



Modeling Guide for 3D Objects

Part 1: Basics

Rules for Validating GML Geometries in CityGML

G. Gröger, V. Coors

Version of May 31, 2011

Document history:

Issue	Date	Author	Comment
1	2010-10-01	Coors	First draft
2	2010-10-14	Coors	Rules for solid type added
3	2010-10-23	Gröger	Merging with document by G. Gröger
4	2010-11-23	Coors	Editorial changes, rules for planarity of polygons added
5	2010-11-24	Gröger	Comments by K.-H. Häfele (CityGML forum) included
6	2010-12-15	Gröger	Comments by G. Juen included
7	2011-05-31	Coors	English version with minor editorial changes

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.

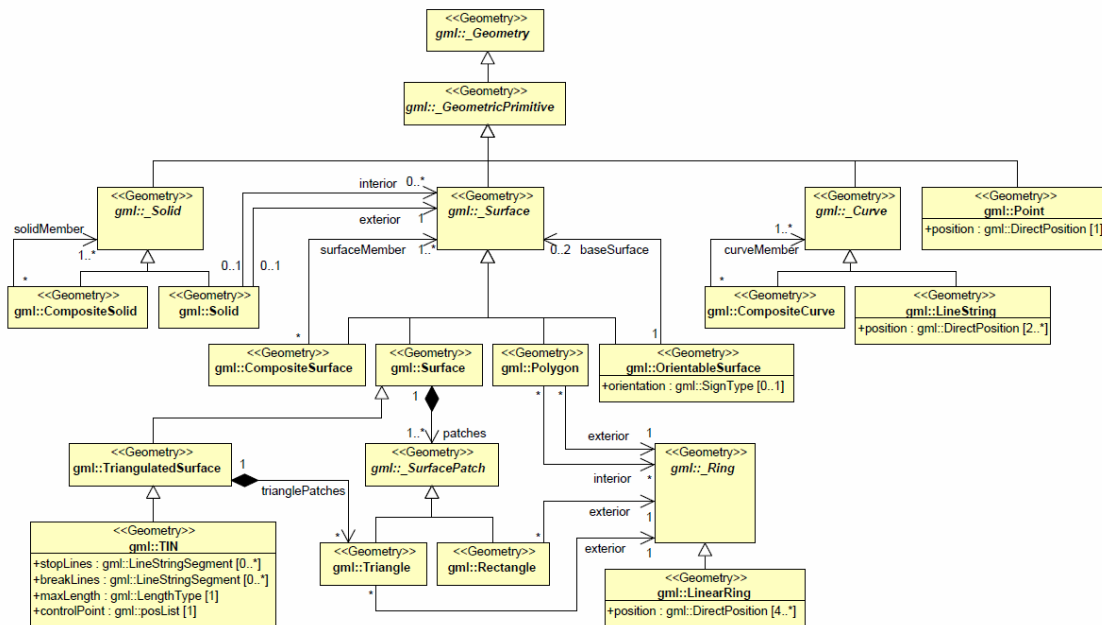


Figure 1: Profile of GML that is used in CityGML (Primitives and Composites)

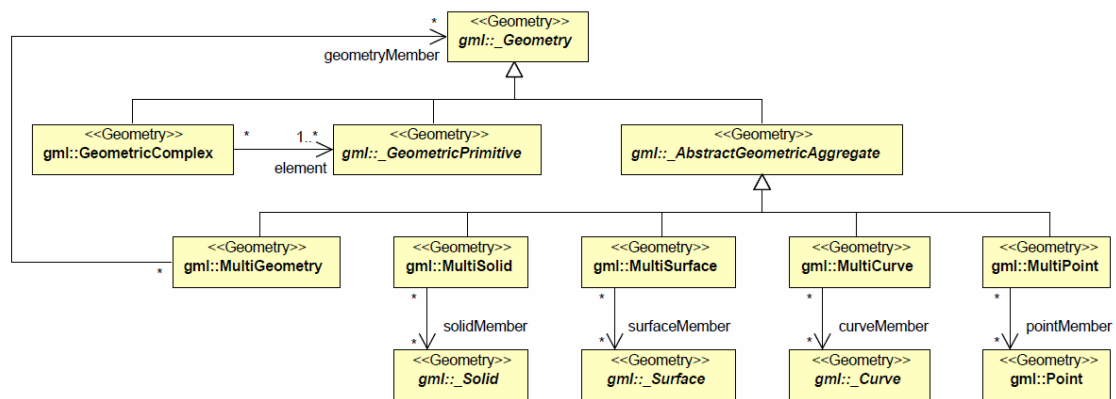


Figure 2: Profile of GML that is used in CityGML (Complexes and Aggregates)

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 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.

3 gml:posList

- The number of elements in a gml:posList equals the product of the dimension of the SRS and the number of direct positions.

4 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** iff

(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:

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

(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>
```

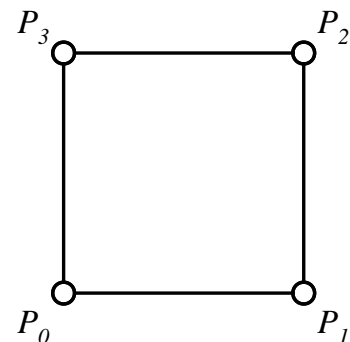


Figure 3: Planar Linear Ring: definition, GML and graphic representation

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.

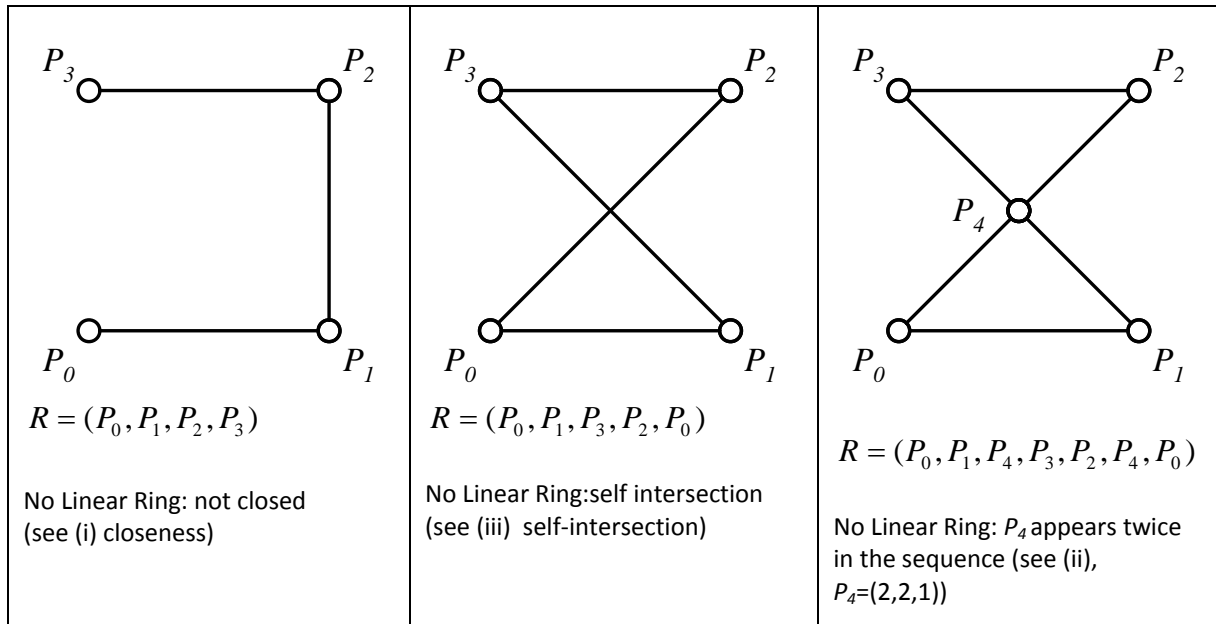


Figure 4: Examples of non-valid Linear Rings

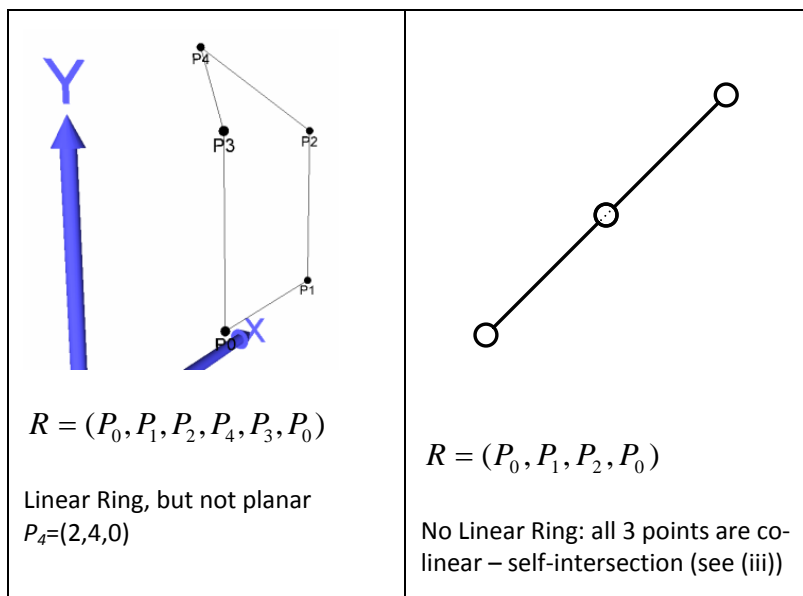


Figure 5: Special cases: non-planar and co-linear Linear Ring Elements.

4.1 Planarity of Linear Rings

„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.

Intuitively, a Linear Ring R would be defined as planar if there is a plane E with distance $d(E, P_i)$ less than a given threshold ε for all points P_i of R . However, this is true for Linear Rings with small bends and folds as shown in Figure 6 as well.

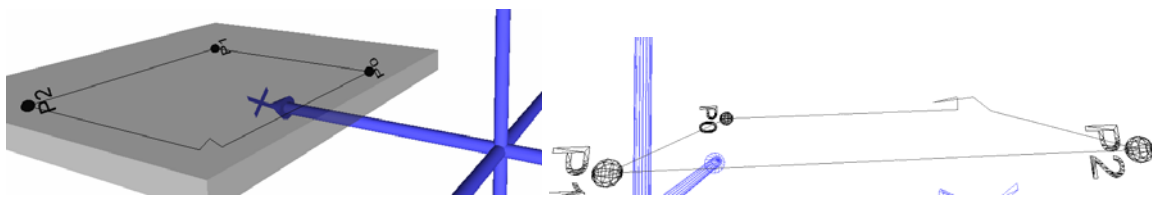


Figure 6: Bends and folds should not appear in planar Linear Rings

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 iff 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, P_k and is less than a given threshold ε :

$$\forall E_{ijk} \forall P_a \in R \quad dist(P_a, E_{ijk}) \leq \varepsilon$$

5 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 are 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. Interior rings must not overlap other Linear Rings or be included in another Linear Ring (except the exterior Linear Ring).
- Interior rings and the exterior ring may touch each other in a finite number of points. The inner area¹ of the polygon has to be connected.
- 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.

Note:

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

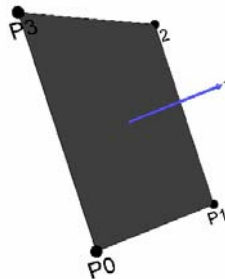


Figure 7: Polygon with normal vector n

¹ The inner area of a polygon is defined as the area of the polygon excluding its boundary (exterior and interior Linear Rings).

6 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.

7 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 iff:

- (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^2 .

² 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.

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**.

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 8. Volumetric parts should be modeled as solids (see gml:Solid)

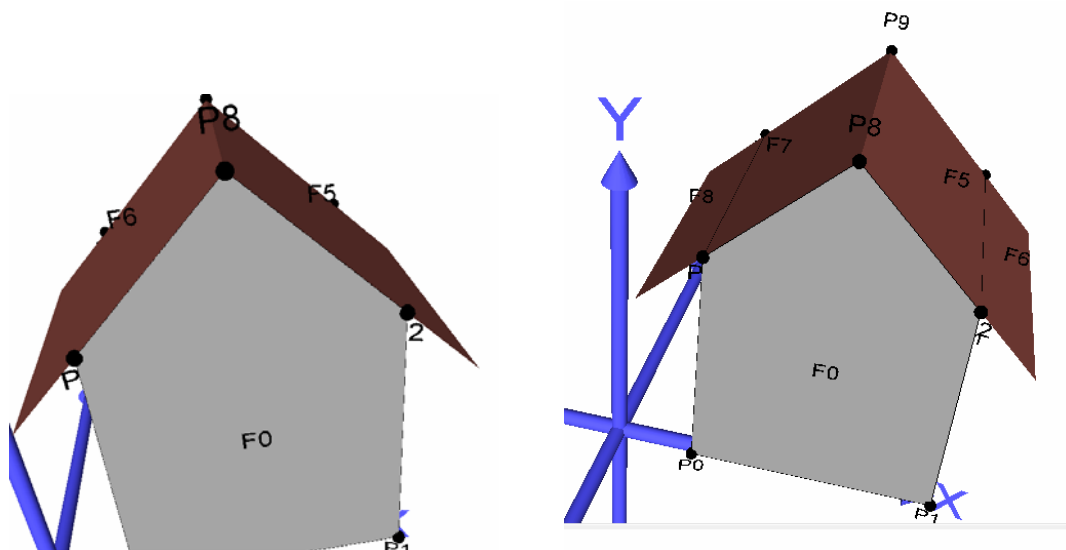


Figure 8: Modeling overlapping roof parts

Please note: the left model in Figure 8 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:Triangle

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

10 gml:TriangulatedSurface

A triangulated surface is a CompositeSurface (c.f. section 7) that consists of triangles only, but is not closed (it may have boundaries). There is no restriction on how the triangulation is derived.

11 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.

- 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**:

For each triangle in the network, the circle passing through its vertices does not contain, in its interior, the vertex of any other triangle.

- Stoptlines 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.
 - 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.
 - 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.
1. 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.