3D modelling requirements for IMGeo

Requirement	Brief Description	Validation
3.1 IMGeo 2.1.1 CityGML: Generic Requirements		
Requirement 1	The 3D data should be structured according to IMGeo-CityGML format.	Check this by using the developed validation tool schema
Requirement 2	The IMGeo-CityGML data must comply with CityGML 2.0. In some cases we have more stringent requirements than CityGML	Check this by using the developed validation tool schema
Requirement 3	Employ the EPSG 7415 Spatial Reference System (coordinate system).	Check if the EPSG code 7415 is to be found in the CityGML file. schema
3.2 Specifications for LOD0 Representation.		
Requirement 4	Every object in IMGeo is represented by a LOD0 geometry i.e. a TIN surface (triangulatedSurface) per object (tessellation of the object's footprint) . The LOD0 terrain is formed by a collection of such adjacent TIN surfaces, with recognizable object boundaries (constrained TIN)	Check if the number of polygons in LOD0 is the same as that in 2D IMGeo CityGML
Requirement 5	The LOD0 geometries of all IMGeo polygons (water, road, building, land use, vegetation) at ground level should form a planar partition in 2.5D (no holes or overlap).	Check by looking for holes or overlap
Requirement 6	The height difference between the terrain in reality and its representation in TINs is allowed to be maximum X cm. X can be dependent on the object type (for example another X can be chosen for hard surfaces with curbs than that for pasture). Individual apexes are acceptable until up to 3 times X, but connected pieces of a TIN of more than Y m2 may deviate no more than this X cm.	It can be requested that a colour coded point file be supplied in which the terrain points are coloured as a function of the height deviation with respect to the object surface that the terrain models. It can then easily be seen if areas (greater than Y m2) show greater deviation.
Requirement 7	Vertical surfaces in the TIN may not occur, because many GIS software crashes on such data. Instead, vertical surfaces should be approached by maximum sloping surfaces. How this should be done depends on which objects are left and right of the vertical jump. The sloping surfaces need to be attached as follows to the relevant object: • Boundary of (Auxiliary) Traffic Area - Terrain Area, to the (Auxiliary)Traffic Area • Boundary of (Auxiliary) Traffic Area - Terrain Area to (Auxiliary)Traffic Area • Boundary (Auxiliary) Traffic Area – (Auxiliary) Traffic Area, to the highest object • Boundary Terrain Area- Terrain Area, to the highest object • Boundary of (Auxiliary) Traffic Area/ Water Body/ Terrain Area/ Division – BuildingPart, to the BuildingPart • Boundary of (Auxiliary) Traffic Area/Water Body/ Terrain Area/ Division – Other Construction, to the construction • Boundary of (Auxiliary) Traffic Area/ Water Body/ Terrain Area/ Division – Construction, to the construction • Boundary of (Auxiliary) Traffic Area/ Water Body/ Terrain Area – Division, to the Division • Boundary of Object – Waterbody, to the object	Testing the Z component of the TIN triangles' normal vectors. These Z components may not be equal to 0. An alternative, but incomplete check, is to look for points with the same XY co ordinates, but the same Z co ordinates.
Requirement 8	When very precise vertical intervals between specific objects are necessary, this should be recorded in the technical specifications. A minimum height should be defined and vertical intervals must be visible. Examples are the height jumps at the location of curbs.	Check randomly if small vertical intervals have been modelled.

Requirement 9	Waterbodies are always flat, horizontal surfaces.	Testing the X and Y component of surfaces' normal vectors. These must be equal to 0.
Requirement 10	IMGeo polygons which are above or below the terrain should be modelled with a triangulatedSurface **which connects up to the topologically consistent ground level**. The result is the stacking of 2.5 objects.	Overlapping objects with differing levels may not intersect each other in height.
Requirement 11	All IMGeo polygons should be assigned to the IMGeo LOD0 representation, i.e. both those at ground level as well as the ones above and below ground level	Check if a number of polygons in LOD0 are in agreement with the polygons in 2D IMGeo CityGML.
Requirement 12	Terrain Intersection Curves (TIC's) should be used in order to make ClosingSurfaces where 3D objects hang above or in the terrain model. This results in a closed topologically correct terrain model.	
3.3 Building Specifications		
Requirement 13	The ground surface of a building at LOD1 and LOD2 must be horizontal. The ground surfaces should, though, be determined per individual building and not per block of buildings . This surface is then positioned at the lowest height of the terrain at the location of this surface so that the building sinks "in" the terrain and gaps between ground surface are avoided.	if semantics are present it's easy. If not then it's tougher
Requirement 14	Notwithstanding the CityGML specification, LOD0 footprint must be determined where the outside wall touches the terrain.	
Requirement 15	An LOD1 representation should be supplied for every IMGeo building. **And other constructions?**	Easy to check if the building's IMGeo ID is saved as an attribute to the LOD1 representation.
Requirement 16	The building height of LoD1 is **the median** of the height of the points which are positioned within the footprint.	Check randomly if the median of the height of the points on one roof lies within a margin of X cm from the height in the model.
Requirement 17	If a building's roof has significant vertical intervals (for example a church with a tower), then these differing height levels should be distinguished in 3D, particularly if the interval is greater than, for example, 1.5 metres and if the surface area is greater than 4 square metres **LoD1.3**	
Requirement 18	The lower surfaces of the building's block geometry must correspond to the 2D and LOD0 geometry in IMGeo.	
Requirement 19	The lower surface of a LOD1 block should be horizontal, taking the lowest point of the footprint's terrain triangulation as its height (see LOD0 building)	
Requirement 20	For buildings which bridge roads or water, through passage should be guaranteed. This may be artificially applied.	
Requirement 21	The geometry of LOD1 Buildings should be defined in CityGML as GML:Solids (closed volumes, also from below) and not as GML:MultiSurface, which is permitted for LOD1 buildings..	Each building object consists of exactly one solid. CompositeSolid allow?
Requirement 22	Each LOD2 IMGeo building is modeled by the GML:Solid geometry type in which the semantics of the boundaries (surfaces) are made explicit (e.g. footprint, roof surface, wall surface). LOD2 buildings can be represented as a collection of a solid with other geometry types such as a multisurface for a roof overhang	Each building object consists of a minimum of one solid.
Requirement 23	The locations of the outer walls of 3D building models should be in alignment with the 2D boundaries from the BGT and the BAG (preferably the BGT).	Randomly test if boundaries from the BGT or BAG have been taken up.
Requirement 24	Roof boundaries of 3D building models are in agreement with 2D boundaries from the BGT or BAG (preferably the BAG).	

external code

code. – in theory schema would but in json one and more generic.

problem here is if Building has Parts, then does that still apply.

same

Requirement 25	Building models should be complete in the sense that the combination of all of a building's surfaces collectively forms a closed volume, a 3D solid. No surface from another building may be positioned within a building model. **Building models may touch each other, but not overlap.**	Check by means of the developed validation tool.
Requirement 26	When a roof overhang is explicitly modelled, roof surfaces should be split at the roof overhang's location in order to result in a solid geometry. These roof overhangs should be modeled as a (multi)surface and the rest of the roof should form a part of the solid geometry's boundary.	
Requirement 27	When a roof overhang is explicitly modeled, roof surfaces should be split at the roof overhang's location to obtain a valid solid geometry. These roof overhangs should be modeled as a (multi)surface and the rest of the roof should form a part of the solid geometry's boundary	The supplier can be asked to supply a colour coded point cloud in which the points colour within BAG/BGT polygons a function is of the height difference with the modelled roof. Larger deviations can then be spotted easily. The surface area of each "connected component" of points which deviate too much can be calculated with a little more effort.
Requirement 28	LOD2 roof surfaces with a minimum surface area of X m2 may not deviate more than Y m in height from the corresponding points from the point cloud	Checking can be done with a colour coded file, just as by the previous Requirement, although this time with a colour dependent on the angle difference between normal vectors which have been estimated from points which lie within a certain radius and normal vectors from the modelled surfaces.
Requirement 28	Roof surfaces with a minimum surface area of X m2 may not deviate more than Y degrees in the normal direction from a surface because of the corresponding points from the point cloud. This prevents very flat saddle roofs to be modeled by flat roofs and mansard roofs to be modeled by saddle roofs	Check in the same way as Requirement 31.
Requirement 29	Curved surface areas should be represented by a triangulation in which deviation between the true surface area and the triangulation is not more than Xm.	
Requirement 30	Roof surface corner points in the model (for as much as they haven't been misaligned by the BAG) must lie within a distance of Xm from the closest neighbouring data points	Check if there are data points present within a radius of X m from a vertex (and within a BAG outline). Use a 3D query option or specialised software.
Requirement 31	The solids of buildings in LOD1 and LOD2 should conform to the requirements which are discussed in 4.3.4.	Check by means of the developed validation tool.