Comments on candidate InfraGML encoding standards

Submitted by: 3D Geoinformation Group, Delft University of Technology, the Netherlands.

Comments

1. There is significant overlap with CityGML in modelling of urban objects. For instance, there are several InfraGML requirement classes analogous to CityGML modules (see table 1 below). The question is why to model a separate schema when CityGML has an extension mechanism to include application specific classes/attributes as an ADE? This will eliminate the need to later integrate with CityGML.

<table>
<thead>
<tr>
<th>InfraGML classes</th>
<th>CityGML modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>LandSurface</td>
<td>Relief</td>
</tr>
<tr>
<td>LandElement</td>
<td>Water/Vegetation</td>
</tr>
<tr>
<td>Roads</td>
<td>Transportation (Rods)</td>
</tr>
<tr>
<td>Railways</td>
<td>Transportation (Railways)</td>
</tr>
<tr>
<td>LandDivision</td>
<td>LandUse</td>
</tr>
<tr>
<td>Facilities</td>
<td>CityObject/Site</td>
</tr>
<tr>
<td>Project</td>
<td>CityModel</td>
</tr>
<tr>
<td>FacilityPart</td>
<td>Bridges/Buildings/Tunnels/Railway/Road</td>
</tr>
</tbody>
</table>

2. InfraGML v0.5.0 presented Terrain as one of the main subject areas of InfraGML, but now it is modelled as LandSurface in v1.0.0. No justification is given for this change. LandSurface can be anything, either the ground or
the subsurfaces. Terrain should be a separate module to mark this distinction. There is no separate modelling of terrains as DTM and DSM.

3. In section 7.9.3 of LandInfra Conceptual model document, a TIN is used for the spatial representation of LandSurface (terrain). The TIN is based on GML 3.3 SimpleTriangle. Section 7.9.8.1 defines a Simple Triangle as a “surface defined by three points (three Points which, when connected, form a clockwise triangle; the first point is repeated as the last, closing point)”. By definition, it is a Linear Ring. So, apart from removing one last vertex of the ring, what’s the advantage of this representation over CityGML Relief representation which uses gml:Triangle?

4. In Section 7.9.8.2, gmltin:elementGeometry is the element for specifying the geometry of the gmltin:TINElement. The allowable geometry types for TIN are based on the values of gmltin:TINElementType. There is a parameter value called “user-defined TINElement, the choice of geometry is user-defined”. Do you have any working example to elaborate on this? The user can specify TIN geometry using different TIN representation data structures like for instance tristrips or indexed triangles. How this is going to integrate with the CityGML TINs? There is a requirement for a codelist of allowed TIN data structures. Similar is the case for the TINElementType in Section 7.9.8.3.

5. There is significant reuse of GML documentation. For instance, in section 7.9.8.3 (Fig. 1), the explanation of TIN Element type is the same as in the GML 3.3 documentation (Fig. 2).

6. As InfraGML supports all the TINElement Types listed in Section 7.9.8.3, the sample XML data file for LandFeature class should be extended to give an idea of the data representation with all these types.

7. In the sample XML file of LandFeature class, xlink is used to point to the geometry of LandElement. What exactly is the spatial representation type of this element? Can it be any type e.g. point, polygon, etc.? Is it a SimplePolygon of GML 3.3 or a gml:polygon?
A TINElement of type RandomPoints represents points on the surface of known elevation from which triangles can be generated or which represent linestring points contained in other TINElement geometries.

A TINElement of type GroupSpot represents a collection of related points on the surface of known elevation from which triangles can be generated or which represent linestring points contained in other TINElement geometries.

A TINElement of type Boundary is used to specify the boundary of the TIN surface. As a constraint applied to a TIN surface after the initial Delaunay triangulation, application of the boundary causes the surface to be clipped to the Boundary's Polygon geometry value. This may result in the elimination or addition of points contained in the initial random points or group spots TINElements. This may also result in localized re-triangulation at the boundary. Support for interior boundaries is implementation specific.

A TINElement of type Breakline is used to represent a local ridge or depression in the TIN surface. When a breakline is specified for a TIN surface, simple triangle patches must be adjusted so that no triangle is crossed by the breakline. Part or all of the breakline becomes an edge of two or more triangles. The elevation along the breakline takes precedence over the elevation of the original triangulated surface for the entire length of the breakline.

A TINElement of type SoftBreak behaves as a ‘breakline’ (see above) except that contour lines generated for the surface can be smoothed where they cross soft breaks.

A TINElement of type ControlContour behaves as a ‘breakline’ (see above). The z coordinate values are identical for all points in the ‘control contour’ LineString geometry. Triangles within the vicinity of a control contour need to be assessed during re-triangulation to ensure that they are not zero slope triangles (all three vertices fall on the same control contour).

Voids enclose a voided area of the TIN surface. They can be represented by either break, drape or (regular) void types of TINElement. Void geometry is of type Polygon, which might include interior boundaries. Triangles within a void still exist in the TIN surface but are considered to be void. Triangles inside of the Polygon interior boundary are in the exterior of the Polygon and therefore not in the void: these triangles are therefore not considered to be void.

For a void TINElement of type BreakVoid the boundary LineStrings of the 3D Polygon behave as breaklines in that triangles in the simple triangle patch collection must be adjusted so that no triangle is crossed by the break void boundary. Part or all of the break void boundary becomes an edge of two or more triangles. The elevation of this break void boundary takes precedence over the elevation of the original triangulated surface for the entire length of the boundary.

For a void TINElement of type DrapeVoid the boundary LineStrings of the 2D or 3D Polygon behave as breaklines in that triangles in the simple triangle patch collection must be adjusted so that no triangle is crossed by the drape void boundary. Part or all of the drape void boundary becomes an edge of two or more triangles. However, for drape voids, the elevation of the original triangulated surface takes precedence over the elevation of the drape void boundary.

Fig. 1: Snippet from Section 7.9.8.3 of LandInfra Conceptual Model
A TIN element of type 'random points' (gml:tin:randomPoints) represents points on the surface of known elevation from which triangles can be generated or which represent linestring points contained in other TIN element geometries.

A TIN element of type 'group spot' (gml:tin:groupSpot) represents a collection of related points on the surface of known elevation from which triangles can be generated or which represent linestring points contained in other TIN element geometries.

A TIN element of type 'boundary' (gml:tin:boundary) is used to specify the boundary of the TIN surface. As a constraint applied to a TIN surface after the initial Delaunay triangulation, it causes the surface to be clipped to the boundary TIN element's gml:Polygon geometry value. This may result in the elimination or addition of points contained in the initial random points or group spots TIN elements. It may also result in localized re-triangulation at the boundary. It is implementation-defined whether interior boundaries are supported.

A TIN element of type 'breakline' (gml:tin:breakline) is used to represent a local ridge or depression in the TIN surface. When a breakline is specified for a TIN surface, simple triangle patches must be adjusted so that no triangle is crossed by the breakline. Part or all of the breakline becomes an edge of two or more triangles. The elevation along the breakline takes precedence over the elevation of the original triangulated surface for the entire length of the breakline.

A TIN element of type 'soft break' (gml:tin:softBreak) behaves as a 'breakline' (see above) except that contour lines generated for the surface can be smoothed where they cross soft breaks.

A TIN element of type 'control contour' (gml:tin:controlContour) behaves as a 'breakline' (see above). The z coordinate values must be identical for all points in the 'control contour' gml:LineString geometry. Triangles within the vicinity of a control contour need to be assessed during re-triangulation to insure that they are not zero slope triangles (all three vertices fell on the same control contour).

Voids enclose a voided area of the TIN surface. They can be represented by either break, drape or (regular) void types of TIN element. Void geometry is of type gml:Polygon, which might include interior boundaries. All Triangles within a void still exist in the TIN surface but are considered to be void. Triangles inside of the gml:Polygon interior boundary are in the exterior of the gml:Polygon and therefore not in the void; these triangles are therefore not considered to be void.

For a void TIN element of type 'break void' (gml:tin:breakVoid) the boundary linestrings (gml:LineString) of the 3D gml:Polygon behave as breaklines in that triangles in the simple triangle patch collection must be adjusted so that no triangle is crossed by the break void boundary. Part or all of the break void boundary becomes an edge of two or more triangles. The elevation of this break void boundary takes precedence over the elevation of the original triangulated surface for the entire length of the boundary.

For a void TIN element of type 'drape void' (gml:tin:drapeVoid) the boundary linestrings (gml:LineString) of the 2D or 3D gml:Polygon behave as breaklines in that triangles in the simple

Fig. 2: Snippet from GML 3.3 documentation