



# Handbuch für die Modellierung von 3D-Objekten

## Teil 1: Grundlagen

### (Regeln für valide GML Geometrie-Elemente in CityGML)

SIG3D AG Qualität

Version 0.7.1 DE - November 2014

<http://www.sig3d.de>  
<http://wiki.quality.sig3d.de>

#### Dokumentversionen

Version	Datum	Autor/en	Bemerkungen
0.1.0	01.10.2010	Coors	Erstfassung
0.2.0	14.10.2010	Coors	Solid ergänzt
0.3.0	23.10.2010	Gröger	Verschmelzung mit Paper Gröger
0.4.0	23.11.2010	Coors	Redaktionelle Änderungen und Ergänzungen zu Planarität von Polygonen
0.5.0	24.11.2010	Gröger	Einarbeitung der Änderungen, die im CityGML-Forum von K.-H. Häfele vorgeschlagen wurden
0.6.0	15.12.2010	Gröger	Einarbeitung der Anregungen von G. Juen
0.7.0	20.01.2012	Häfele	Konvertierung in Html, Verschiebung des Kapitel Planarität von Polygonen, Neues Kapitel CompositeSolid
0.7.1	11.11.2014	Casper	Abgeleitete PDF-Version

# Handbuch für die Modellierung von 3D-Objekten - Teil 1: Grundlagen (Regeln für valide GML Geometrie-Elemente in CityGML)

## Inhaltsverzeichnis

1	Spatial Reference Systems (SRS) .....	5
2	Definitionen .....	5
2.1	Geometrische Toleranzen .....	5
2.2	Planarität von Polygonen .....	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

## Referenzen

- 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.

## Vorbemerkungen

- Die Regeln beziehen sich auf GML Version 3.1.
- Es wird davon ausgegangen, dass die Daten valide sind bzgl. des XML-Schemas von GML 3.1.
- Die Regeln sind eingeschränkt auf die Elemente von GML, die von CityGML genutzt werden (Profil von GML, siehe Abbildung 1 und Abbildung 2).
- Falls eine Bedingung nur für CityGML und nicht für GML im Allgemeinen gilt, so ist dies explizit vermerkt.

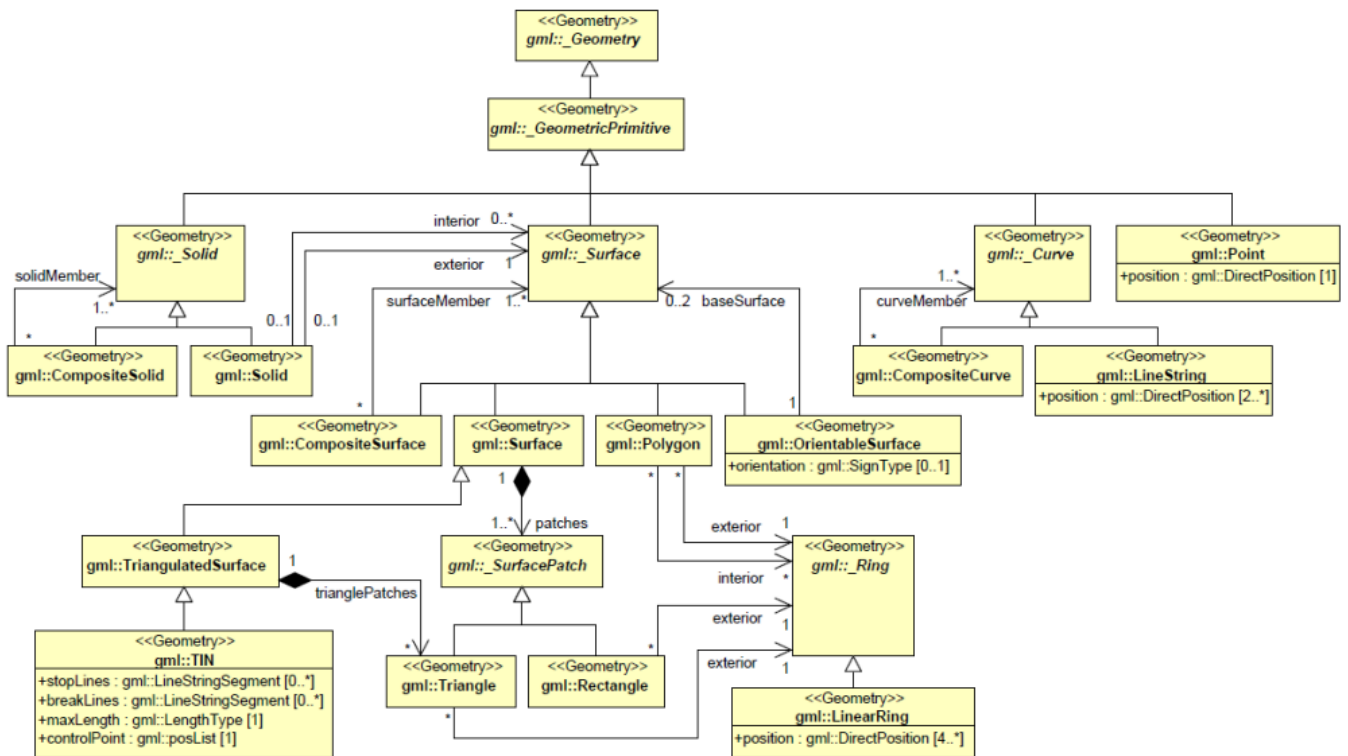


Abbildung 1: Profil von GML, das in CityGML verwendet wird (Primitive und Komposite)

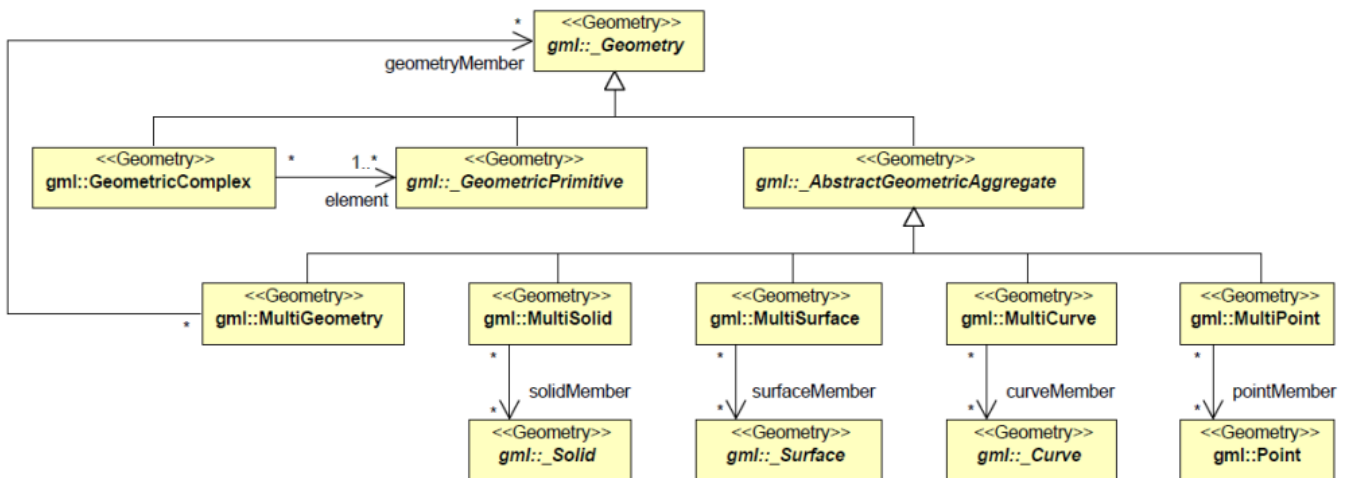


Abbildung 2: Profil von GML, das in CityGML verwendet wird (Komplexe und Aggregate)

# 1 Spatial Reference Systems (SRS)

- Jedes Geometrie-Element (einschließlich pos, posList and coordinates) muss
  - entweder im srsName-Attribut mit einem Wert belegt sein, oder
  - den Wert des srsName-Attributs erben
    - von seiner Elterngeometrie (auch rekursiv, von deren Elterngeometrie usw.) oder
    - von der gml:Envelope (oder gml:Box), die der Wert der gml:boundedBy Property des Eltern-Features oder der Eltern-FeatureCollection ist (auch rekursiv).
- ein lokaler Wert des srsName Attribut geht dem geerbten Wert des srsName- Attributs vor
- sollte aus dem SRS nicht zweifelsfrei die Dimension erkennbar sein, muss das Attribut srsDimension bei Geometrieelementen (pos, posList) die Dimension des SRS enthalten

## 2 Definitionen

### 2.1 Geometrische Toleranzen

### 2.2 Planarität von Polygonen

Im CityGML müssen Polygone von einem planaren linearen Ring begrenzt sein . Aus praktischen Erwägungen werden auch Linear Ring Elemente, deren Punkte, die nicht exakt auf einer Ebene liegen sondern minimal davon abweichen, als planar akzeptiert. Intuitiv würde man fordern, dass ein Linearer Ring planar ist, wenn es eine Ebene E gibt, so dass die Distanz aller Punkte des Rings zu E einen gegebenen Grenzwert nicht überschreitet. Diese Ebene E kann über eine Ausgleichung aller Punkte aus R ermittelt werden, etwa durch Minimierung der maximalen Abstände von Punkten in R zu E (Ausgleichung gemäß Maximumsnorm) oder durch Minimierung der Quadrate der Abstände (Ausgleichung nach Gauß). Problematisch bei dieser Definition ist, dass auch minimale Knicke und Falten, wie in Abbildung 6 dargestellt, als planar akzeptiert werden. Dies entspricht aber in der Regel nicht dem intuitiven Verständnis einer planaren Fläche.

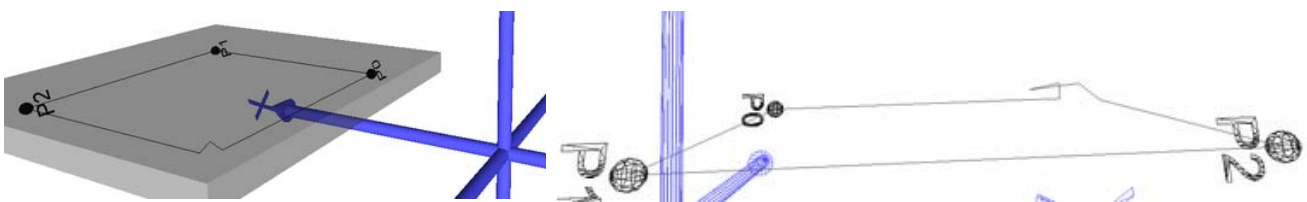


Abbildung 3: Minimale Knicke und Falten sollen auch bei nahezu planarenlinearen Ringen vermieden werden.

Wenn solche Knicke und Falten erkannt werden sollen, muss das Kriterium zur Toleranz bei planaren linearen Ringen ergänzt werden. Es ergibt sich eine alternative Definition der Planarität Linearer Ringe:

**Definition 1:** Ein Linearer Ring  $R$  heißt **planar**, wenn mindestens 3 Punkte des Rings nicht ko-linear sind und der Abstand der Punkte zu allen Ebenen  $E_{ijk}$ , die durch 3 nicht ko-lineare Punkte  $P_i$ ,  $P_j$  und  $P_k$  aufgespannt werden, kleiner ist als eine gegebene Schranke  $\epsilon$  :

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

**Anmerkung:** Für die Überprüfung der Planarität wäre es wünschenswert, eine Vorgabe für die Schranke  $\epsilon$  zu haben. Da die Eigenschaft der Planarität invariant gegenüber Skalierungen sein sollte, sollte in den Wert von  $\epsilon$  die Ausdehnung bzw. Größe (Flächeninhalt, max. Punktabstand) des Linearen Ringes eingehen. Dazu sollte die Punktgenauigkeit bekannt/festgelegt sein.

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

- Die Anzahl der Einträge in der Liste entspricht dem Produkt der Dimension des SRS und der Anzahl der direct positions.

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

- als `_Curves` sind nur `gml:LineStrings` erlaubt (gilt nur für CityGML)
- ein `gml:LineString` hat lineare Interpolation
- für die Kindelemente eines `gml:LineString` (Kontrollpunkte) gilt folgendes:
  - diese bilden eine Folge von "pos" (DirectPositionType) oder "pointProperty" (PointPropertyType) Elementen. "pos"-Elemente sind Kontrollpunkte, die nur zu dieser Curve gehören. "pointProperty"-Elemente enthalten Punkte, die von anderen Geometrien referenziert werden können, oder referenzieren andere Punkte (über XLinks), die in anderen Geometrien definiert sind.
  - Alternativ kann der `gml:LineString` aus einem "posList"-Element bestehen, das eine kompakte Möglichkeit der Angabe von Koordinaten bietet. In diesem Fall müssen alle Kontrollpunkte zu derselben Curve gehören und im selben SRS vorliegen. Die Anzahl der Punkte (Direct Position) ist mindestens zwei.
- Jeder Kontrollpunkt im `gml:LineString` erscheint nur einmal, außer dem ersten und letzten, die identisch sein können.
- Liniensegmente eines `gml:LineString` ergeben sich jeweils durch zwei aufeinanderfolgende Kontrollpunkte. Diese Liniensegmente dürfen sich nicht schneiden bzw. haben keine gemeinsamen Punkte. Ausgenommen sind Anfangs- und Endpunkt (definiert durch einen gemeinsamen Kontrollpunkt) zweier aufeinander folgender Liniensegmente.

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

Der Linear Ring ist das grundlegende Element zur Geometriebeschreibung in CityGML. Jedes einzelne Polygon einer Gebäudegeometrie wird durch seinen Umring definiert. Eben dieser Umring wird durch das Element „Linear Ring“ beschrieben.

Zur formalen Definition des Linear Ring muss zunächst der Begriff der Sequenz eingeführt werden. Eine Sequenz ist eine geordnete Liste von Elementen. Im Gegensatz zu einer Menge ist die Reihenfolge der Elemente in einer Sequenz von Bedeutung. Ebenso kann in einer Sequenz ein Element mehrfach vorkommen. Eine endliche Sequenz  $a$  mit  $n + 1$  Elementen wird durch die Elemente der Sequenz beschrieben:  $a = (a_0, a_1, \dots, a_n)$ . Die leere Sequenz  $a = ()$  hat keine Elemente.

Eine endliche Sequenz von Punkten  $R = (P_0, P_1, \dots, P_n), n \geq 3, P_i = (x_i, y_i, z_i)$  ist ein Linear Ring genau dann, wenn gilt:

- Der erste und der letzte Punkt der Sequenz sind identisch:  $P_0 = P_n$  (**closeness**)
- Mit Ausnahme des ersten und letzten Punktes sind alle Punkte verschieden, d.h.

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

(iii) Zwei Kanten  $(P_i, P_{i+1})$  und  $(P_k, P_{k+1})$  mit  $i = 0 \dots (n - 1), k = 0 \dots (n - 1), i \neq k$  dürfen sich nur in einem Start-/ Endpunkt berühren. Weitere Schnitt- bzw. Berührungspunkte sind nicht zulässig (**no self intersection**).

Sind alle Punkte der Sequenz ko-planar, wird der Linear Ring [planar](#) genannt.

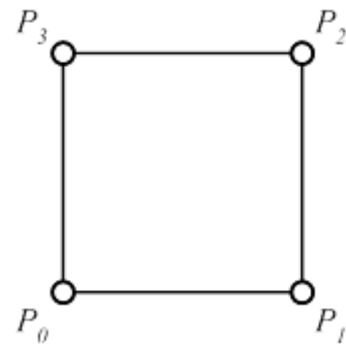
**Beispiel:**

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

mit

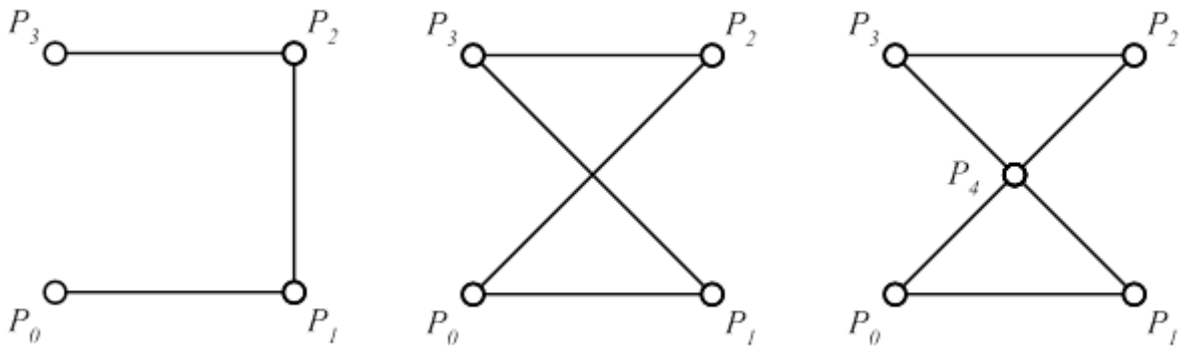
- $P_0 = (1, 1, 1)$
- $P_1 = (3, 1, 1)$
- $P_2 = (3, 3, 1)$
- $P_3 = (1, 3, 1)$

```
<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>
```



**Abbildung 4:** Planarer Linear Ring nach Definition, in GML und als graphische Darstellung

**Hinweis:** Die beiden linearen Ringe  $R_1 = (P_0, P_1, P_2, P_3, P_0)$  und  $R_2 = (P_0, P_3, P_2, P_1, P_0)$  sind nicht identisch.



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

Kein Linear Ring, da nicht geschlossen (siehe (i) closeness)

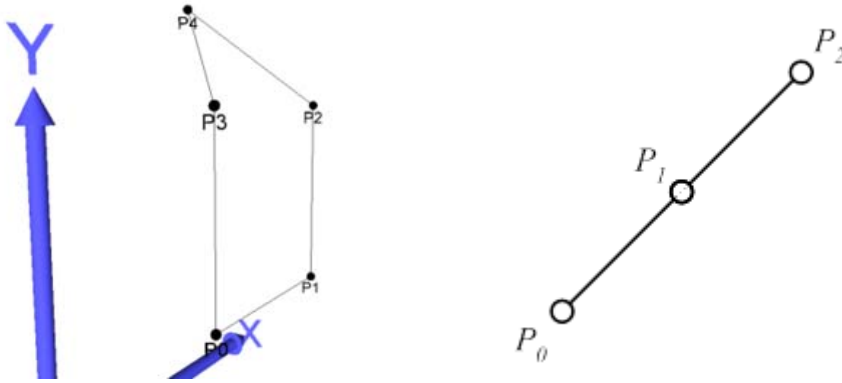
$$R = (P_0, P_1, P_3, P_2, P_0)$$

Kein Linear Ring, da sich zwei Kanten schneiden (siehe (iii) self-intersection)

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

Kein Linear Ring, da  $P_4$  in der Sequenz doppelt vorkommt (siehe (ii),  $P_4 = (2, 2, 1)$ )

**Abbildung 5:** Beispiele für Punktsequenzen, die keine Linear Ring Elemente beschreiben.



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

Linear Ring, der nicht planar ist  
 $P_4 = (2, 4, 0)$

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

Kein Linear Ring, da die 3 Punkte  
 ko-linear sind,  $P_1 = (2, 2, 1)$ ;  
 Kanten berühren sich im Innern  
 (siehe (iii))

**Abbildung 6:** Sonderfälle: nichtplanare und ko-lineare Linear Ring Elemente.



## 6 gml:Polygon

Ein planarer linearer Ring  $R_s$  definiert den Rand eines Polygons  $S$  (äußerer Ring).

- Ein Polygon ist durch genau einen solchen äußeren Ring und  $n \geq 0$  innere Ringe definiert. Jeder innere lineare Ring muss ebenfalls planar sein, und der äußere und alle inneren linearen Ringe müssen in derselben Ebene (im Rahmen einer gegebenen Toleranz) liegen.
- Jeder **innere** lineare Ring **muss innerhalb des Gebiets** der Ebene liegen, das der **äußere** Ring begrenzt.

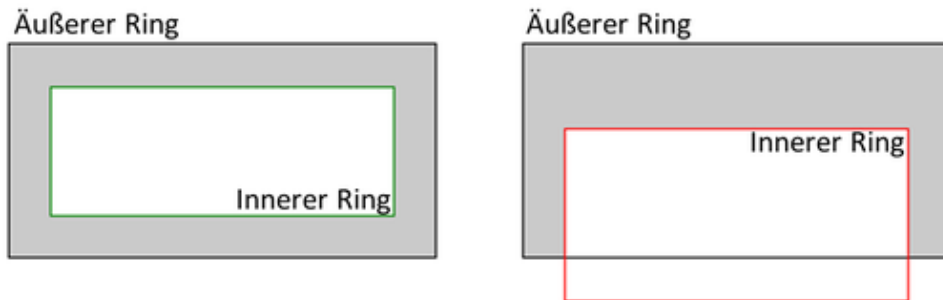


Abbildung 5: Polygon mit **korrektem inneren Ring** (links) und **nicht korrektem inneren Ring** (rechts)

- Die inneren linearen Ringe dürfen nicht verschachtelt sein, d.h. kein innerer Ring liegt in dem Gebiet der Ebene, das ein anderer innerer Ring definiert.

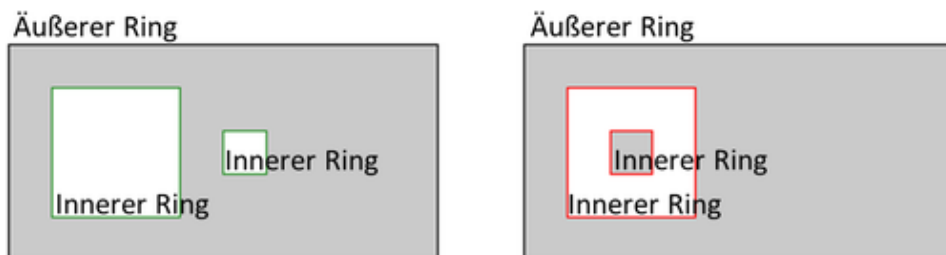


Abbildung 5: Polygon mit **zwei korrekten inneren Ringen** (links) und zwei verschachtelten, **nicht korrekten inneren Ringen** (rechts)

- Die inneren Ringe und der äußere Ring dürfen sich paarweise in **endlich !!!!! (evtl. nur einer)** vielen Punkten berühren. Dabei muss das Innere des Polygons zusammenhängend sein.



Abbildung 5: Polygon mit **korrektem, innerem Ring**, der den äußeren Ring an einem Punkt berührt (links) und **nicht korrektem innerem Ring**, der den äußeren Ring an zwei Punkten berührt und damit das Polygon spaltet (rechts)

- Die Reihenfolge der Punkte des äußeren Linear Rings definiert die **Orientierung** des Polygons. In  $R$  existieren mindestens drei nicht ko-lineare Punkte  $P_i$ ,  $P_j$  und  $P_k$ , die eine Ebene  $E(P_i, P_j, P_k)$  aufspannen. Der Vektor  $\vec{n}$ , der sich aus dem normierten Kreuzprodukt der beiden Vektoren  $\vec{P_iP_j}$  und  $\vec{P_jP_k}$  ergibt, wird als Flächennormale des Polygons bezeichnet:

$$\vec{n} = \frac{\vec{P_iP_j} \times \vec{P_jP_k}}{\|\vec{P_iP_j} \times \vec{P_jP_k}\|}$$

Anmerkung:

Aufgrund der Toleranz planarer linearer Ringe ist die Flächennormale des Polygons nicht eindeutig bestimmt.

Hinweis:

Die beiden von den linearen Ringen  $R_1 = (P_0, P_1, P_2, P_3, P_0)$  und  $R_2 = (P_0, P_3, P_2, P_1, P_0)$  definierten Polygone haben dieselbe geometrische Ausprägung, unterscheiden sich aber hinsichtlich der Orientierung.

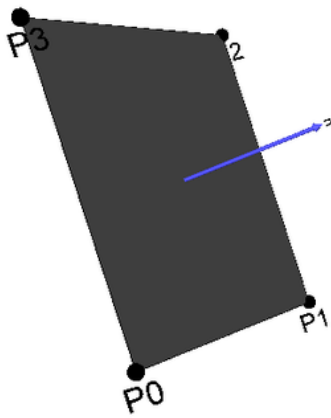


Abbildung 5: Polygon mit Flächennormale  $n$

## 7 [gml:OrientableSurface](#)

- Hat das Attribut "orientation" den Wert "+", so ist die OrientableSurface identisch zu der baseSurface. Ist der Wert der Orientierung "-", dann entspricht die OrientableSurface einer Surface mit einer Normalen, die der Normalen der baseSurface entgegen gesetzt gerichtet ist. "+" ist der Default-Wert des Attributs "orientation".
- Entweder referenziert das Element "baseSurface" die base surface durch ein XLink-Attribut, oder das Element "baseSurface" enthält die base surface als Kindelement. Die base surface hat eine positive Orientierung.

## 8 [gml:MultiSurface](#)

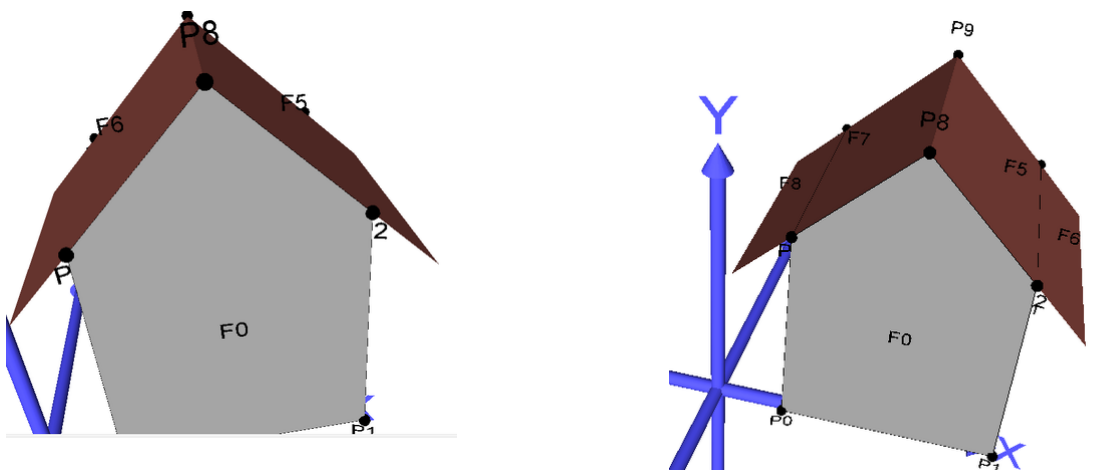
Eine Menge  $M = \{S_1, S_2, \dots, S_n\}$  von Polygonen wird als **MultiSurface** bezeichnet. Es gelten keinerlei Bedingungen für die Struktur der in  $S$  enthaltenen Polygone. Sie können sich beispielsweise überlappen oder durchdringen. Es ist auch nicht notwendig, dass die Polygone in  $S$  eine zusammenhängende Oberfläche beschreiben. Die Polygone können beliebig orientiert sein.

Das MultiSurface-Element ist daher sehr einfach zu nutzen.

Allerdings sind die häufigsten Fehler, die in einem Modell enthalten sind, auf die Verwendung von MultiSurface-Elemente zurückzuführen, da keinerlei strukturgebende Randbedingungen vorhanden sind.

In der Computergraphik wird eine solche Menge von Polygonen ohne explizite Struktur auch „Polygon-Suppe“ genannt.

Problematisch ist bei der Verwendung von MultiSurface Elementen, dass mit diesem Geometrielement Objekte dargestellt werden können, die keine Festkörper sind. Dies ist zwar ein sehr flexibler Ansatz, sollte aber nur verwendet werden, wenn explizit keine Festkörper modelliert werden sollen.



**Abbildung 6:** Modellierung von Dachüberständen als einfache Polygone. Im linken Modell wird das Gebäude mit einer MultiSurface-Geometrie beschrieben, die aus 7 Polygonen besteht. Eine Volumenberechnung ist nicht möglich, da es sich um keinen geschlossenen Körper handelt. Die rechte Abbildung zeigt dasselbe Modell, nun durch eine Solid (Gebäude) und eine zusätzliche MultiSurface-Geometrie (2 Polygone für Dachüberstand) beschrieben (vgl. CityGML Standard v1.0, S.61). Diese Variante ist zu empfehlen.

## 9 gml:CompositeSurface

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

1. 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:

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

2. 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  $P_{i+1}^k = P_j^l$ .

3. Die Polygone aus  $C$  sind so orientiert, dass die Flächennormale benachbarte Polygone in dieselbe Richtung zeigen.
4. 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).

# 10gml:Solid

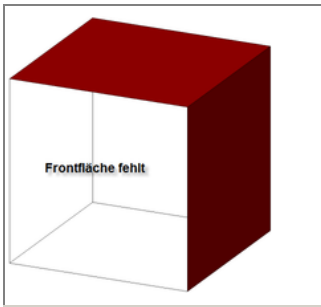
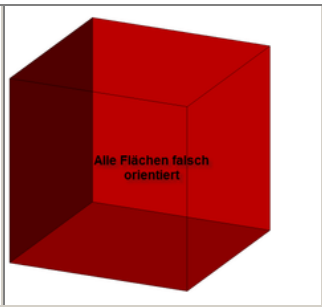
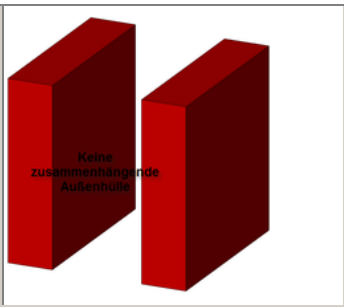
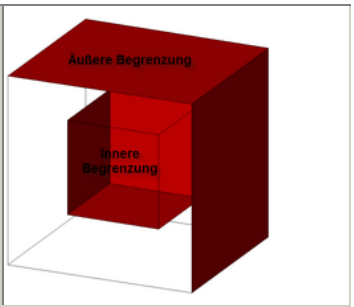
Ein Solid modelliert einen beliebigen Festkörper. Die Oberfläche des Solid wird über eine Menge von Polygonen beschrieben, die bestimmte strukturgebende Eigenschaften erfüllen müssen. Die Menge  $C = \{S_1, S_2, \dots, S_n\}$  von Polygonen beschreibt die Oberfläche eines Solid genau dann, wenn gilt:

- Die Schnittmenge zweier Polygone  $S_k$  und  $S_l$  aus  $C$  ist entweder leer oder besteht nur aus Punkten und 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:
 
$$S_i \cap S_k = \begin{cases} \emptyset \\ \{Q_0, Q_1, \dots, Q_m\}, Q_j = P_k^i \\ \{e_0, e_1, \dots, e_m\}, e_j = P_i^k P_{i+1}^k \end{cases}$$
- Jede Kante  $e_k = P_i^k P_{i+1}^k$  eines linearen Rings  $R_k = (P_0^k, P_1^k, \dots, P_n^k)$ , der ein Polygon  $S_k \in C$  definiert, wird genau einmal als Kante  $e_l = P_j^l P_{j+1}^l$  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  $P_{i+1}^k = P_j^l$ .
- Die Polygone aus  $C$  sind so orientiert, dass die Flächennormalen nicht ins Innere des Festkörpers zeigen, sondern nach außen.
- Die Polygone aus  $C$  sind zusammenhängend, d.h. in dem dualen Graphen von  $C$  gibt es einen Weg, der alle Knoten umfasst. Der duale Graph  $G_C = (V_C, E_C)$  von  $C$  besteht aus einer Menge  $V_C$  von Knoten und einer Menge  $E_C$  von Kanten. Jeder Knoten  $v$  aus  $V_C$  repräsentiert genau ein Polygon aus  $C$ . Eine Kante zweier Polygone  $S_k$  und  $S_l$  aus  $C$  wird in  $G_C$  durch eine Kante  $e = (v_{s_k}, v_{s_l})$  in  $E_C$  dargestellt.
- Für jeden Punkt  $P$ , der in einem linearen Ring eines Polygons aus  $C$  vorkommt, gilt: Der Graph  $G_P = (V_P, E_P)$ , der aus Polygonen und Kanten gebildet wird, die  $P$  berühren, ist zusammenhängend. Dabei repräsentiert jeder Knoten  $v$  aus  $V_P$  genau ein Polygon, dessen linearer Ring  $P$  enthält. Zwei Knoten sind genau dann mit einer Kante  $e$  aus  $E_P$  verbunden, wenn die Polygone, die durch die Knoten repräsentiert werden, eine gemeinsame Kante haben, die  $P$  berührt.

Aus (1) und (2) ergibt sich, dass die Oberfläche, die durch  $C$  beschrieben wird, keine Löcher enthalten darf. Mit den weiteren Bedingungen (4) und (5) ergibt sich, dass das Innere des durch  $C$  beschriebenen Festkörpers zusammenhängend sein muss.  $S$  wird auch als geschlossene **CompositeSurface** bezeichnet.

## Beispiele:

Korrekt: 6 Flächen, geschlossen, Flächenorientierung korrekt	Korrekt: 11 Flächen, geschlossen, Flächenorientierung korrekt	Korrekt: 10 Flächen, geschlossen, Flächenorientierung korrekt	Korrekt: 30 Flächen, geschlossen, Flächenorientierung korrekt

			
<p>Nicht Korrekt: 5 Flächen, nicht geschlossen, Flächenorientierung korrekt</p>	<p>Nicht Korrekt: 6 Flächen, geschlossen, Flächenorientierung nicht korrekt</p>	<p>Nicht Korrekt: 12 Flächen, nicht geschlossen (zwei Außenhüllen), Flächenorientierung korrekt</p>	<p>Nicht Korrekt: 12 Flächen, nicht geschlossen (innere und äußere Hülle), Flächenorientierung korrekt</p>

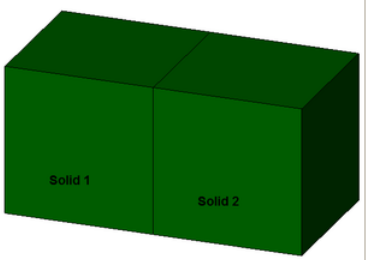
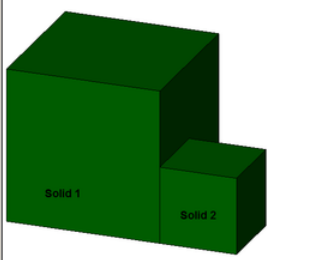
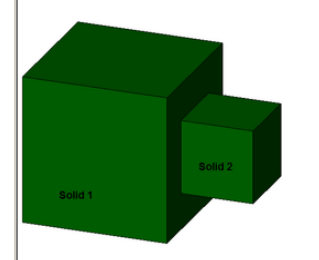
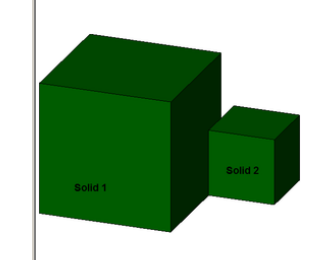
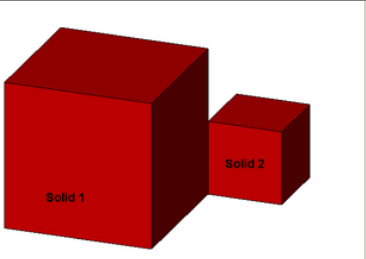
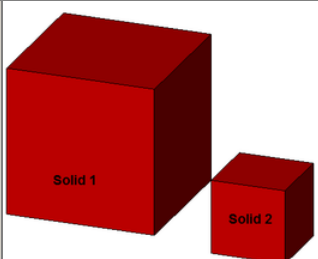
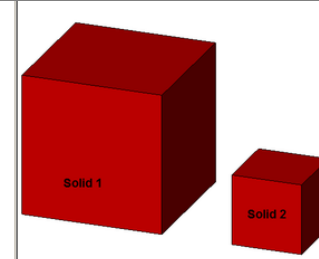
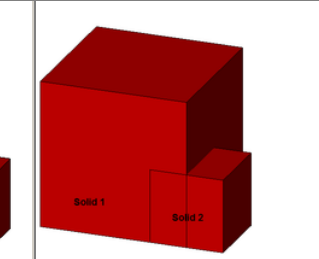
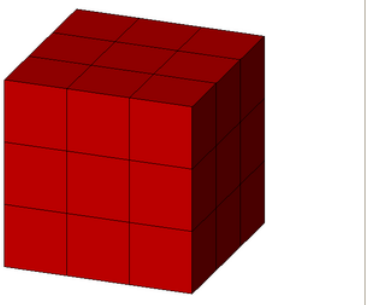
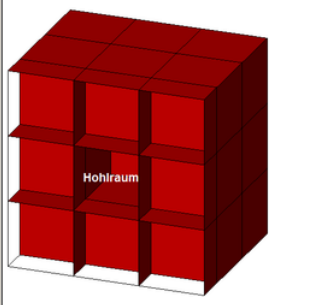
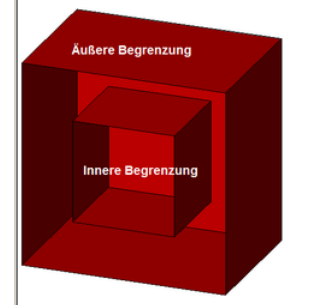
# 11 gml:CompositeSolid

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.

## Beispiele:

			
Korrekt; Flächenberührung	Korrekt; Flächenberührung	Korrekt; Flächenberührung	Korrekt; Flächenberührung
			
Falsch; Linienberührung	Falsch; Punktberührung	Falsch; keine Berührung	Falsch; Durchdringung
			
Falsch; 26 Quader mit Flächenberührung	aber mit Hohlräum	nicht korrekt; Innere Begrenzung	

## 12 [gml:Triangle](#)

- Ein Triangle ist ein Spezialfall eines Polygons, das von einem äußeren linearen Ring mit vier Punkten begrenzt wird.
- Es gibt keine inneren Ringe (kein Element "interior").

## 13 [gml:TriangulatedSurface](#)

Eine triangulated surface ist eine Composite Surface (vgl. Abschnitt 10), die nur aus triangles besteht, jedoch nicht geschlossen ist, d.h. sie darf Ränder besitzen. Bezüglich der Art und Weise, wie die Triangulation hergeleitet wurde, gibt es keine Einschränkung.

## 14 [gml:Tin](#)

Im Gegensatz zur `gml:TriangulatedSurface` sind in einem `gml:Tin` die triangles nicht explizit repräsentiert, sondern nur durch ihre Eckpunkte (genannt Kontrollpunkte). Zusätzlich können breaklines, stop lines und die maximale Länge einer Dreiecksseite in einem `gml:Tin` repräsentiert werden.

1. Ein TIN ist eine triangulierte Oberfläche, die Ergebnis der Anwendung eines Delaunay Algorithmus oder eines ähnlichen Verfahrens ist. Diese wird ergänzt um die Berücksichtigung von Stoplinien, Bruchkanten und der max. Länge von Dreiecksseiten. Diese Netze erfüllen das Delaunay Kriterium:
  1. Für jedes Dreieck im TIN enthält der Kreis, der durch die drei Punkte des Dreiecks definiert ist, keinen weiteren Eckpunkt eines anderen Dreiecks.
2. Stoplinien sind Linien, an denen die lokale Kontinuität und Regularität der Oberfläche zweifelhaft ist. In diesen pathologischen Gebieten sollen Dreiecke, die von Stoplinien geschnitten werden, entfernt werden. Dabei können Löcher in der Oberfläche entstehen. Werden Dreiecke am Rand der Oberfläche entfernt, ergibt sich eine Veränderung des Randes der Oberfläche.
3. Bruchkanten (Breaklines) sind Linien, an denen sich die Gestalt der Oberfläche signifikant ändert (Grate, Tallinien, ...). Solche Segmente sollen im TIN enthalten sein, auch wenn diese das Delaunay-Kriterium verletzen.
4. Max. Länge von Dreiecksseiten: Gebiete der Oberfläche, in denen die Datendichte für eine angemessene Ermittlung der Oberfläche nicht ausreichend ist, werden durch Angabe eines Kriteriums entfernt. Dieses Kriterium bezieht sich auf die Länge der Dreiecksseiten. Alle Dreiecke, deren Seiten (mind. eine) diese Länge überschreiten, werden aus der Oberfläche entfernt.
5. Es gibt mindestens drei Kontrollpunkte.
6. Die Oberfläche, die durch das TIN repräsentiert ist, hängt nicht von der Reihenfolge der Kontrollpunkte ab. Anwendungsschemata können Angaben zur Reihenfolge der Kontrollpunkte beinhalten, um die Rekonstruktion des TIN aus den Kontrollpunkten zu beschleunigen.