



## Modeling Guide for 3D Objects

### Part 1: Basics ( Rules for Validating GML Geometrics in CityGML )

SIG3D Quality Working Group

Version 0.7.1 EN - November 2014

<http://www.sig3d.org>  
<http://en.wiki.quality.sig3d.org>

#### Document History

Version	Date	Author/s	Remarks
0.1.0	2010-10-01	Coors	First draft
0.2.0	2010-12-14	Coors	Rules for solid type added
0.3.0	2010-10-23	Gröger	Merging with document by G. Gröger
0.4.0	2010-11-23	Coors	Editorial changes, rules for planarity of polygons added
0.5.0	2010-11-24	Gröger	Comments by K.-H. Häfele (CityGML forum) included
0.6.0	2010-12-15	Gröger	Comments by G. Juen included
0.7.0 EN	2012-01-20	Coors, Häfele	English version with minor editorial changes, CompositeSolid added
0.7.0 EN	2013-11	Casper, Häfele	English version
0.7.1 EN	2014-11-11	Casper	Derived PDF version

# Modeling Guide for 3D Objects - Part 1: Basics ( Rules for Validating GML Geometrics in CityGML )

## Table of Content

1	Spatial Reference Systems (SRS) .....	5
2	Definitions .....	5
2.1	Geometric Tolerances .....	5
2.2	Planarity of Polygons .....	5
3	gml:posList.....	6
4	gml:_Curve, gml:LineString .....	6
5	gml:LinearRing .....	6
6	gml:Polygon.....	9
7	gml:OrientableSurface.....	10
8	gml:MultiSurface.....	11
9	gml:CompositeSurface .....	12
10	gml:Solid .....	13
11	gml:CompositeSolid .....	15
12	gml:Triangle.....	16
13	gml:TriangulatedSurface .....	16
14	gml:Tin .....	16

## References

- Cox, S., Daisey, P., Lake, R., Portele, C., Whiteside, A. (2004): OpenGIS Geography Markup Language (GML) Implementation Specification. Version: 3.1.1, OGC 03-105r1,
- Herring J. (2001): The OpenGIS Abstract Specification, Topic 1: Feature Geometry (ISO 19107 Spatial Schema), Open Geospatial Consortium, OGC Document Number 01-101. OGC.
- Gröger, G., Kolbe, T., Czerwinski, A., and Nagel, C. (2008) (Eds.): OpenGIS City Geography Markup Language (CityGML) Encoding Standard, OGC reference number OGC 08-007r1 version 1.0.0, 2008
- Gröger, G., Plümer, L. (2011): How to Achieve Consistency for 3D City Models. Geoinformatica, 15(1):137-165, DOI: 10.1007/s10707-009-0091-6.

## Preliminaries

- The rules refer to GML Version 3.1.
- The rules are based on the validity of the data with regard to the XML-Schema of GML 3.1.
- The rules are restricted to the elements of GML that are used in CityGML (Profile of GML, c.f. Figure 1 and Figure 2).
- If a condition applies to CityGML only (not to GML in general), it is stated explicitly.

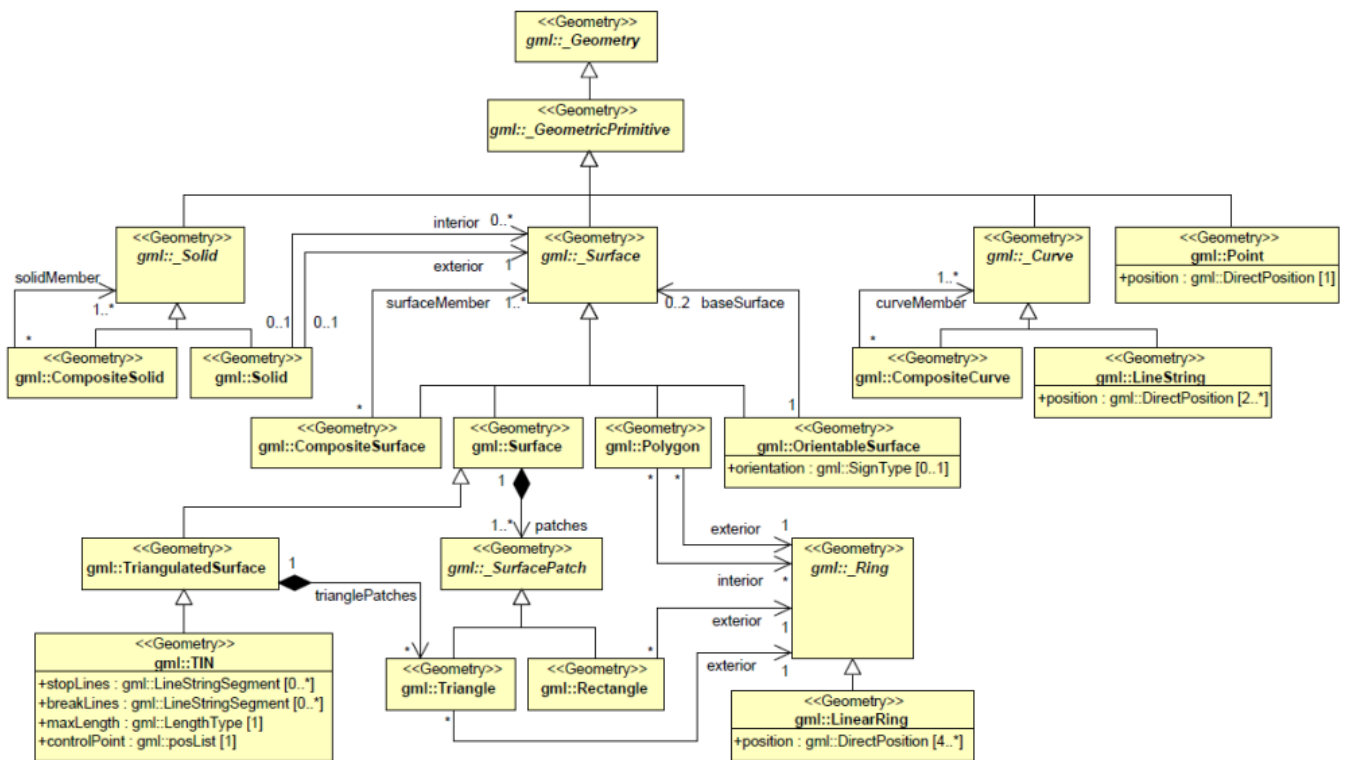


Figure 1: Profile of GML that is used in CityGML (Primitives and Composites)

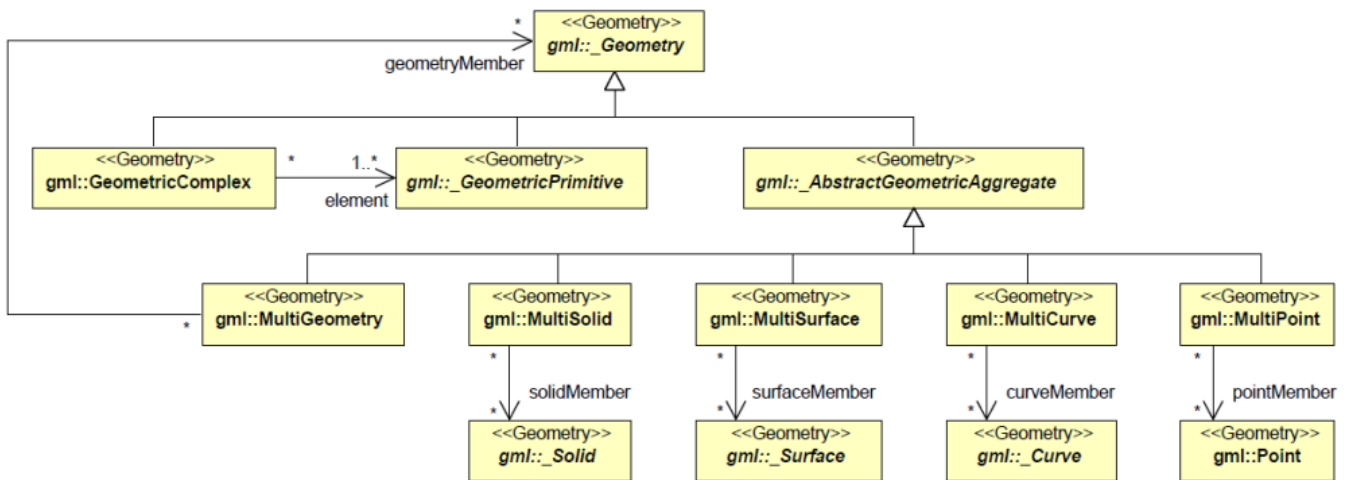


Figure 2: Profile of GML that is used in CityGML (Complexes and Aggregates)

# 1 Spatial Reference Systems (SRS)

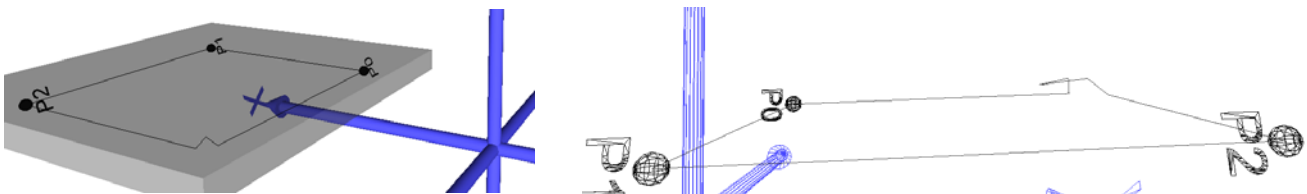
- Each geometry element (including pos, posList and coordinates) must
  - either have a srsName attribute with proper value, or
  - inherit the value of the srsName attribute from
    - its parent geometries (even if its obtained recursively from the parents of the parents, etc.), or
    - the gml:Envelope (or gml:Box) which is the value of the gml:boundedBy property of the parent feature or the parent feature collection (even if it is obtained recursively).
- A local value of a srsName attribute precedes the inherited value of a srsName attribute.
- If in geometries (pos, posList) the dimension of a SRS can not be derived from the SRS without a doubt, the attribute srsDimension must contain the dimension of the SRS.

## 2 Definitions

### 2.1 Geometric Tolerances

### 2.2 Planarity of Polygons

„Surfaces in CityGML are represented by polygons, which define a planar geometry, i.e. the boundary and all interior points are required to be located in one single plane.“ (Gröger et al., 2008). The boundary of a Polygon is defined by one or more planar Linear Rings (see gml:Polygon). From practical point of view, Linear Rings that are not exactly planar should be accepted as a valid as well.



**Figure 3:** : Bends and folds should not appear in planar Linear Rings

Intuitively, a Linear Ring  $R$  would be defined as planar if there is a plane  $E$  with distance  $\epsilon$  for all points  $P_i$  of  $R$ . However, this is true for Linear Rings with small bends and folds as shown in Figure 3 as well.  $\epsilon$  less than a given threshold

In order to avoid such situations, a slightly different definition of planarity of Linear Rings is introduced:

**Definition 1:** A Linear Ring  $R$  is **planar**, if it is a valid Linear Ring and the distance of all points to any plane  $E_{ijk}$ , that is defined by three co-linear points  $P_i, P_j$  and  $P_k$ , is less than a given threshold  $\epsilon$  :

$$\forall E_{ijk} \forall P_a = \text{dist}(P_a, E_{ijk}) \leq \epsilon$$

### 3 [gml:posList](#)

- The number of items in the list should correlate with the product of the SRS dimension (usually in CityGML 3) and the number of positions.

### 4 [gml:Curve](#), [gml:LineString](#)

- Only gml:LineStrings are allowed as subtypes of \_Curves (rule applies to CityGML only)
- A gml:LineString has linear interpolation.
- For the child elements of a gml:LineString (control points) the following statements must hold true:
  - They form a sequence of "pos" (DirectPositionType) or "pointProperty" (PointPropertyType) elements. "pos" elements are control points that belong to this curve only. "pointProperty" elements contain points that may be referenced from other geometries or that reference other points (via XLinks) which are defined in other geometries.
  - Alternatively, a gml:LineString can be composed of a "posList" element, providing a compact way of representing coordinates. In that case all control points must belong to the same Curve and must be given in the same SRS. The number of points (Direct Position) must be at least two.
- Each control point in a gml:LineString occurs only once, beside the first and the last one which may both be identical.
- Line segments of a gml:LineString are defined by pairs of two subsequent control points. Those line segments must not intersect mutually and must not have any identical points. Exceptions of that rule are the pairs which consist of an endpoint of one line segment and the start point of the next line segment in the sequence.

### 5 [gml:LinearRing](#)

The Linear Ring is the fundamental element to describe 3D geometry in CityGML. Every single polygon of a building geometry is defined by its boundary – the Linear Ring.

A sequence is an ordered list of elements. Unlike a set, order matters, and the exact same elements can appear multiple times at different positions in the sequence. A finite sequence  $a$  with  $n + 1$  elements is denoted as  $a = (a_0, a_1, \dots, a_n)$ . The empty sequence  $a = ()$  has no elements.

A finite sequence of points  $R = (P_0, P_1, \dots, P_n), n \geq 3, P_i = (x_i, y_i, z_i)$  is a Linear Ring if:

i. the first and last point  $P_0$  and  $P_n$  represent the same point:  $P_0 = P_n$  (**closeness**)

ii. All points of the sequence besides start and end point are different:

$$\forall_{\substack{i=0..n-1 \\ k=0..n-1 \\ i \neq k}} P_i \neq P_k$$

iii. Two edges  $(P_i, P_{i+1})$  and  $(P_k, P_{k+1})$   $i = 0 \dots n - 1, k = 0 \dots n - 1, i \neq k$  do only intersect in one start-/ endpoint. No other intersection is allowed (**no self intersection**).

If all points of the sequence are co-planar, the Linear Ring is **planar**.

**Examples:**

$$R = (P_0, P_1, P_2, P_3, P_0)$$

with

$$P_0 = (1,1,1)$$

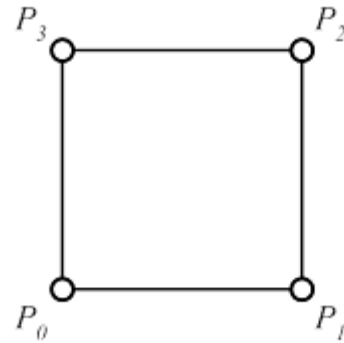
$$P_1 = (3,1,1)$$

$$P_2 = (3,3,1)$$

$$P_3 = (1,3,1)$$

```
<gml:LinearRing>
  <gml:pos> 1 1 1 </gml:pos>
  <gml:pos> 3 1 1 </gml:pos>
  <gml:pos> 3 3 1 </gml:pos>
  <gml:pos> 1 3 1 </gml:pos>
  <gml:pos> 1 1 1 </gml:pos>
</gml:LinearRing>
```

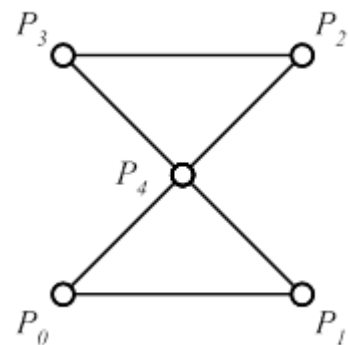
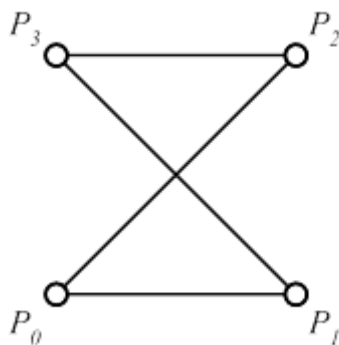
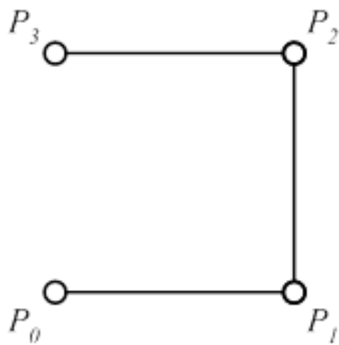
>



**Figure 4:** Planar Linear Ring: definition, GML and graphic representation

**Please note:**

The two Linear Rings  $R_1 = (P_0, P_1, P_2, P_3, P_0)$  and  $R_2 = (P_0, P_3, P_2, P_1, P_0)$  are not identical.



$$R = (P_0, P_1, P_2, P_3)$$

$$R = (P_0, P_1, P_3, P_2, P_0)$$

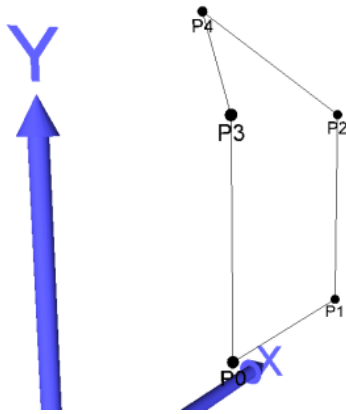
$$R = (P_0, P_1, P_4, P_3, P_2, P_4, P_0)$$

No Linear Ring: not closed (see (i) closeness)

No Linear Ring: self intersection (see (iii) self-intersection)

No Linear Ring: P<sub>4</sub> appears twice in the sequence (see (ii), P<sub>4</sub>=(2,2,1))

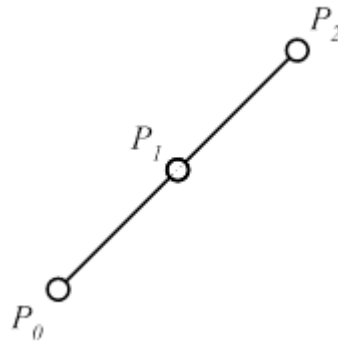
**Figure 5:** Examples of non-valid Linear Rings



$$R = (P_0, P_1, P_2, P_4, P_3, P_0)$$

Linear Ring, but not planar

$$P_4=(2,4,0)$$



$$R = (P_0, P_1, P_2, P_0)$$

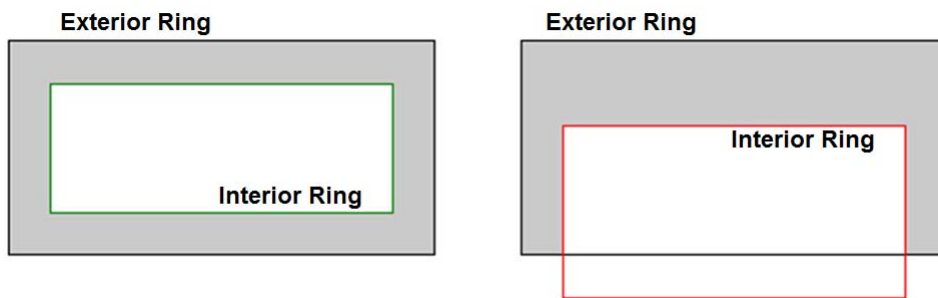
No Linear Ring: all 3 points are co-linear; self-intersection (see (iii))

**Figure 6:** Special cases: non-planar and co-linear Linear Ring Elements.



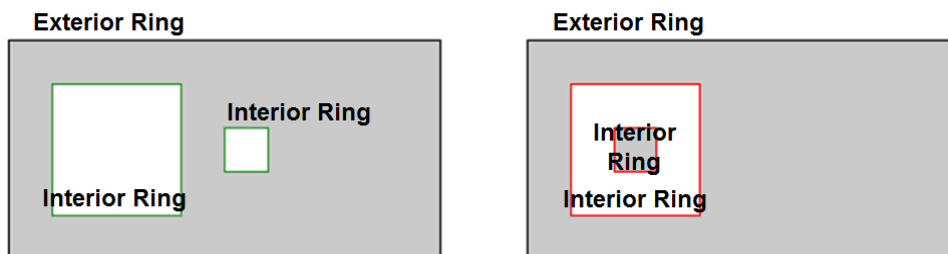
## 6 gml:Polygon

- A Polygon  $S$  is defined by a planar Linear Ring  $R_s$  (exterior ring
- A polygon has exactly one exterior ring. In addition, a polygon can have  $n \geq 0$  interior rings. These interior rings a planar Linear Rings as well. All interior rings and the exterior ring must be on the same plane (within the given tolerance).
- As interior rings define holes in the polygon, they have to be completely included in the area defined by the exterior ring.



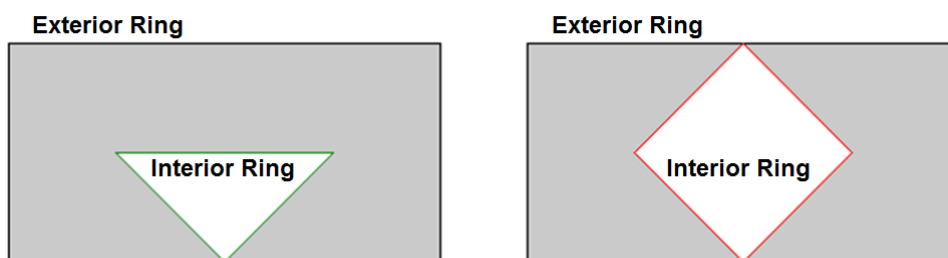
**Figure 7:** Polygon with **correct interior ring (left)** and **incorrect interior ring (right)**

- Interior rings must not overlap other Linear Rings or be included in another Linear Ring (except the exterior Linear Ring).



**Figure 8:** Polygon with **two correct interior rings (left)** and two interlaced, **incorrect interior rings (right)**

- Interior rings and the exterior ring may touch each other in a finite number of points. The inner area of the polygon has to be connected.



**Figure 9:** Polygon with **correct, interior ring touching the outer ring at exactly one point (left)** and **incorrect interior ring touching the outer ring at two points and splitting the polygon (right)**

- The order of points of the exterior ring defines the orientation of the polygon. The non co-linear points  $P_i$ ,  $P_j$ , and  $P_k$ , define the plane  $E (P_i, P_j, P_k)$ . The normal vector  $\vec{n}$ , is given by the cross product of the two vectors  $\overrightarrow{P_i P_j}$  and  $\overrightarrow{P_j P_k}$  :

$$\vec{n} = \frac{\overrightarrow{P_i P_j} \times \overrightarrow{P_j P_k}}{\left\| \overrightarrow{P_i P_j} \times \overrightarrow{P_j P_k} \right\|}$$

**Please note:** that due to the tolerance of planar Linear Rings, the normal vector can be ambiguous.

Remark:

The two Linear Rings  $R_1 = (P_0, P_1, P_2, P_3, P_0)$  and  $R_2 = (P_0, P_3, P_2, P_1, P_0)$  define polygons with the same geometric extend but different orientation.

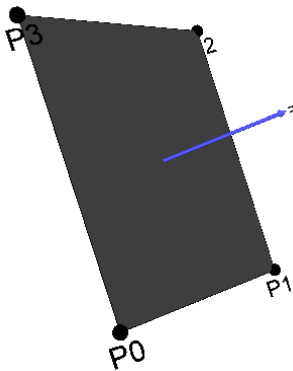


Figure 10: Polygon with normal vector n

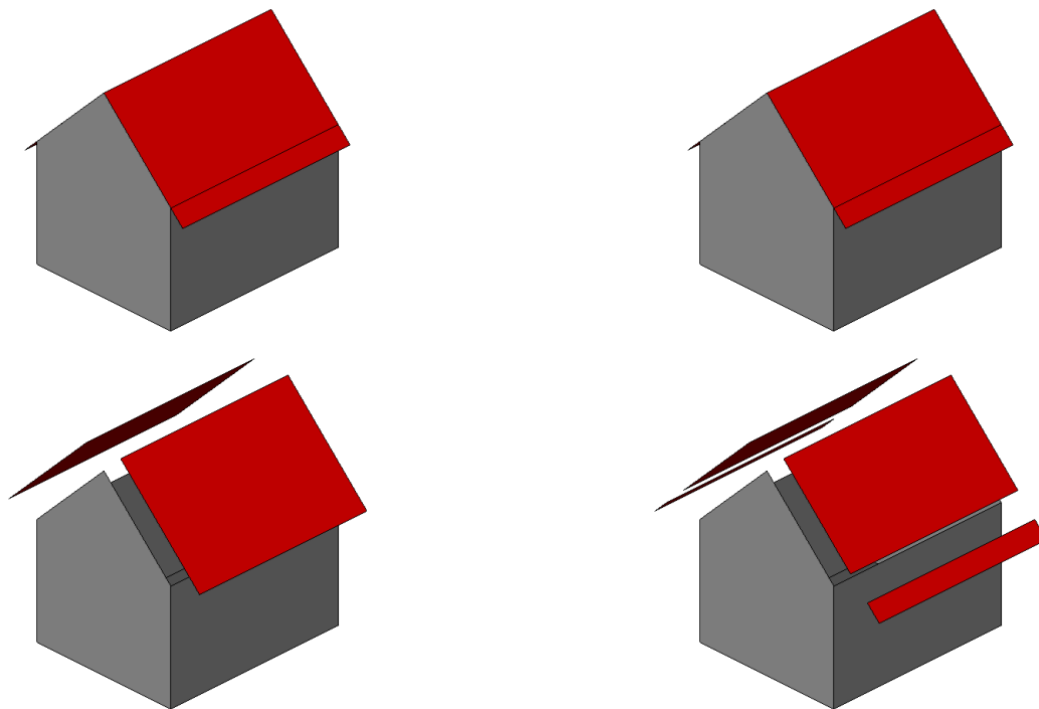
## 7 [gml:OrientableSurface](#)

- If the value of the attribute "orientation" of an OrientableSurface is "+", the orientation of the OrientableSurface is identical to the one of the baseSurface. If the value of "orientation" is "-", the normal of the OrientableSurface has a direction which is opposite to the normal of the baseSurface. The default value of "orientation" is "+".
- Either the element "baseSurface" references the base surface via a XLink attribute or the element "baseSurface" contains the base surface as child element. The base surface has positive orientation.

## 8 [gml:MultiSurface](#)

A **MultiSurface** is an unstructured set  $M = \{S_1, S_2, \dots, S_n\}$  of polygons. No further constraints are defined for a MultiSurface element. It is not necessary that  $M$  is a connected surface. Orientation of the polygons is arbitrary. By a MultiSurface any existing polygonal boundary representation of a rigid body can be described. Unfortunately, also objects that do not exist in real space can be modeled.

Due to the missing constraints it is recommended that MultiSurface elements should only be used to model non-volumetric parts of the building model as shown in Figure 11. Volumetric parts should be modeled as solids (see `gml:Solid`)



**Figure 11:** Modeling overlapping roof parts (**above:** complete models, polygons outlined; **below:** roof polygons exploded)

**Please note:** the left model in Figure 11 is created with a MultiSurface element containing 7 polygons. However, the polygons do not represent a valid boundary of a rigid body. Calculating a volume is not possible. The right model is the same building geometry but modeled with 1 Solid element (7 polygons) and 1 MultiSurface element (2 polygons F6 and F8) as recommended (see CityGML Standard v1.0,p.61)

## 9 gml:CompositeSurface

Eine CompositeSurface ist eine Menge  $C = \{S_1, S_2, \dots, S_n\}$  von Polygonen, für die folgendes gilt:

Die Schnittmenge zweier Polygone  $S_k$  und  $S_l$  aus  $C$  ist entweder leer oder besteht nur aus Punkten und/oder Kanten, die auch in den beiden linearen Ringen vorkommen. Bezeichne

$R_k = (P_0^k, P_1^k, \dots, P_n^k)$  den planaren linearen Ring, der das Polygon  $S$  definiert. Dann gilt:

$$S_i \cap S_k = \begin{cases} \emptyset \\ \{Q_0, Q_1, \dots, Q_m\}, Q_j = P_k^i \\ \{e_0, e_1, \dots, e_m\}, e_j = \overline{P_i^k P_{i+1}^k} \end{cases}$$

1. Jede Kante  $e_k = \overline{P_i^k P_{i+1}^k}$  eines linearen Rings  $R_k = (P_0^k, P_1^k, \dots, P_n^k)$ , der ein Polygon  $S_k \in C$  definiert, wird höchstens einmal als Kante  $e_l = \overline{P_j^l P_{j+1}^l}$  in einem linearen Ring  $R_l = (P_0^l, P_1^l, \dots, P_m^l)$  genutzt, der ein anderes Polygon  $S_l \in C$  definiert.  
Es gilt  $P_i^k = P_{j+1}^l$  und  $P_{i+1}^k = P_j^l$ .
2. Die Polygone aus  $C$  sind so orientiert, dass die Flächennormale benachbarte Polygone in dieselbe Richtung zeigen.
3. Die Vereinigung aller Polygone aus  $C$  ohne die Kanten oder Punkte, in denen sich die Polygone berühren, ist isomorph zu einem Polygon.

Aus (1) und (2) ergibt sich, dass die Oberfläche, die durch  $C$  beschrieben wird, keine sich gegenseitig überlappenden oder durchdringenden Polygone enthalten darf (Polygone berühren sich höchstens in Punkten oder Kanten).

## 10 gml:Solid

A solid geometry represents a rigid body. The surface of the solid is defined by a set of polygons with the following properties. The set  $C = \{S_1, S_2, \dots, S_n\}$  of polygons bounds a solid if:

- i. The intersection of two polygons  $S_k$  and  $S_l$  of  $C$  is either empty or contains only points  $P$  and edges  $e$  that are part of both Linear Rings.

The polygon  $S_k$  is defined by the Linear Ring  $R_k = (P_0^k P_1^k, \dots, P_n^k)$ . The intersection of  $S_k$  and  $S_l$  equals:

$$S_l \cap S_k = \begin{cases} \emptyset \\ \{Q_0, Q_1, \dots, Q_m\}, Q_j = P_i^k \\ \{e_0, e_1, \dots, e_m\}, e_j = \overline{P_i^k P_{i+1}^k} \end{cases}$$

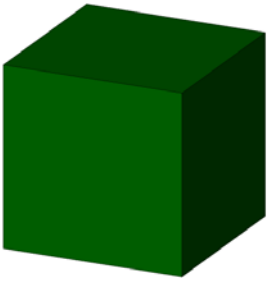
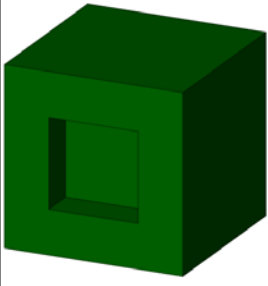
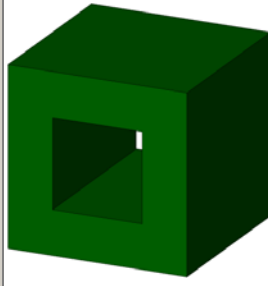
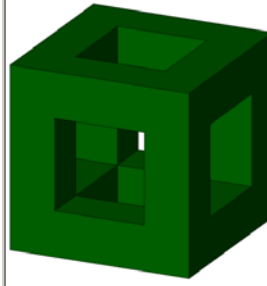
- ii. Every Edge  $e_k = \overline{P_i^k P_{i+1}^k}$  of a Linear Ring  $R_k = (P_0^k P_1^k, \dots, P_n^k)$ , defining a polygon  $S_k \in C$ , is used exactly once as an edge  $e_l = \overline{P_j^l P_{j+1}^l}$  in a Linear Ring  $R_l = (P_0^l P_1^l, \dots, P_m^l)$  defining another polygon  $S_l \in C$  with  $P_i^k = P_{j+1}^l$  and  $P_{i+1}^k = P_j^l$ .
- iii. All polygons in  $C$  are oriented such that the normal vector of each polygon points to the outside of the solid.
- iv. All polygons in  $C$  are connected, that is the dual graph of  $C$  has a path containing all nodes. The dual graph  $G_C = (V_C, E_C)$  of  $C$  consists of a set  $V_C$  of nodes and a set  $E_C$  of edges. Every node  $v$  of  $V_C$  represents exactly one polygon of  $C$ . An edge shared by two polygons  $S_k$  and  $S_l$  of  $C$  is represented by an edge  $e = (v_{S_k}, v_{S_l})$  in  $E_C$ .
- v. For every point  $P$  of a Linear Ring of a polygon of  $C$  applies: The graph  $G_P = (V_P, E_P)$ , that is only build by polygons and edges that touch  $P$  is connected. Each node  $v$  of  $V_P$  represents a polygon which Linear Ring contains  $P$ . Two nodes are connected by an edge  $e$  of  $E_P$  iff the two polygons represented by the nodes have a common edge that touches  $P$ <sup>1</sup>.

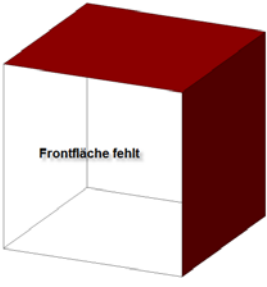
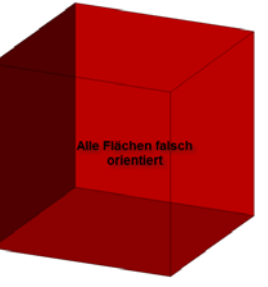
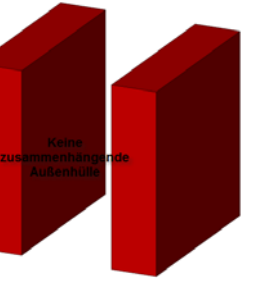
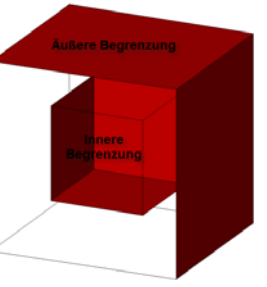
It follows from (i) and (ii) that the surface defined by  $C$  has no holes. Together with conditions (iv) and (v) it follows that the inner of the solid defined by  $C$  is connected.

$S$  is denoted as closed [CompositeSurface](#).

<sup>1</sup> The condition (v) is equivalent to the *umbrella axiom* presented in Gröger & Plümer 2011. This axiom is defined as follows: Each point is surrounded by exactly one cycle that is an alternating sequence of line segments and polygons.

**Examples:**

			
correct: 6 surfaces, closed, correct surface orientation	correct: 11 surfaces, closed, correct surface orientation	correct: 10 surfaces, closed, correct surface orientation	correct: 30 surfaces, closed, correct surface orientation

			
incorrect: 5 surfaces, not closed, correct surface orientation	incorrect: 6 surfaces, closed, incorrect surface orientation	incorrect: 12 surfaces, no closed (zwei Außenhüllen), correct surface orientation	incorrect: 12 surfaces, not closed (innere und äußere Hülle), correct surface orientation

# 11 gml:CompositeSolid

Ein CompositeSolid wird durch eine nichtleere Menge  $C = \{S_1, \dots, S_n\}$  von Solids (vgl. 12.) beschrieben, für die gilt:

1. Der Schnitt der Inneren zweier Solids  $S_i, S_j, 1 \leq i < j \leq n$ , ist leer, d.h. entweder sind beide Solids  $S_i, S_j$  disjunkt oder beide berühren sich nur in Flächen, Linien oder Punkten
2. Sei  $C'$  die Vereinigung aller Solids aus  $C$ . Dann ist die Begrenzung von  $C'$  (die Oberfläche von  $C'$  ohne die Flächen oder Punkte, in denen sich die Solids berühren) die Begrenzung eines Solid.

In CityGML werden bei CompositeSolids ebenso wie bei Solids innere Einschlüsse (Hohlräume) nicht betrachtet.

### Examples:

correct; surface contact	correct; surface contact	correct; surface contact	correct; surface contact
incorrect; line contact	incorrect; point contact	incorrect; no contact at all	incorrect; intersection
incorrect; 26 boxes with surface contact	but with internal hollow space	incorrect; interior boundaries	

## 12 [gml:Triangle](#)

- A Triangle is a special case of a polygon that is bounded by one outer Linear Ring with four points.
- There are no interior rings (no element "interior").

## 13 [gml:TriangulatedSurface](#)

A triangulated surface is a [CompositeSurface](#) that consists of triangles only, but is not closed (it may have boundaries). There is no restriction on how the triangulation is derived.

## 14 [gml:Tin](#)

In contrast to a [gml:TriangulatedSurface](#), the triangles of a [gml:TIN](#) are not represented explicitly, but by its three vertices (called control points). In addition, breaklines, stop lines and the maximal length of a side of a triangle can be represented in a [gml:TIN](#).

1. A TIN is a triangulated surface that is the result of the application of a Delaunay Algorithm or of a similar algorithm (extended by breaklines, stop lines and the maximal length of a side of a triangle). The surface satisfies the Delaunay criterion:
  1. For each triangle in the network, the circle passing through its vertices does not contain, in its interior, the vertex of any other triangle.
2. Stoplevelines are lines where the local continuity or regularity of the surface is questionable. In the area of these pathologies, triangles intersecting a stopline shall be removed from the TIN surface, leaving holes in the surface. If triangles incident to the boundary of the surfaces are removed, the boundary of the surfaces changes.
3. Breaklines are lines that are critical for the shape of the surface, representing local ridges, or depressions (such as drainage lines) in the surface. As such their constituent segments must be included in the TIN even if doing so violates the Delaunay criterion.
4. Areas of the surface where the data is not sufficiently dense to assure reasonable calculations shall be removed by adding a retention criterion for triangles based on the length of their sides. For any triangle sides exceeding maximum length, the adjacent triangles to that triangle side shall be removed from the surface. # The number of control points is at least three.
5. The surface is not affected by the order in which the control points are given. Application schemas may add information based on the ordering of the control points to facilitate the reconstruction of the TIN from the control points.