRY3D
   - Package Geometry3D
*/

!! ######################################################################
!!@technicalContact=mailto:models@geo.admin.ch
!!@furtherInformation=https://www.geo.admin.ch/de/geoinformation-
     schweiz/geobasisdaten/geodata-models.html
TYPE MODEL Geometry3D_V2 (en)
  AT "http://www.geo.admin.ch" VERSION "2023-03-21" =

  IMPORTS UNQUALIFIED INTERLIS;
  IMPORTS Units;
```

```
IMPORTS CoordSys;
IMPORTS Geometry_V2;

DOMAIN

  LocalCoord2 = COORD
     -1000.000 .. 1000.000 [INTERLIS.m],
     -1000.000 .. 1000.000 [INTERLIS.m],
    ROTATION 2 -> 1;
  LocalSurface = SURFACE WITH (STRAIGHTS,ARCS)
     VERTEX LocalCoord2 WITHOUT OVERLAPS > 0.001;

  !! abstract base structure for all 3d curve types
  STRUCTURE Curve3D (ABSTRACT) =
     Directed: MANDATORY BOOLEAN;
  END Curve3D;

  !! polyline in 3d
  STRUCTURE PolylineStraight3D EXTENDS Curve3D =
    Geometry: MANDATORY POLYLINE WITH (STRAIGHTS) VERTEX
   Geometry_V2.Coord3 WITHOUT OVERLAPS > 0.001;
  END PolylineStraight3D;

  !! composite (connected) curve
  STRUCTURE CompositeCurve3D =
    Simplified: LIST {1..*} OF PolylineStraight3D;
    Native: LIST {0..*} OF Curve3D;
    Resolution: 1 .. 100000; !! Default 1
  END CompositeCurve3D;

  STRUCTURE CrossSection (ABSTRACT) =
  END CrossSection;

  STRUCTURE CircularCrossSection EXTENDS CrossSection =
     Radius: MANDATORY 0.000 .. 1000.000 [INTERLIS.m];
  END CircularCrossSection;

  STRUCTURE SurfaceCrossSection EXTENDS CrossSection =
     Surface: MANDATORY LocalSurface;
  END SurfaceCrossSection;
```

```
STRUCTURE Pipe3D EXTENDS CompositeCurve3D =
    CrossSection: MANDATORY CrossSection;
END Pipe3D;

!! abtract base structure for all 3d surface types
STRUCTURE Surface3D (ABSTRACT) =
  FillData: BLACKBOX BINARY;
  FillMimeType: TEXT*100;
END Surface3D;

!! function for calculating edge point count
FUNCTION getEdgePointCount(geometry:SURFACE):NUMERIC;

!! planar surface in 3D with no arcs
STRUCTURE PlanarSurface3D EXTENDS Surface3D =
  Geometry: MANDATORY SURFACE WITH (STRAIGHTS) VERTEX
 Geometry_V2.Coord3 WITHOUT OVERLAPS > 0.001;
END PlanarSurface3D;

!! planar surface with only three points no arcs
STRUCTURE Triangle3D EXTENDS PlanarSurface3D =
MANDATORY CONSTRAINT
  getEdgePointCount(Geometry) == 3;
END Triangle3D;

!! non planar surface in 3D
!! all inner edge curves are shared by two surfaces
!! except edge curves of composite boundary(ies)
!! surfaces may intersect
STRUCTURE CompositeSurface3D =
  Simplified: BAG {1..*} OF Triangle3D;
  Native: BAG {0..*} OF Surface3D;
  Resolution: 1 .. 100000; !! Default 1
END CompositeSurface3D;

!! triangle network (TIN)
STRUCTURE Tin3D EXTENDS CompositeSurface3D =
END Tin3D;

!! closed surface shell
!! all surface edge curves are shared by two surfaces
```

```
        !! surfaces may not intersect
        STRUCTURE SurfaceShell3D EXTENDS CompositeSurface3D =
        END SurfaceShell3D;

        !! 3D solid with one outer shell and 0..* inner shells
        !! shells may not intersect
        !! inner shells are enclosed by outer shell
        STRUCTURE Solid3D =
          OuterShell: MANDATORY SurfaceShell3D;
          InnerShells: BAG {0..*} OF SurfaceShell3D;
          FillData: BLACKBOX BINARY;
          FillMimeType: TEXT*100;
        END Solid3D;

        !! collection of 3D points
        STRUCTURE PointCloud3D =
          Points: BAG {1..*} OF Geometry_V2.Coord3;
        END PointCloud3D;

        !! function for testing composite solids in set constraints
        FUNCTION isCompositeSolid3D(solids:BAG OF Solid3D):BOOLEAN;

    END Geometry3D_V2.

!! ##########################################################################
```

B. Literaturverzeichnis

[1] Basismodule des Bundes für „minimale Geodatenmodelle“
https://www.geo.admin.ch/content/geo-internet/de/geo-information-switzerland/geobasedata-harmonization/geodata-models/_jcr_content/contentPar/tabs/items/hilfsmittel_f_r_die_/tabPar/downloadlist_1419651270/downloadItems/5_1458208422619.download/basismoduledesbundeschbasev.1.020120118.pdf

[2] eCH-0031: INTERLIS 2.4 Referenzhandbuch
https://www.ech.ch/dokument/266625ed-0a4a-4c20-b7ff-504cd908c0d6