

Three-dimensional primitives in the context of the CityGML QIE

Hugo Ledoux
h.ledoux@tudelft.nl

October 5, 2014

1 What is an ISO 19107 solid?

ISO 19107 defines different geometric primitives¹: a 0D primitive is a GM.Point, a 1D a GM.Curve, a 2D a GM.Surface, and a 3D a GM.Solid. A primitive is built with lower-dimensional primitives, eg in Figure 1 the GM.Surface is composed of 2 (closed) GM.Curves, which are composed of several GM.Points. Observe that primitives do not need to be linear or planar, ie curves defined by mathematical functions are allowed.

The following three definitions from ISO (2003) are relevant:

Definition 1. A GM.Solid is the basis for 3-dimensional geometry. The extent of a solid is defined by the boundary surfaces. The boundaries of GM.Solids shall be represented as GM.SolidBoundary. [...] The GM.OrientableSurfaces that bound a solid shall be oriented outward.

Definition 2. A GM.Shell is used to represent a single connected component of a GM.SolidBoundary. It consists of a number of references to GM.OrientableSurfaces connected in a topological cycle (an object whose boundary is empty). [...] Like GM.Rings, GM.Shells are simple.

Definition 3. A GM.Object is simple if it has no interior point of self-intersection or self-tangency. In mathematical formalisms, this means that every point in the interior of the object must have a metric neighbourhood whose intersection with the object is isomorphic to an n -sphere, where n is the dimension of this GM.Object.

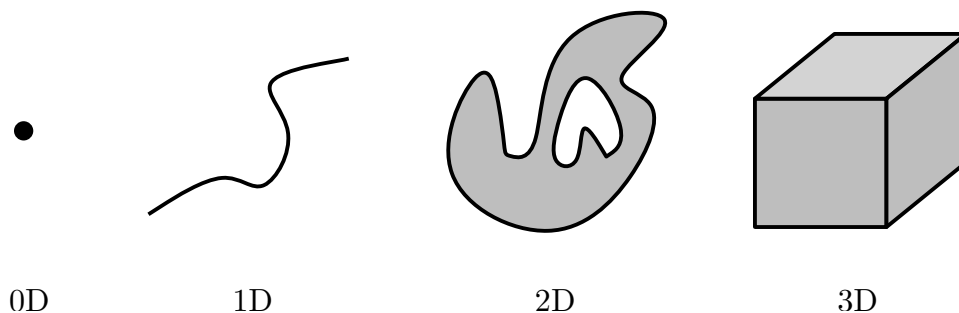


Figure 1: ISO 19107 primitives.

¹All the geometric primitive have the prefix 'GM.'

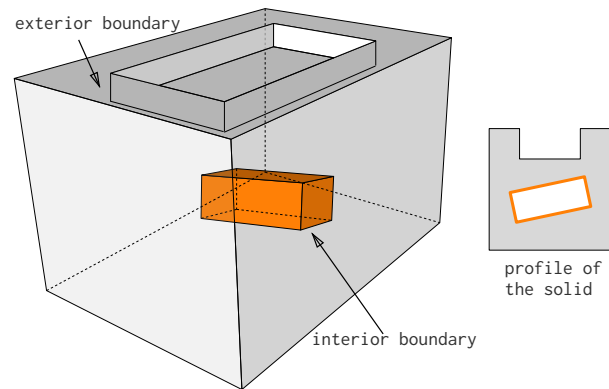


Figure 2: One solid which respects the international definition. It has one exterior shell and one interior shell (forming a cavity).

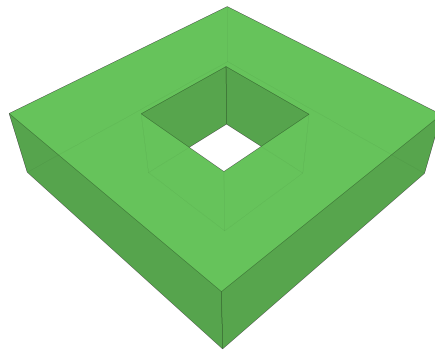


Figure 3: A 'squared torus' is modelled with one exterior boundary formed of ten surfaces. Notice that there are no interior boundary.

The bounding surfaces of a Shell thus form a *two-dimensional manifold* (or 2-manifold for short). It is a space that is topologically equivalent to \mathbb{R}^2 (the 2D Euclidean space). An obvious example is the surface of the Earth, for which near to every point the surrounding area is topologically equivalent to a plane. If we have a Shell stored in a data structure, it implies that each edge is guaranteed to have two incident faces, and that around each vertex the incident faces form one 'umbrella'.

Figure 2 shows a solid that respects that definition. First observe that the solid is composed of two shells (both forming its boundaries), one being the exterior and one being the interior shell. The exterior shell has eleven surfaces, and the interior one six. An interior shell creates a cavity in the solid—cavities are also referred to as “voids” or holes in a solid. A solid can have no inner shells, or several. Observe that a cavity is not the same as a hole in a torus (a donut) such as that in Figure 3: it can be represented with one exterior shell having a genus of 1 and no interior shell.

2 Primitives in CityGML

CityGML uses the ISO 19107 geometric primitives for representing the geometry of its objects. However, as shown in Figure 4 only a subset is used, with the following 2 restrictions: (1) GM.Curves can only be *linear*, thus LineStrings and LinearRings are used; (2) GM.Surfaces

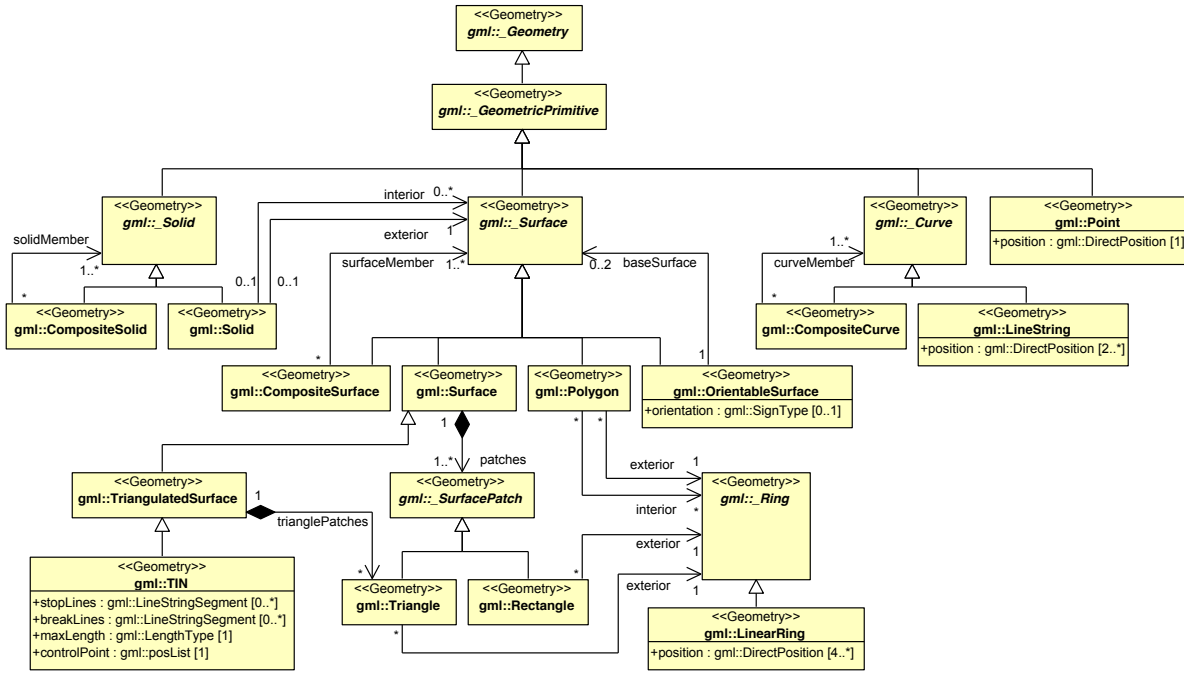


Figure 4: UML diagram of the CityGML geometry model.

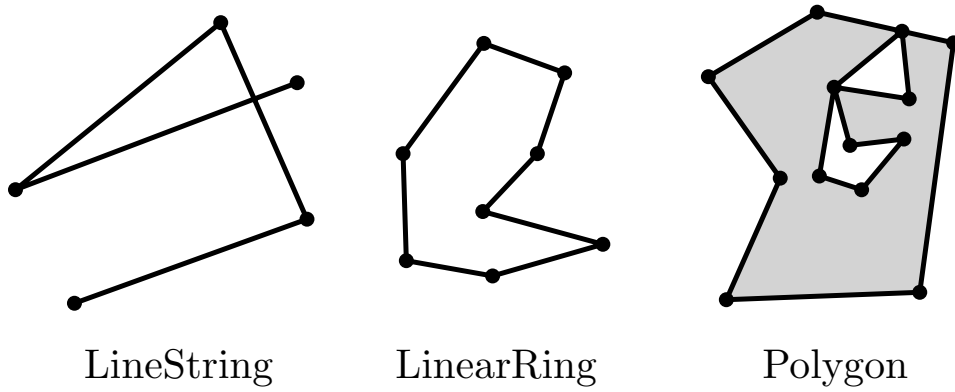


Figure 5: 2D CityGML primitives.

can only be *planar*, thus Polygons are used. The primitives for constructing Shells and Solids are shown in Figure 5.

Definition 4. A *LineString* is a *Curve* with linear interpolation between *Points*. Each consecutive pair of *Points* defines a *Line* segment. A *LinearRing* is a *LineString* that is both closed and simple.

Definition 5. A *Polygon* is a surface patch that is defined by a set of boundary curves and an underlying surface to which these curves adhere. The default is that the curves are coplanar and the polygon uses planar interpolation in its interior.

Each shell of a solid is thus composed of Polygons, and these can have inner rings (which are often referred to as holes). Observe that the top polygon of the solid in Figure 2 has one inner ring, but that other polygons “fill” that hole so that the exterior shell is “watertight”.

1. Polygons are topologically closed;
2. The boundary of a Polygon consists of a set of LinearRings that make up its exterior and interior boundaries;
3. No two Rings in the boundary cross and the Rings in the boundary of a Polygon may intersect at a Point but only as a tangent, eg

$$\forall P \in Polygon, \forall c1, c2 \in P.Boundary(), c1 \neq c2,$$

$$\forall p, q \in Point, p, q \in c1, p \neq q, [p \in c2 \Rightarrow q \notin c2];$$

4. A Polygon may not have cut lines, spikes or punctures eg:

$$\forall P \in Polygon, P = P.Interior.Closure;$$

5. The interior of every Polygon is a connected point set;
6. The exterior of a Polygon with 1 or more holes is not connected. Each hole defines a connected component of the exterior.

Figure 6: The six assertions for the validity of a polygon (OGC, 2006, pages 26–27).

3 QIE = no cavaties

It should be noticed that during the QIE, only buildings in LOD1, LOD2 and LOD3 will be considered, and, as a consequence, cavities in solids are ignored.

This imply that a solid has exactly *one* shell representing its exterior boundary. However, this does *not* mean that inner rings are also excluded, because simple LOD1 buildings having an inner yard for instance (as in Figure 3) require inner rings (or the faces should be decomposed).

4 Requirements for validity of the 3D primitives

Each primitive used to construct a higher-dimensional primitive should be valid. For rings and polygons, observe that these will be embedded in 3D (ie the points used to construct rings will have (x, y, z) coordinates). This implies that they have to be projected to a plane so that a two-dimensional coordinate system is used.

4.1 Rings & Polygons

According to the ISO 19107 abstract specifications, the different boundaries of a polygon are allowed to interact with each other, but only under certain circumstances. The implementation specifications defined by the OGC (OGC, 2006) give clear requirements, see the assertions in Figure 6.

Some concrete examples of invalid polygons are shown in Figure 7, and here are a few explanations:

1. Each ring should be closed (p_{11}): its first and its last points should be the same.

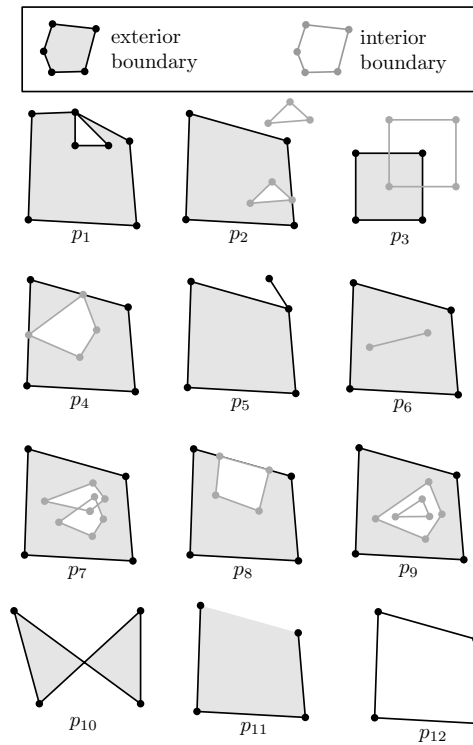


Figure 7: Some examples of invalid polygons. Polygon p_{12} has its exterior and interior rings defined by the same geometry.

2. Each ring defining the exterior and interior boundaries should be *simple*, ie non-self-intersecting (p_1 and p_{10}). Notice that this prevents the existence of rings with zero-area (p_6), and of rings having two consecutive points at the same location. It should be observed that the polygon p_1 is not allowed (in a valid representation of the polygon, the triangle should be represented as an interior boundary touching the exterior boundary).
3. The rings of a polygon should not cross (p_3 , p_7 , p_8 and p_{12}) but may intersect at one tangent point (the interior ring of p_2 is a valid case, although p_2 as a whole is not since the other interior ring is located *outside* the interior one).
4. A polygon may not have cut lines, spikes or punctures (p_5 or p_6); removing these is known as the *regularisation* of a polygon (a standard point-set topology operation).
5. The interior of every polygon is a connected point set (p_4).
6. Each interior ring creates a new area that is disconnected from the exterior. Thus, an interior ring cannot be located outside the exterior ring (p_2) or inside other interior rings (p_9).

4.2 Planarity requirement

A polygon must be planar, ie all its points (for both the exterior and interior rings) must lie on a plane. Interestingly, the concept of *tolerance* is not mentioned in the standards ISO and OGC.

Two requirements are proposed:

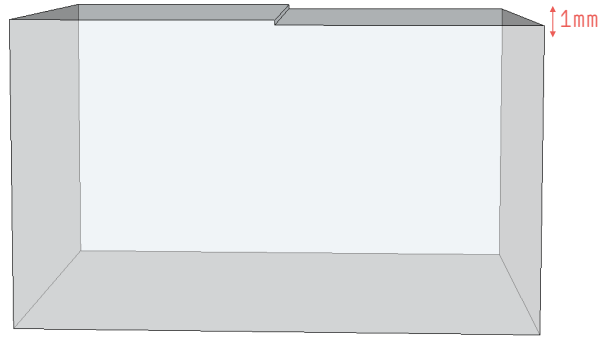


Figure 8: All the points of the top polygon are within 1mm but the polygon cannot be considered planar.

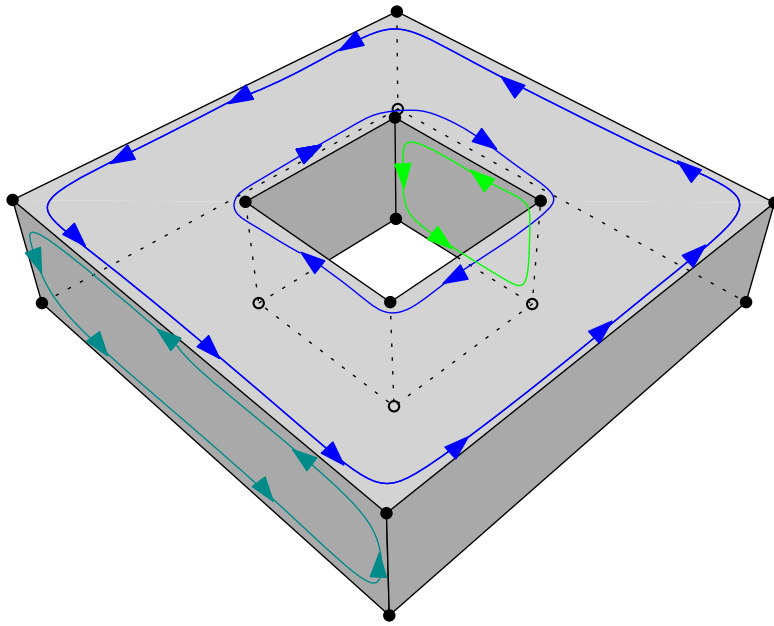


Figure 9: One solid and the orientation of 3 of its polygons (different colours).

1. the distance between every point forming a polygon and a plane is less than ϵ_1 , a given tolerance (eg 1mm).
2. based on a triangulation of the polygon (any triangulation), the orientation of the normal of each triangle must not deviate more than ϵ_2 (eg 1 degree).

The second requirement is to ensure that cases such as that in Figure 8 are detected.

4.3 Orientation requirement

For a polygon embedded in the 2D plane, the orientation of its exterior ring must be the opposite of that of its interior rings (clockwise versus counterclockwise).

If one polygon is used to construct a shell, it must be oriented in such a way that when viewed from outside the shell its points are ordered counterclockwise. Figure 9 shows an example. In other words, the normal of the surface points outwards if a right-hand system is used, ie when the ordering of points follows the direction of rotation of the curled fingers of the right hand,

then the thumb points towards the outside. If the polygon has interior rings, then these have to be ordered clockwise.

If the polygon is part of a MultiSurface, then there is no prescribed orientation besides the fact that the outer ring must have a different orientation than the inner ring(s).

4.4 Requirements for shells and solids

To understand the requirements for shells and solids, we can simply generalise the assertions in Figure 6: polygons become solids, rings become shells, and holes become cavities.

The first assertion means that a solid must be closed, or ‘watertight’. The solid s_1 is thus not valid but s_2 is since the hole in the top surface is ‘filled’ with other faces.

The second assertion implies that each shell must be simple, ie that it is a 2-manifold.

The fourth assertion states that a shell is a 2-manifold and that no dangling pieces can exist (such as that of s_6); it is equivalent to the *regularisation* of a point-set in \mathbb{R}^3 .

The other assertions refer to solids having interior shells, which are out of scope for the QIE, and thus ignored.

References

ISO (2003). ISO 19107:2003: Geographic information—Spatial schema. International Organization for Standardization.

OGC (2006). OpenGIS implementation specification for geographic information—simple feature access. Open Geospatial Consortium inc. Document 06-103r3.