Representations, 3D

Sisi Zlatanova
Delft University of Technology
s.zlatanova@tudelft.nl

This is an early version and may differ from the published. For last updated version refer to https://doi.org/10.1002/9781118786352.wbieg1157

Abstract

3D modelling has gained increased attention in the last two decades. Technology has developed to such a degree that large data sets are able to be collected, processed, analyzed, managed and visualized in reasonable time. Domain applications are looking with much more interest in 3D representations, acknowledging the increased possibilities for display and analysis of data. The tendency of using 3D data from different domains in one application is growing, and this poses many challenges. Developed independently through the years, 3D applications involve various representations (vector, raster, freeform curves and surfaces), levels of detail, appearance, semantics and topology.

Representing real world in 3D differs significantly from representation in 2D. Technology and approaches for data collection (detect vertical elements of objects, overhanging sections -- as sheds, bridges, and roofs), data processing, data structures, data models and visualization have to be adapted, extended and/or improved to deal with vertical and overhanging elements as well as objects located inside, above or below each other. Many of the well-known approaches in 2D GIS to represent the real world are not readily available for 3D representation. This requires investigations of new possibilities for representation, structuring and visualization of data.

This article outlines the most commonly used 3D geometric representations in terms of their geometry, resolution, topology and semantics, elaborates on their use in wide range of applications and discusses some of the challenges in research and development.

Main Text

Approaches to 3D representations have been independently developed in two broad domains: the real-world and the design-world. Modelling of real-world objects has developed quite differently with respect to the objects of interest and the associated application domain. Real world objects above the surface such as buildings, streets, terrain, rives, etc. have been traditionally in the focus of Geographic Information Systems and represented as 2D objects (vector or raster) for many years. Subsurface modelling of geological and geo-technical phenomena, has long recognized the need for 3D representations (voxels and 3D surfaces). The design-world, i.e. Computer Aided Design (CAD), Computer graphics, Architecture Engineering and Construction (AEC), Building Information Modelling (BIM) on the other hand, has been always dealing with 3D representations, and naturally many of the developments are currently investigated for applications to represent the real world.

Several tendencies are observed:

- The previously sharp boundary between real-world and design-world applications is diminishing. All systems regardless of their origin and focus (GIS, DBMS, BIM, AEC, Computer graphics, serious games, Virtual and Augmented reality) attempt to extend their core functionality to be able to handle 3D geographical data.

- The demand for 3D applications is increasing. The cities grow and the dependencies between buildings, transportation, infrastructure, green and water areas, etc. are becoming more elaborated, which cannot be solved on 2D maps. Numerous examples demonstrate the demand for 3D such as large civil engineering works (Tegtmeier et al 2014), urban planning, environmental monitoring and utility management (Figure 1)

Figure 1: The complexity of underground infrastructure in the area of Port of Rotterdam: above the surface (left) and beneath the surface (right)

- The importance of integrating data from different environments (indoor/outdoor, above/below), applications (cadaster, urban planning, water management), and domains (GIS, BIM) is becoming evident, which results in developing and investigating new approaches and frameworks for integrated data modelling, management and visualisation.
- Technological developments are moving fast: Sensors decrease in size and weight and can be mounted on light platforms (such as drones). 3D sensor measurements are becoming more accurate, cheaper and faster. 3D representations of real world are closely related to the real world phenomena to be modelled. The interest in geo-sciences has traditionally been directed toward real-world objects with spatial extents, a.k.a. spatial objects. A spatial object refers to a real-world object with geometric and thematic (semantic) characteristics (Aronoff 1995, Longley at al 2005). Lately, authors have drawn the attention to a third group of characteristics, i.e. *radiometric* characteristics of spatial objects, which have to reflect surface properties of the object. A further distinction is made between real-world objects with respect to discernible (determined) or indiscernible (undetermined) boundaries (Raper 1998). Usually, discernible objects are *man-made* objects such as buildings, bridges, and streets, and indiscernible objects are *nature objects,* such as geological formations.

Another aspect that influences the 3D representations is the digital visual appearance of the 3D objects. In contracts to 2D, 3D visualizations require a certain level of realism so that users are able to understand the model and the intended message. Digital 3D models have been traditionally a topic of research and investigation of computer graphics and solid modelling (Mäntylä 1988). One of the main purpose of the 3D modelling is to achieve maximal similarity to reality. The model is used to be visualized on a screen as an image or sequence of images (video or animation). The word "scene" is used to denote the rendered image on the screen. Several key components are necessary to achieve readable 3D dimensional images on the screen or, in other words, to create realistic scenes: geometry, illumination, shading, texture, and camera position (or view) (Figure 2). These components attempt to capture features of 3D real objects and represent them in accordance with human perception. 3D geometry determines the position, shape and size of the objects. Illumination and shading models control lighting from artificial light sources and corresponding reflections on the surfaces. Colors and textures aim to represent surface properties of the objects. Textures can be artificial or obtained from real world images. The camera position specifies the location of the user inside the model and orients the model accordingly. The readability of the scene depends on the components used to render the model or part of it on the screen. In this respect, three basic techniques can be distinguished, i.e. points, "empty" polygons (wire frame) and shaded (textured) polygons, containing information about the surface texture (or radiometric properties).

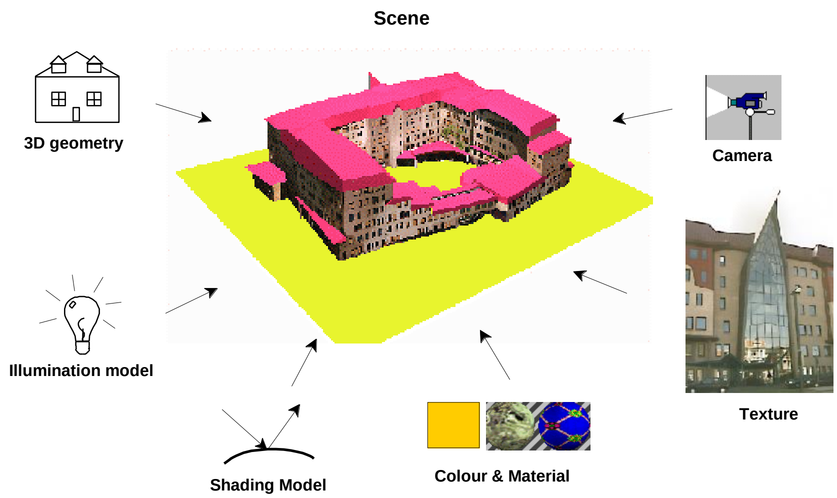


Figure 2: A 3D realistic scene

For digital representations, a description of real-world objects requires a priori clarification of several aspects: 1) space subdivision, 2) geometric primitives to define position/orientation, shape and size of objects, 3) levels of detail, 4) appearance (realism, radiometric properties), 5) topology (spatial relationships), 6) thematic semantics and attributes, and last but not least 7) operations to be performed on the model.

Conventionally, two approaches to space abstraction are utilized in geospatial data modelling, i.e. *field-oriented* and *object-oriented* (Worboys 1995). The field-oriented approach assumes complete subdivision of the space into smaller, often regular partitions, e.g. pixel. In the object-oriented, the space is "an empty container" and all the objects are placed (embedded) in it, i.e. house in a 3D model. Both approaches have advantages and disadvantages and are appropriate for different applications. While the field-oriented approach is better suited for the representation of continuous phenomena, e.g. height fields, rainfalls, the object-oriented approach represents better discrete phenomena, e.g. buildings, roads.

The space subdivision and the geometric primitives have a large impact on the final data structure. Traditionally, within GIS, vector and raster approaches are most used. CAD/AEC/BIM domain distinguishes between larger varieties of 3D geometric representations such as Boundary representations (B-reps), meshes, Constructive Solid Geometry (CSG), voxels, octrees, NURBS, etc. (Mäntylä, 1988). However, they can be also subdivided into two large groups: 3D vector and 3D raster representations. These representations have been analyzed for 3D GIS as well (Latuada, 2006, Abdul-Rahman and Pilouk, 2008)

3D Vector representations

This group of representations comprises approaches that describe the boundary of the objects. The used primitives are 0D (points), 1D (curves) and 2D (surfaces) (Figure 3). Three large groups of representations can be further distinguished: boundary representations (B-reps), Meshes, CSG, parametric and freeform curves and surfaces, such as Non-Uniform Rational Basis Spline (NURBS).

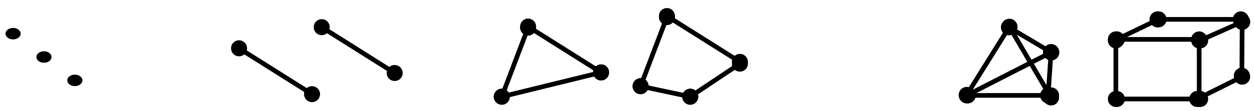


Figure 3: Simple primitives used in B-reps

Boundary representation

Boundary representation is a relatively straight forward approach because the visible surfaces of the geographic objects are modelled. This approach complies with most of the data collection approaches, which also measure properties of visible surfaces. Therefore it has been widely used, applied and investigated for modelling of real-world phenomena above the surface (man-made and nature-made objects). However B-reps can result in very large data sets (in case of high resolution) and very complex data structures if validity and consistency has to be ensured. Many 3D data sets modelled only with geometric data structures reveal many inconsistencies, which obstruct the visualization and complicated performing spatial operations (Ledoux et al 2014). B-reps have been largely employed for geometric representation in GIS. Each primitive is described by low-dimensional primitives applying certain rules. For example, a curve is described by two points, a surface is described by three or more curves and a solid is described by three or more surfaces. The primitives are utilized to describe objects such as buildings and streets. The objects are usually embedded in the 3D space, i.e. full subdivision of space is assumed rarely. A number of rules are to be applied to the geometric primitives and to the way they are combined to define the geographic object. Examples of such rules are: A curve can be restricted to be the straight line between the two points or curves are not allowed to intersect. In case of intersection, the curve has to be split. The two points of a line must not have the same coordinates. A surface (polygon) can be restricted to be composed of only three straight lines (e.g. triangle), otherwise it has to be planar. The order of the curves have to be specified to be able to define the orientation of the surface. The three points of a triangle must not be on one line. A polyhedron (a 3D object) has to comply with the following characteristics (Arens et al 2005): flatness (all polygons should be planar), 2-Manifold (the polyhedron should bound only one volume), simplicity (tunnels are allowed but no inner rings; a line has only two points and all polygons must have area), and orientable (inside and outside must be specified).

The rules and relations between the primitives can be specified in a geometrical or topological data structure (see examples of 3D geometrical data structures below). The data structures take account for the space subdivision, define primitives, apply geometric rules and explicitly store (or not) relationships. The difference between the two types of structures is in representation of the coordinates. While geometrically coordinates define locations of individual primitives, topology maintains the spatial relationships among the primitives. The geometric representation facilitates spatial indexing and metric operations and hence commonly used in DBMS for spatial data management. Topological structures, on the other hand, ensure object validity on editing and data manipulation as well as reducing data duplication. Development in boundary representation with rules for geometrical and topological structures is being extended to 3D data models and data handling.

Open Geospatial Consortium (OGC; www.opeangeospatial.org) has developed two standards for the representation of geometrical structures: Abstract (Topic 1 Feature Geometry, also ISO 19107) and

Implementation specifications. The Abstraction specifications provide a guidance to what geometric primitives should be used for representing geographical features (Figure 4). The implementation specifications for Simple Feature Access (SFA) are described for three different platforms: Structural Query Language (SQL), Common Object Request Broker Architecture (CORBA), and Object Linking and Embedding (OLE)/Component Objects Model (COM). In 1999, the first implementations of the SQL/SFS became available, which marked significant step forward in maturing spatial database management systems (DBMS). Currently, almost all commercial DBMSs support spatial data types, i.e. Oracle, PostGIS, Ingres, MonetDB, Informix, IBM DB2 as most of them adopted OGC standard. Figure 5 shows a SQL geometric feature hierarchy of simple primitive points, curves and surfaces according to OGC SQL/SFS. As it can be observed much richer set of geometric primitives, including 3D shapes (Solid, Cylinder, Sphere, Spline) is proposed in the Abstract specifications (Figure 4).

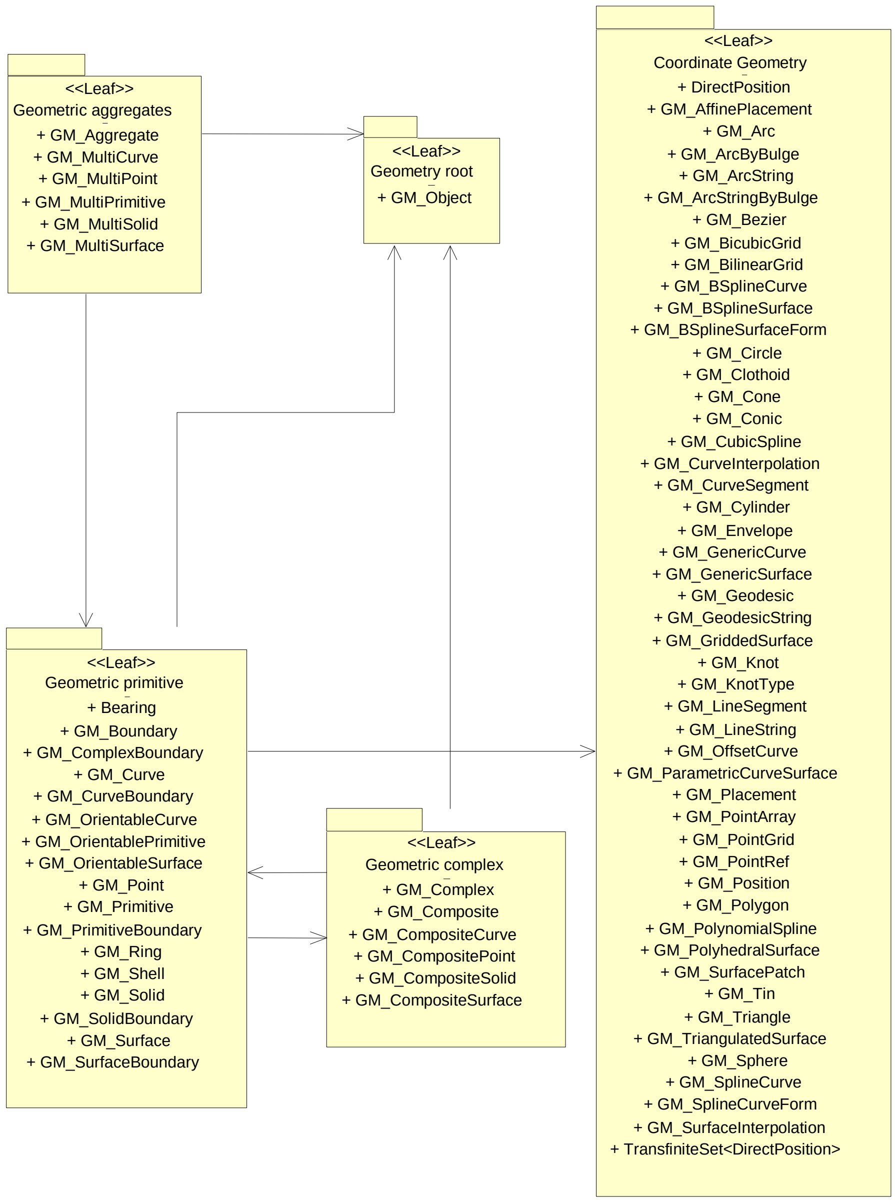


Figure 4: OGC Geometry package (ISO 19107, Feature Geometry, www.iso.org,)

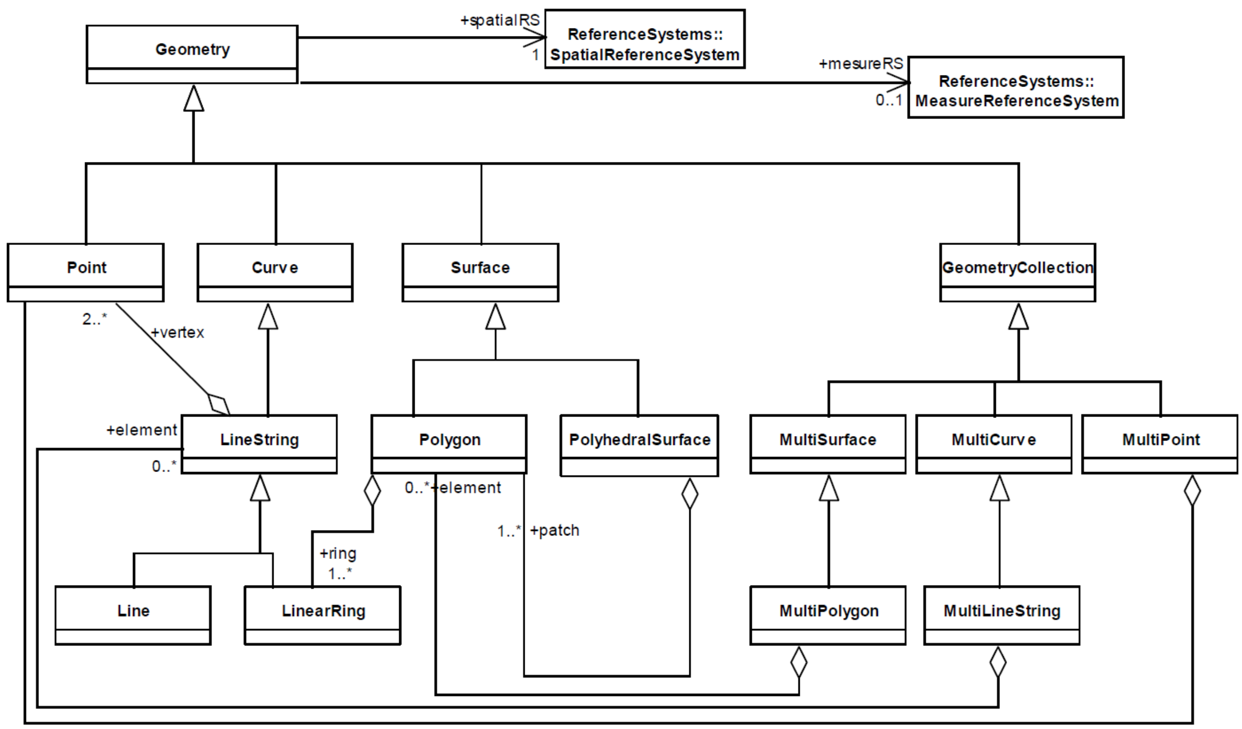


Figure 5: SQL Geometry Feature Hierarchy (OGC Implementation specification for Geographic Information, Part 2: SQL option, hwww.opengeospatial.org/standards/sfs)

Following the Implementation specifications for SQL, two ways of storing geometries in DBMS can be distinguished: using the polygon and multipolygon data type (Figure 6). In the first case, a 3D object will have as many rows as the number of polygons that constitute the 3D object. The table for volumetric objects represented by tetrahedrons will be simpler and can be normalized. It can be organized as a table with five columns: one for the ID of the tetrahedron and four for the composing polygons (triangles). In general, such an organization can be seen as a partial topological model since the 3D object can be defined in reference to the composing polygons, which can be organized in a separate table.

Polyhedron1 (idb, idf, geom)

1	1	geom1
1	2	geom2
1	3	geom3
1	4	geom4
1	5	geom5
…		

Tetrahedron (idt, t1, t2, t3, t4)

1	geom1	geom2	geom3	geom4
2	geom2	geom5	geom6	geom7
3	geom5	geom8	geom9	geom10
…				

Polyhedron2 (idb, geom)

1	geom1
2	geom2
3	geom3
…	

Figure 6: Examples of storing of Polyhedron/Tetrahedron: by polygon (left), triangle and multipolygon

The query below retrieves the geometry of Polyhedron 1, Face 1 as stored in PostGIS:

```
select asewkt(geom) from polyhedron1 where idb=1 and idf=1;

asewkt
----------------------------------
POLYGON((172.578 290.563 7.647,174.891 288.281 7.573,178.625 288.438 7.552,181.
016 291.156 7.574,180.75 294.625 7.586,178 296.938 7.582,174.563 296.781 7.586,1
72.234 293.906 7.65,172.578 290.563 7.647))
(1 row)
```

```
Time: 0.782 ms
```

When multipolygons are the data type to represent 3D features, a 3D object is stored in one row with its object ID and geometry (represented by one multipolygon). This case allows management of only one table, which will be composed of two columns: ID of the object and GEOM for the spatial data type multipolygon. An apparent advantage of the 3D multipolygon approach is the one-to one correspondence between a record in a relational table and an object.

Presently, only few DBMS (e.g. Oracle Spatial, PostGIS) have a volumetric data type. The challenges to the type of volumetric object, its validity and the functions that have to be performed on it are numerous. A simple volumetric object can be represented by polyhedron, triangulated polyhedron and tetrahedron, all of which can be realized with provided spatial data types of polygons and multipolygons (Zlatanova 2006). Moreover, there is no practical difference in the implementation of the polyhedron and triangulated polyhedron, since a separate triangle data type does not exist. Tetrahedrons would allow for a bit simpler representation than polyhedrons or triangulated polyhedrons since a tetrahedron has only four triangles.

User defined spatial data types can be implemented using different approaches from the simple SQL create data type statement, to more complex implementations, using a Procedural Language (PL), Java, C++, etc. The common drawback of such an implementation is that native spatial functionality (operations and indexing) of DBMS cannot be applied to user-defined spatial data types. Moreover the user-defined spatial data types cannot be stored in the same column of the natively supported spatial data types. Visualisation for front- end applications would be possible only by developing individual connections. User-defined spatial data types, nevertheless, are very useful for prototyping of new concepts. For examples Arens et al. (2005) showed implementation of polyhedron data type. Penninga and van Oosterom (2008) proposed an implementation of a tetrahedron data type.

Constructive Solid Geometry (CSG)

CSG has been seen by many as a modification of vector representation (e.g. Latuada 2006) because it utilizes parametrized primitives such as cube, sphere, cylinder, pyramid, etc., which are essentially described by their outer shell. Boolean operations (union, intersection, difference, complement, and cutting) may be operated on the selected initial primitives until the desired shape of the object results. A CSG tree is constructed with primitive solids at the leaves, operations at the inner nodes and the final object at the top level. This approach of 3D representation has many advantages: the primitives are simple volumetric objects, the operators, such as drilling, cutting and gluing, are applicable to the CSG tree, and volumes can be easily computed. This modelling approach has been widely used in Computer-Aided manufacturing to construct machine components, for example. CSG approach has been also used in software design and specifically in BIM (Revit, SketchUp). Many building components (e.g. walls and windows) are structured following the CSG concept.

CSG has limited usage in modelling of real world phenomena. While CSG is easy for design, the modelling of real-world objects is quite tricky and not intuitive. The shape of a real-world object has to be approximated and subdivided into predefined primitives. Only regularly shaped objects, such as buildings, might require less effort (e.g. CityGML LOD1 and LOD2). However, representation of a whole city will result in a complex CSG trees. A positive aspect of this way of modelling is that all objects must be represented by solids as in reality. Real world objects that are currently modelled by 2D primitives (streets, digital terrain models) or even 1D primitives (cables and pipes) can naturally evolve to 3D objects. Research and developments are needed to establish appropriate data management of CSG trees in DBMS. Alternatively, robust operations are needed for conversions of CSG trees to B-reps, and the data types currently available in DBMS (Figure 7). As mentioned previously, the OGC Abstraction specifications suggest that parametric shapes be considered for implementation, but if the CSG tree should be stored in the database remains a question.

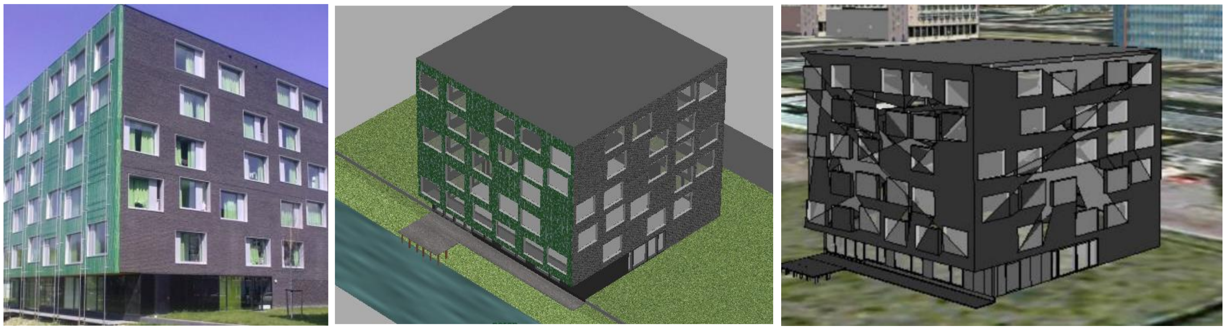


Figure 7: Modelling with CSG: real building (left), modelled as CSG in Maya (middle), exported as B-reps in Google (right) (courtesy of E. Boufidou and M. Koudijs)

The first step towards utilizing CSG for modelling geographical objects is introducing parametric shapes such as cylinder, cone and sphere. Such representation is sufficient for performing numerous analysis but faces challenges when the depth has to be taken into consideration or they have to be visualized in 3D environments, for example in utility works. Three-dimensional lines alone may not be sufficient for 3D visualization because of lacking the volumetric appearance that is required to produce depth perception. Several approaches have been proposed in in the literature. One approach is to model pipelines as 3D lines in the database and only create their 3D volumetric representation for visualization. Another approach is to store the centerlines and geometry of pipeline networks with parametric representation or their real 3D representation. While all approaches have advantages and disadvantages, a smooth transition to manage real 3D pipeline systems depends on the availability of parametric data types and corresponding spatial operations (Doner et al 2011). Experiments with 3D representation of pipelines have revealed that many intersection errors can easily be detected (Figure 8).

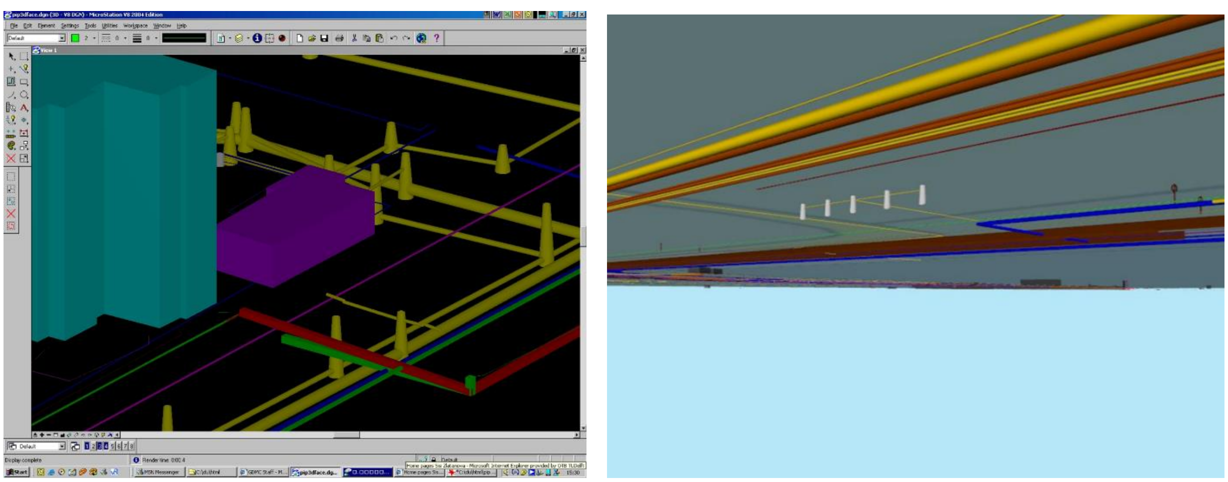


Figure 8: 3D visualization of underground pipes: intersection of two pipe segments (left) and pipes beneath the surface in vicinity of silos foundations (right)

Freeform curves and surfaces

Another approach to 3D representation is utilizing freeform curves and surfaces. There are several methods to represent freeform curves and surfaces. Bézier, B-spline and NURBS methods are among the most commonly used in practice. They are all represented by parametric functions (Piegl and Tiller 1997).

Parametric functions have several advantages over triangulated and polygonal surfaces (meshes): parametric functions have more degrees of freedom to model shapes than predefined shapes, points on a curve or surface can be evaluated numerically and with accurate, and they allow for more compact representations than meshes. The simplest of the three methods to represent a freeform curve is Bézier curve. Its shape is defined by a sequence of n+1 control points P_i (i=0..n) in 3D space. A Bézier surface is, similarly, defined by a grid of (n+1)*(m+1) control points $P_{i,j}$ (i=0..n, j=0..m). To be able to keep the degree of the curves low (3 or 4) the objects are usually modelled by Bézier patches. Such approach cannot ensure the smoothness at the edge of two patches. B-splines overcome this limitation, because the degree of the curve can be defined independently from the number of control points. A B-spline curve of degree d, is defined by a sequence of n+1 control points P_i (i=0..n) in 3D space, and a knot vector of m+1 knots such that m= n+d+1. Though Bézier and B-splines are widely used representations, the most popular method currently for representing freeform shapes is the NURBS method (Figure 9, right). The main difference between NURBS and B-splines is that the control points of a NURBS have weighs that give extra degrees of freedom in modelling curves and surfaces. The important properties of NURBS curves include:

- A NURBS curve is a piecewise rational polynomial curve, and has the same continuity conditions at knots as a B-spline curve.
- NURBS curves are projective invariant, i.e. one can apply affine and projective transformations to the control points.
- NURBS curves can represent conic sections, such as circles and ellipses; that is, they can be used to represent parametric shapes.
- NURBS curves are, just like B-splines curves, locally modifiable and contained within the convex hull of their control points.

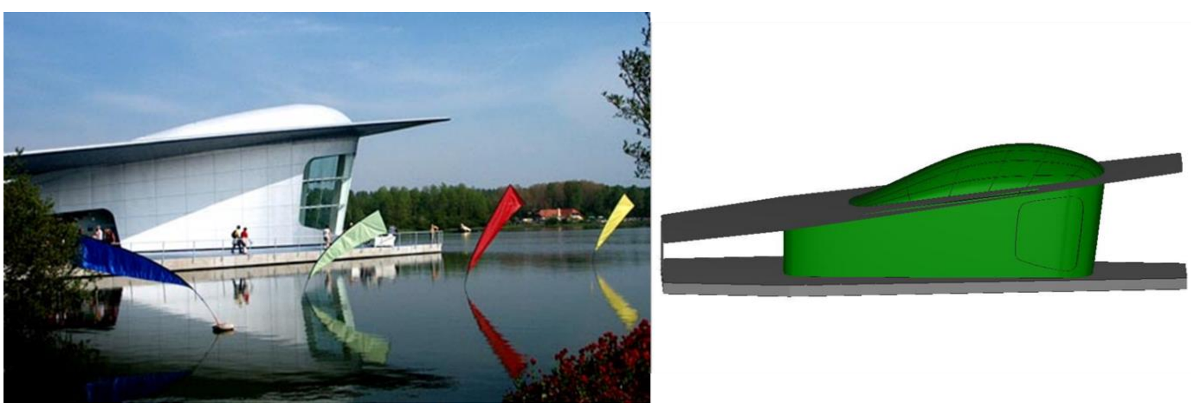


Figure 9: A pavilion modeled with NURBS: building (left) and 3D model (right)

OGC has recognized the importance of freeform curves and surfaces and has included them in the Abstract Specifications, but currently no GIS software of DBMS supports them. They are mostly applied in the design phase. When the construction is built, the surfaces are modelled as simple surfaces and stored as meshes (utilizing the available data types) in the database. Zlatanova et al (2006) demonstrated that

NURBS can easily be stored, visualized and analyzed at DBMS level (Figure 9). A range of questions have to be investigated. For example, the validation rules for freeform curves and surfaces have to be further specified. The present validation functions follow the mathematical definitions of freeform curves and surfaces. The function AnyIntersect, for example, has been implemented using the convex hulls of the control points of two shapes to identify intersections. Opportunities remain for improved accuracy in determining intersections of 3D shapes. NURBS have been used in AEC and for approximation of underground geological formations. A broader implementation in GIS will improve 3D GIS functions and utilities.

3D Raster representation

Another large group of 3D representations is based on exhaustive enumerations: Voxels, Polyhedrons, tetrahedral, octrees, etc. This approach assumes complete subdivision of space into equally shaped (voxels) or irregular shapes (tetrahedrons, polyhedrons) 3D units. Spatial occupancy approach is appropriate for continuous phenomena, mostly nature-made objects such as geological formations, marine and atmosphere phenomena. As rasters in 2D representation, voxels are popular in 3D. Each voxel has distinct properties, with which 3D objects can be created based on a function of properties such as salinity, temperature or pollution parameters.

Voxel representation has the advantage of using a simple primitive and simple data structure. The space subdivision schema can be optimized using octree and binary decompositions. Such schemas allow for flexibility of voxel sizes to account for local or regional homogeneity. In addition, voxel representation offers a convenient data framework for computing, such as calculating volume, intersections, and neighborhood analysis. However they can result in high volumes of data, which need efficient management and indexing. Furthermore, the surface of the voxelized object can be rough, and hence cannot be textured with photo-images (as widely applied in B-reps). Tetrahedrons and polyhedrons are alternatives for better modelling of variant surfaces or boundary objects, albeit at the cost of trading the regularity of voxel data structures.

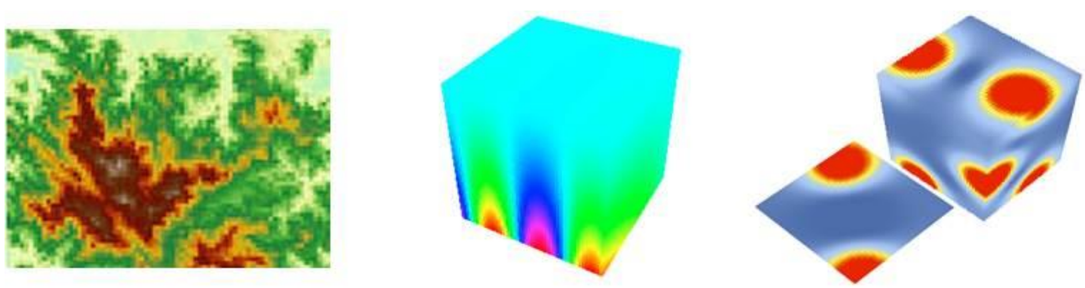


Figure 10: A height map as a 2D raster (left), 3D heat map (middle) and a 3D distance map with a slice of it as a 2D raster.

As mentioned previously many nature phenomena have been modelled with 3D rasters (i.e. voxels). Figure 11 shows the 3D subsurface model of the Netherlands in a voxel representation. It is a nation-wide model covering 41,000km^2 and the upper 500 to 1000 m of the subsurface. The size of the voxels is 100x100x0.5 m^3. The information recorded per voxel describes stratigraphical unit, lithology, grain-size and hydrology, physical and chemical properties as well as associated uncertainty. This data set is used for a number of applications such as groundwater and pollution management, land subsidence studies and infrastructural issues

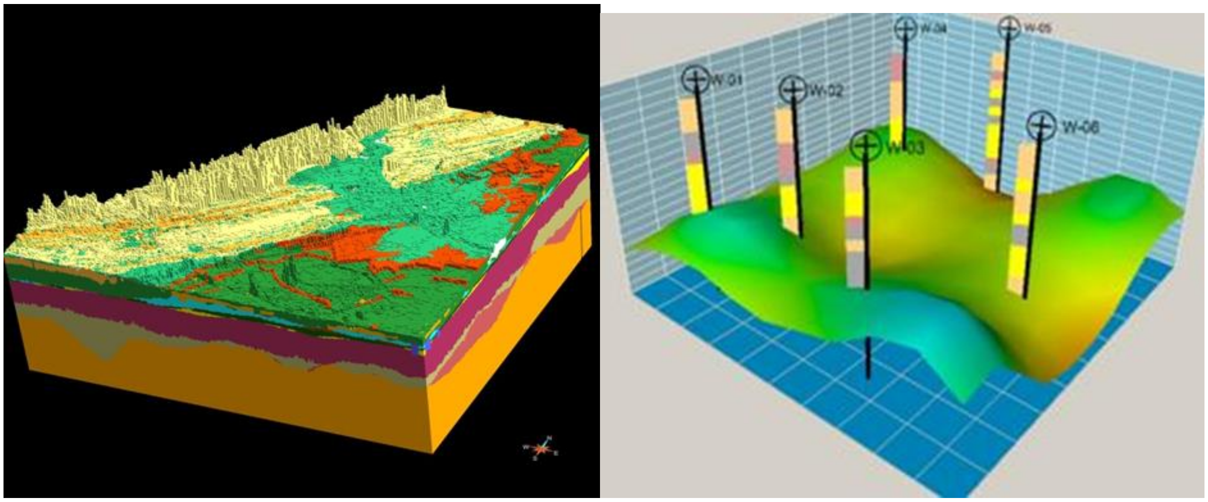


Figure 11: GEOTOP layers with subsurface voxels with similar properties (left) and surface approximation from the voxel data (courtesy The Netherlands Organization for Applied Scientific Research TNO, The Netherlands)

Voxels are commonly used to facilitate the processing of point clouds. Voxelisation processes lead to reduce the size of the point clouds and at the same time introduce regularization for ease of analysis. For example, Figure 12 shows a voxelized point cloud for the Anne Frank tree in Amsterdam the Netherlands. The coloring of the tree (right) is based on the number of points per voxel. The higher the number of the points, the greater the chance is that the voxel represents leaves. The voxels with highest number points are hence colored green. The result appears a good match of reality.

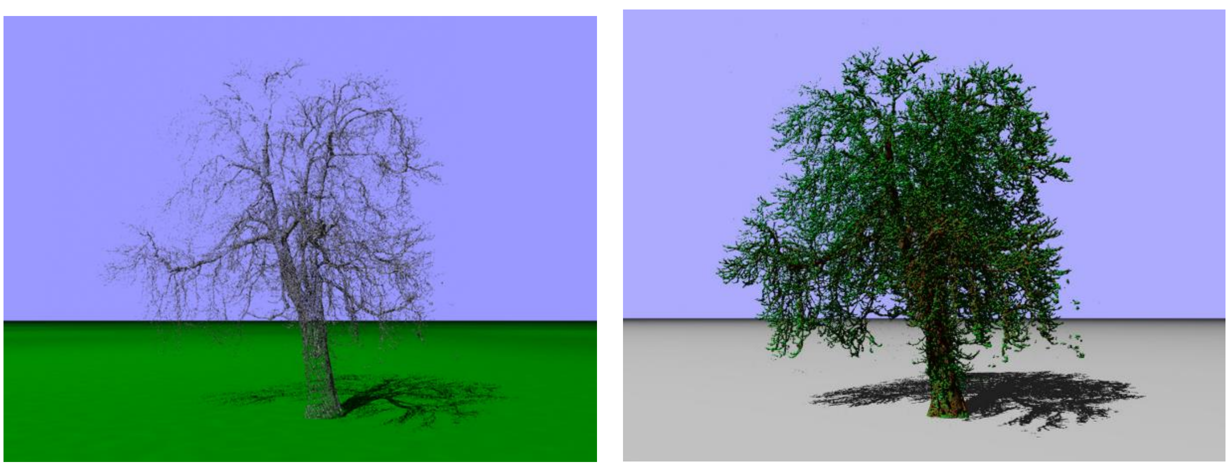


Figure 12: Anne Frank tree: vowelized in a 3D raster of 2500x2500x2500 (left) and 500x500x500 voxels (right).

In recent years, voxel representations have attracted increasing attention due to their simplicity and flexibility in creating representation of different resolutions. Man-made objects such as buildings, streets, bridges are increasingly being modelled with voxels in science, engineering, and entertainment. For example, Minecraft, the game, originally designed for children, is currently being investigated for serious applications as planning and public participation. The game has two modes: survival and building. The building mode allows a complete world to be designed with blocks of $1x1x1m^3$. Figure 13 portrays a Minecraft model of the old city of Delft the Netherlands.

Figure 13: Delft Gate (left) and the city of Delft, year 1572 (right) modelled in Minecraft

Voxels can be created from existing 3D vector models as well. However, 3D vector-raster conversion requires accurate representations of geometry, topology and semantics. Geometrically, voxelization algorithms should be able to deal with points, curves, surfaces and solids. The semantics of the objects depends on the applied semantic model, such as the most commonly used CityGML. Figure 14 illustrates a complete voxelization of a building. Voxels are created from a 3D vector of the model and the semantics of walls, floors, windows, stairs and air is preserved.

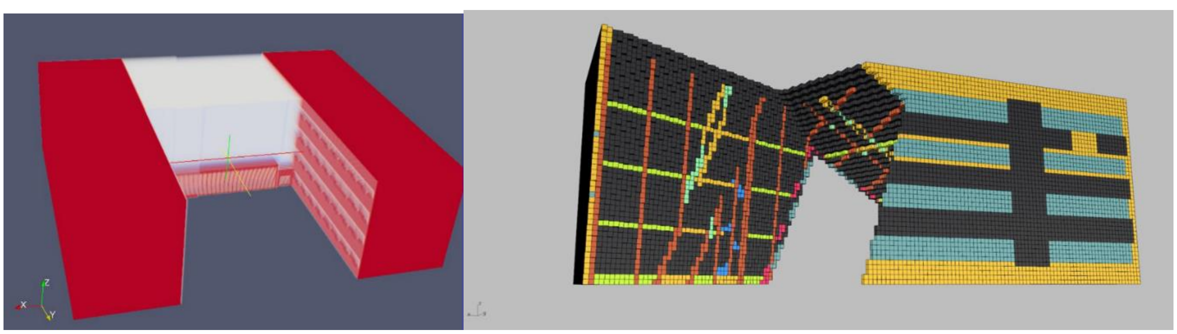
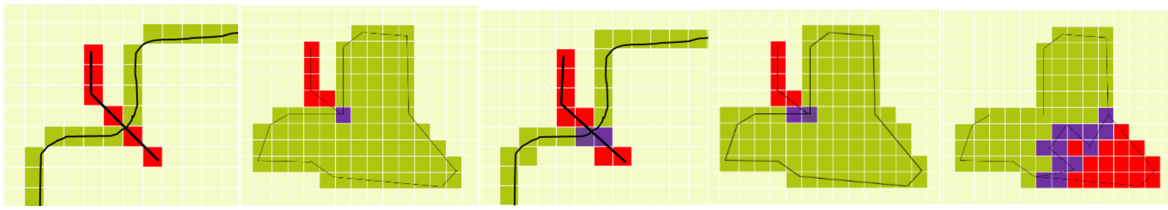


Figure 14: A building of TUDelft campus: vowelized 10x10x10cm^3 *(left) and a cut of the building (walls and floors are visible) (right)*

The neighbourhood relations in the vector domain are represented by the boundaries that completely enclose the objects. Since voxels do not represent boundaries explicitly, the closure of an object is determined by its neighbouring voxels. The neighbouring voxels give an indication of connectivity. Each voxel has 6 neighbouring faces, 12 neighbouring edges or 8 neighbouring vertices. (Rosenfeld, 1981). This results in 6 faces, 18 edges & faces and 26 faces& edges & vertices connectivity in 3D.

The type of connectivity relates to the shape of the voxelized object as well as the topological relationships between the objects. Figure 15 illustrates four cases in which the connectivity may lead to disconnecting or penetrating objects in 2D raster: 8-connectivity is favourable for crossing lines (a), but not for touching between a line and a polygon (b); on the contrary, 4-connectivity is favourable for touching between a line and a polygon (d), but may lead to discontinuity of line in case of crossing lines (c).



a b c d e

Figure 15: 2D raster: crossing 8 connected lines (a), touching 8 connected line and polygon (b), crossing 4 connected lines (c), touching 4 connected line and polygon (d and 8-connected touching polygons. The dark violet pixels in (b), (c) and (e) can lead to penetrated or disconnected objects

One of the best approaches for vector to voxel conversion is the topological method presented by (Laine 2013). The method is intersection-based, which sets a voxel at an intersection target. A voxel's intersection target is a spatial subset of the cubical space occupied by the voxel such as the quadrilaterals bisecting the voxel along its three axes. Intersection targets are chosen based on the desired connectivity level and the dimensionality of the input vector object. Since the proposed method is intersection-based, computational efficiency would be an issue.

3D Semantics

The word "semantics" is used to describe the meaning of things. Semantics can have different interpretations: as opposite to geometry (vector/raster), as a formal connection to reality (nature-oriented) or connection to usage (value-oriented) (Billen et al 2014). In the text below, the semantics is discussed in the context of reality, i.e. the meaning of data objects representing geographical objects in an application model. Application models can emphasize on different aspects of a geographical object, which can lead to associating a real-world object with different semantics. For example a building can be associated with a *building, settlement, construction, house, office, etc.* in different models. The semantics used in 3D city modelling is compatible with the semantics used for 2D topographic maps. Few thematic semantic models for 3D cities exist. A common understanding is that buildings and terrain objects are the most important features to describe. Therefore the following top-level object classes: buildings, streets, green areas, public areas and terrain surface or terrain, vegetation and built forms.

Models can be purely semantic (definitions and relations between the objects on the basis of a theme, e.g. geography) or might be related to spatial, topological and appearance properties. For example, the vocabulary or taxonomy referring to spatial data types (e.g. *polygon*, *multipolygon*, *solid*) as well as topological relationships (*intersect*, *overlap, meet)* is also a kind of semantics. Many on-going standardization initiatives (i.e. OGC, ISO define semantics of geographical objects with respect to their spatial, topological, appearance and thematic properties (geometric representation, time, accuracy). Examples of such standards are Spatial Schema (ISO 19107:2003, 2003), Temporal Schema (ISO 19108:2002, 2002), Quality Principles (ISO 19113:2002, 2002), Quality Evaluation Procedures (ISO 19114:2003, 2003), Metadata (ISO 19115:2003, 2003), Schema for coverage geometry and functions (ISO 19123:2005, 2005) and the XML geometry encoding GML (ISO19136).

In 3D modelling, semantics is commonly used to stress the meaning of the objects and not the meaning of geometric, topological and appearance constructors. A number of semantics models have been defined for 3D city modelling. Towntology is an example of ontology which deals with geographical (named physical), socio-economical, and mental objects in an urban system with part-of relationships. It is possible to observe the first two levels of Physical objects taxonomy (Caglioni 2006). Figure 16 illustrates the main classes of Towntology.

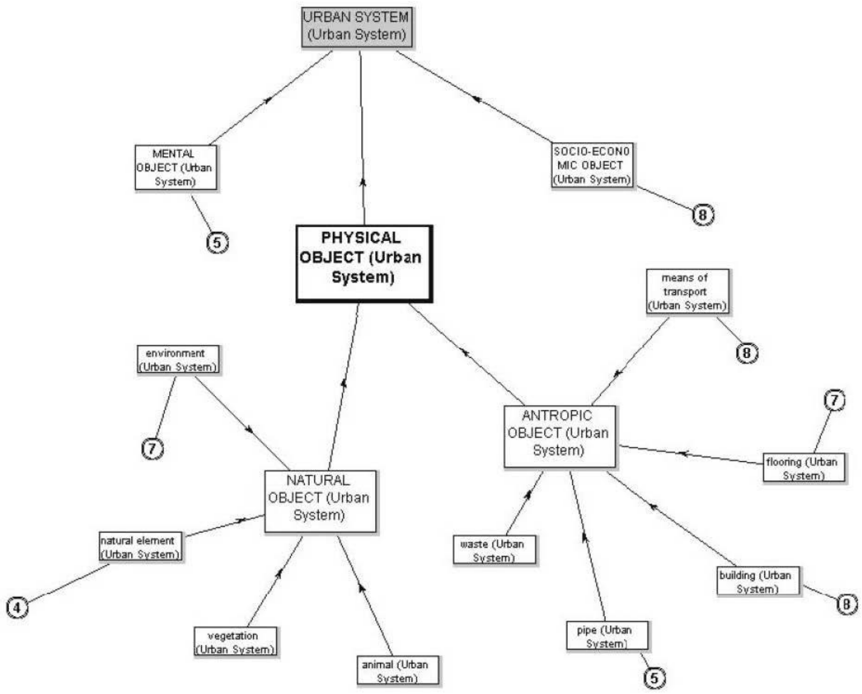


Figure 16 Example of taxonomy for the Urban System built using Towntology software (courtesy Cagliony 2006)

INSPIRE 'Generic Conceptual Model' (INSPIRE, 2008) is another example of defining semantics of objects with considerations of different application models called *themes*. INSPIRE has 34 different spatial data themes, which cover natural, man-made, administrative and environmental objects. The 9 themes ('Annex I', completed by the end of 2009) has an elaborated semantics but not specifically related to 3D. One exception is the theme Cadastral Parcels, which refers to 3D cadastral objects. Annex II (e.g. Elevation and Geology) and Annex III (e.g. Soil, Buildings, Atmospheric conditions, Oceanographic geographical features, and Energy resources) themes are currently being specified to contain more explicit reference to 3D aspects. For example, the draft Data Specification on Energy Resources (INSPIRE TWG ER) mentions 2-D, 2.5-D or 3D geometries.

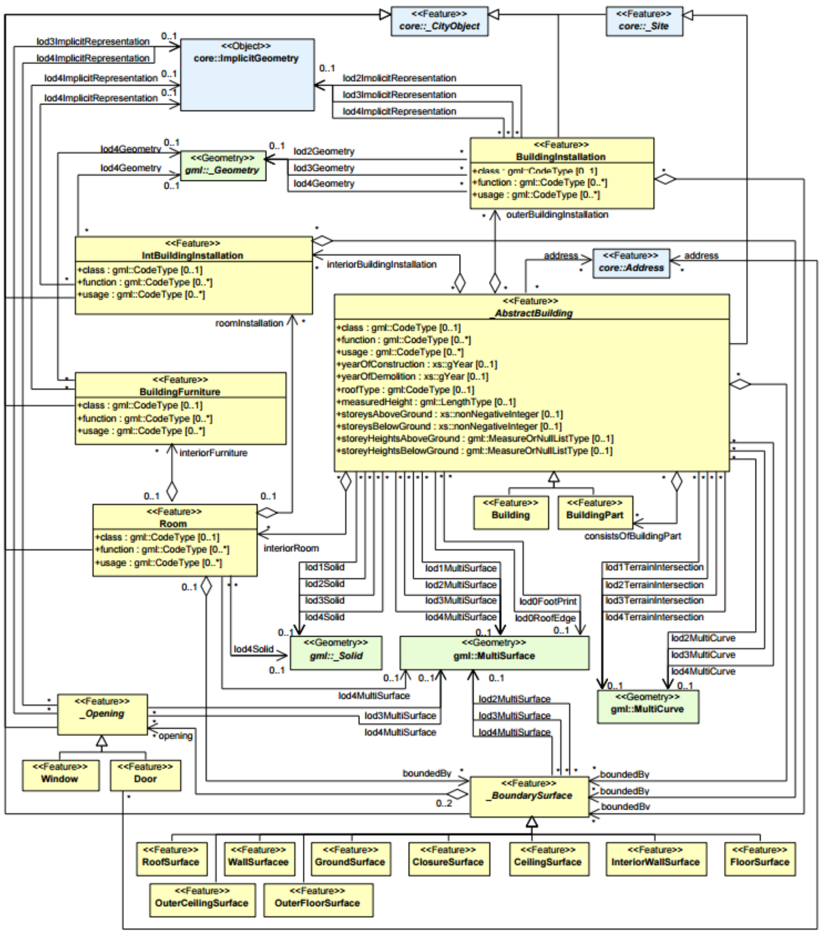


Figure 17: UML diagram of CityGML's building model (courtesy OGC CityGML)

One of the most prominent semantic models for 3D geographic objects is CityGML. Work on CityGML was initiated by Special Interest Group 3D, Germany in 2002 and in 2008 it was adopted as a OGC standard. Since that time many cities in Germany, France, the Netherlands, Switzerland, Turkey, Japan, and Katar have been using and experimenting with CityGML to build their spatial data infrastructure or city models. Vendors such as ESRI, Bentley Systems, Autodesk, FME provide import/export modules. A large number of 3D viewers are also available from KIT Karlsruhe, BS Contact from Bitmanagement Software, Aristoteles from University Bonn, LandXprorer of Autodek, etc.

CityGML maintains information about spatial, semantic, appearance and topological properties of the objects. Semantics is defined for *sites* (buildings, tunnels, and bridges), *transportation facilities*, *water bodies*, *digital terrain models*, *vegetation* (areas, volumes, and solitary objects with vegetation classification), *land use* type, *city furniture* and *generic city* objects. The geometric representation is typical boundary representation, which follows the GML 3D geometry model (based on ISO 19107 Spatial Schema) (Herring 2001). CityGML is also a multi-scale model with 5 predefined Levels of detail: LOD0 – regional landscape; LOD1 – city and region; LOD2 – city districts and projects; LOD3 – architectural models (outside) and landmarks; and LOD4 – architectural models (interior). The LODs have definitions referring to semantic and spatial properties of objects. In this respect, they can be seen as a vocabulary for a LODs. The definition of buildings is most developed and tested among all definitions (Figure 17, Figure 18). The semantic hierarchy goes to *windows*, *doors* and *furniture*. The buildings (or parts of buildings) can be represented by multiple geometries (*solids* or *multisurfaces)*, depending on the purpose of the application. If individual texturing is applied for each surface *multisurfaces* is recommendable, otherwise s*olid* should be more appropriate.

As illustrated in Figure 18, the buildings LOD1 are 'buildings', LOD2 semantic is extended with 'ground surface', 'wall surface', and 'roof surface'; LOD3 adds 'window' and 'door'; LOD4 includes 'room', 'ceiling surface', 'interior wall surface', 'floor surface', closure surface', 'door', 'window', 'building furniture' and 'building installation'. Depending on the LOD, a voxel of a give size can have several different semantic tags, such as (building, roof), (building, room), and (building, window), etc. For example, the 'wall surface' might need to be replaced with the notion of a wall between two rooms. The 'room' is then represented by all voxels that can be placed between the walls.

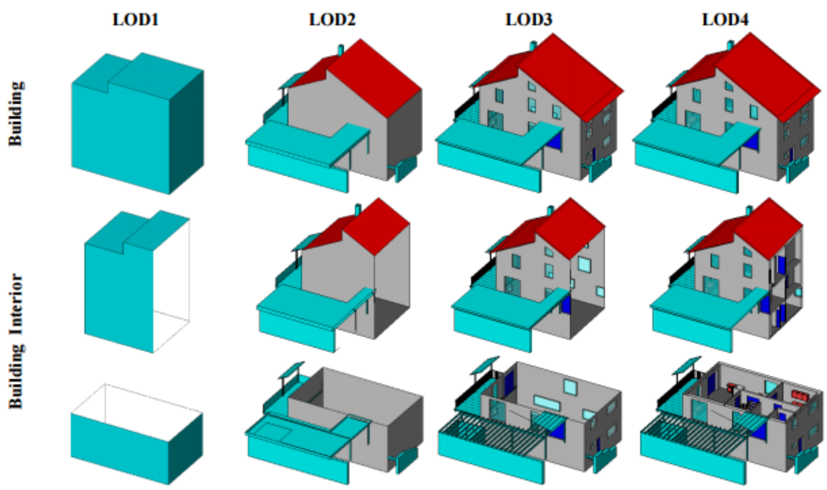


Figure 18: Illustration of LOD1-LOD4 for Buildings (courtesy OGC CityGML)

As mentioned previously, a geographic object can be associated with different semantics. Here examples are given with the 3D model of a building interior. It is differently modelled in three international standards: CityGML, Industrial Foundation Classes (IFC) and IndoorGML.

In CityGML, the building interior is represented similar to the outdoor concept; that is,the visible parts of the interior are represented. *WallSurface, CeilingSurface, InteriorWallSurface, FloorSurface* are identified per room and are represented by surfaces. A building is composed of *Rooms*, a room is composed of the surfaces representing the visible part of walls, celling, floor doors and windows. However, there is no explicit notations for story floors. IFC and IndoorGML take different approaches.

IFC is the mainstream standard of Building Information Modeling (i.e. ISO PAS 16739, 2005). In contract to CityGML, IFC centers on the key building elements (or construction objects), including Beams (*IfcBeam*), Columns (*IfcColumn*), Walls (*IfcWall*), Slabs (*IfcSlab*), and Stairs (*IfcStair*). A building is composed of story floors. Some building elements (i.e. walls, slabs) may contain openings (i.e. voids or holes). All objects are represented by Constructive Solid Geometry or Sweeping primitives. Boundary representation is used very rarely for individual elements such as stairs. The spatial relations are preserved in the form of spatial hierarchies with (spatial) relationship classes. IFC models a wall as a solid (physical wall) between two rooms. Therefore, the notation for a room does not exist. IfcSpace was introduced recently to represent the volume occupied by a space.

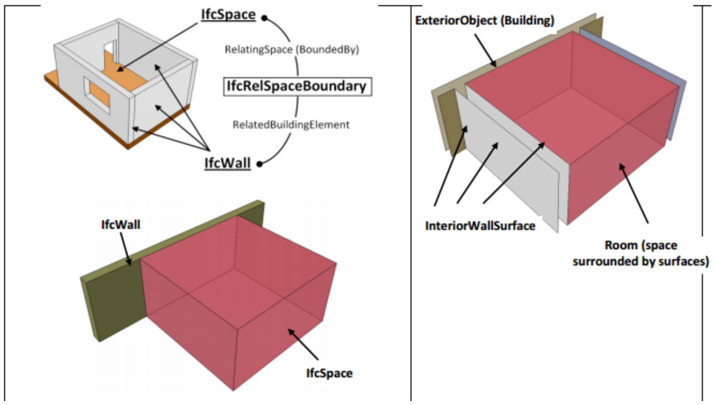


Figure 19: Conceptual difference between IFC (left) and CityGML LOD4 (right) modelling of interior (courtesy Fillipo Mortari).

IndoorGML aims to provide semantic, geometric and navigation model for indoor navigation systems. IndoorGML takes an alternative view to IFC in the Duality graph of indoor spaces and considers spaces (instead of walls and other solid objects) are the most important building elements, which are used to automatically derive network for navigation. The interior of a building is subdivided into appropriate spaces, with respect to user's navigation profile (e.g. visitor, woman, or modes of movement) and the environmental characteristics (e.g. lights, crowds, renovation) of the interior. This subdivision can vary: some spaces can be united in larger units, for example, to represent areas which are not of interest for a visitor, or can be subdivided into smaller units to indicate some functional areas as 'coffee' corner or 'registration area'. The spaces are volumetric objects, conceptually very similar to ifcSpace of IFC. Regardless how the spaces are defined, semantically they are distinguishable only with respect to navigation purpose into *TransitionSpace*, *ConnectionSpace*, *GeneralSpace* and *AnchorSpace* (Figure 20). The spaces should not overlap. The connection space/connection boundary (i.e. doors) is used to derive the logical or metric network.

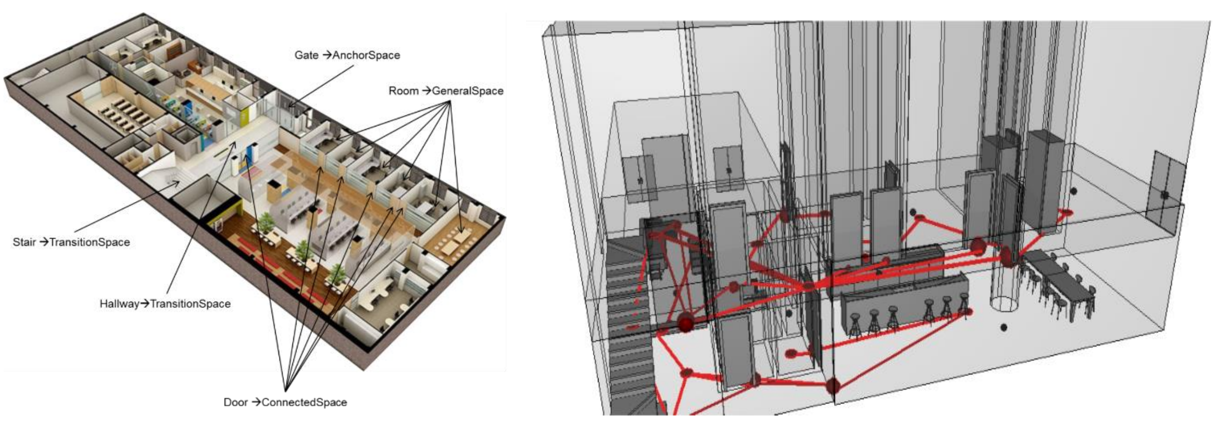


Figure 20: IndoorGML: semantic annotation of spaces (left, courtesy OGC IndoorGML) and space model with automatically derived metric network. (Courtesy L. Liu)

Indoor empty space (free air in a room) can be modelled as *GenericSpace* or *TransitionSpace* (in IndoorGML), *ifcSpace* (in IFC) and *Room* (in CityGML LOD2). The three representations differ geometrically and topologically and should be taken into consideration when modelling interiors. CityGML, IFC and IndoorGML have also been intended for different purposes. CityGML is intended to represent the real world as built, IFC is intended predominantly for design, and IndoorGML is dedicated to a specific application (i.e. indoor navigation). Therefore the semantics differs, which influences also the geometric representation of the topological relationships between objects.

3D topology

Topological data structures are often referred to B-reps. Topology is one of the mechanisms to describe relationships between spatial objects. Models utilizing the topological properties of spatial objects are usually called topological models and are considered best suited to maintain consistency and validity of objects as well as performing certain type of operations. As mentioned previously B-reps require strict rules to maintain consistency of data. They depend on space partitioning, the types and number of primitives, the explicitly stored and derived relationships and the constructive rules. While in 2D, several topological data structures have been developed and widely used (wing edged, wheel-chain), the 3D data structures vary greatly and no 3D topological structure has been implemented. OGC suggests a very general concept of topological data structures but no implementation specifications. The research in the last three decades has contributed to a number of 3D topological data structures such as the 3D Formal Data Structures (3DFDS), Unified Data Model (UMD), Simplified Spatial Structure (SSS), TEtrahedral Network (TEN), and Combinatorial Maps . All data structures have advantages and disadvantages with respect to certain applications. Compared to geometrical, topological structures are slower in metric operations such as computations of area, length, volume or for visualisation, because the coordinates are stored with the nodes. Each metric operation will invoke search of low-dimensional objects until the nodes with the coordinates are reached.

3DFDS is the first data structure that aims at storing real world objects and their spatial relationships. The model consists of three fundamental levels: feature (related to a thematic class), four elementary objects (point, line, body and surface) and four primitives (node, arc, face and edge). The model follows the concept of full space subdivision into non-overlapping objects. Several relationships, i.e. node-on-face, arc-on-face, node-in-body and arc-in-body are explicitly stored. A number of rules specify validy of objects and relationships. The data structure has been tested in research and has exhibited complexity that is difficult to maintain. For example, the faces must be planar, and the explicit relationships should be properly organized.

The TEtrahedral Network (TEN) has been introduced initially by Pilouk (1996) and lately modified by Penninga (2008). TEN has four constructive objects (tetrahedron, triangle, edge, and node). An ARC table stores arc-node relationships; A TRIANGLE table contains tetrahedron-triangle-edge links. A body object is composed of tetrahedrons, a surface object of triangles, a line object of arcs and a point object of nodes. The general rule for creating the model requires that each node is part of an arc, each arc is part of a triangle, and each triangle is part of a tetrahedron. Singularities are not permitted. This data structure simplifies significantly the three-dimensional representations, assuming that the world is partitioned into tetrahedrons and each object (its boundary) is embedded in the tetrahedronisation. The first implementation of this data structure is in a relational data table (Pilouk, 1996). A later implementation of Penninga's TEN model considers mixed geometry/topology description, keeping the coordinates of the primitives as attributes (Penninga 2008). This representation speeds up the search and query of objects and allows spatial indexing to be applied.

The above mentioned models maintain all primitives, i.e. node, edge/arc, face and body. To improve model performance, several researchers suggest to skip maintenance of arc. Two data structures, Simplified Spatial Structure (SSS) and the Urban Data Model (UDM) have clearly demonstrated that search and visualization operations are significantly expedited. While SSS allows polyhedral representations, UDM assumes only triangles to represent the boundaries of all objects. These two data structures are quite similar to the most data structures used for meshes in CAD (e.g. Rhinorous). Data structures without explicit maintenance of arcs (while they still can be implicitly derived) will result in some deficiencies for operations on arc (e.g. 'shortest path' operations). A number of alternatives have been developed to overcome the deficiencies to metric operations. However, the tendency is the same, i.e. improving the performance of topological structures by either reducing the number of primitives or storing the coordinates as attributes of the primitives. More models are discussed in Zlatanova et al (2004).

No one data structure can account for the wide variety of 3D objects and hence cannot be used for all kind of 3D applications. The topological structures depend on many different factors: dimension of the embedding space (2D, 2,5D, 3D, time), used topological primitives (node, edge/arc, face, body), explicit or implicit relationships, topological rules (crossing, on -, in -). Oosterom et al (2002) proposed special encoding of the topological data structures similar to geographical coordinate systems. Table 1 is an example of such a coding. Each topological structure obtains a three-digit code (topological type) and six parameters, i.e. dimension, primitives used, topological tables, explicit relationships, number of tables and rules, which aim at providing all needed metadata. For example TIN has a code 221, which means is 2D and all the information can be organized in two relational tables.

Table 1: Coding of topological data structures (Oosterom et al 2002)

	Topol. type	Dimension	Primitives used	Topological tables	Explicit Relationships	All tables	Rules
TIN	221	2D	Node,edge	node,edge	no	2	Planar Partition
Wing-edge	222	2D	Edge,face	edge,face	no	2	Planar Partition
Wheel (chain)	223	2D	Edge,face	edge,face	no	2	Planar Partition
3DFDS	381	3D	node,arc, edge,face	arc,edge, face	node-on-face node-in-volume arc-partof-line arc-on-face arc-in-volume	8	Space Partition
TEN	352	3D	node,arc, triangle, tetrahedron	arc,triangle, tetrahedron	tri-partof-surf arc-partof-line	5	Space Partition
Cell-tuple	313	3D	0-cell,1-cell 2-cell,3-cell	cells	no	1	Space Partition
SSS	364	3D	Node,face	face,line surface,volume	node-in-volume face-in-volume	6	Space Partition

Hybrid representations

As shown above the 3D geometry can be vector-bases, raster-based but it could be combinations of the two. The interest in hybrid (vector-voxel) representation is becoming increasingly important. Seamless conversions between the two will bring benefits to the type of operations that can be performed. Voxel representations are favorable for volume computations and k-nearest neighbor operations. Vector representation are better for fast, realistic visualization, although much research is completed currently on visualization of (colored) point clouds. *Figure 21* illustrates voxel and vector representation of the same building including interior rasterization. The voxel representation is obtained from the vector (CityGML LOD4), preserving the semantics of the original object. The semantics is added as an attribute to the description of each voxel In this way it is possible to select and visualize all the voxels with a tag 'wall' (*Figure 21*, left).

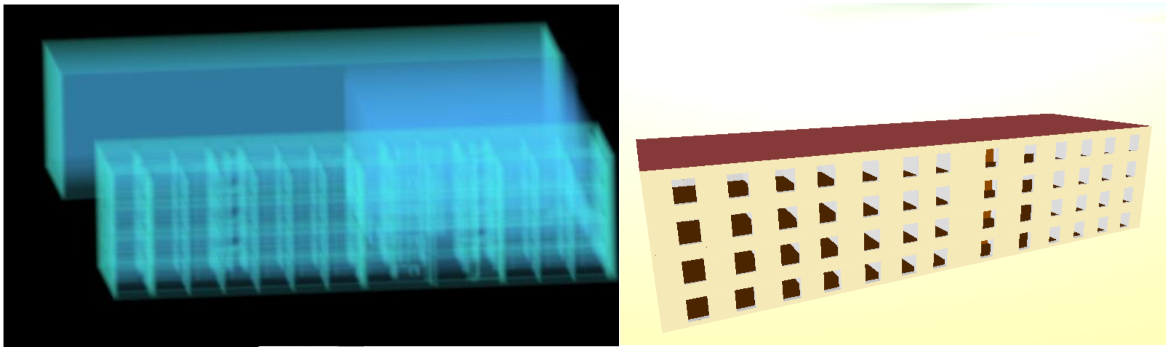


Figure 21: 3Draster 10 cm resolution (left) and vector CityGML LOD4 (right) representation of the same building

The voxel representation allows for quick volume computations (rooms, walls), and various space based simulations e.g. 3D routing, wind/air quality analysis. The vector representation is more convincing for walkthrough, surface computations. The general performance can be improved if a hybrid model is used. Operations can be executed at the representation, which requires less computational complexity. The conversion between the two representations can be performed on the fly if parallel maintenance is not beneficial. Such a flexible voxelisation allows to work with various resolutions. *Figure 22* illustrates the vector model and derived 3Drasters with different voxel size. It clearly shows that objects with crisp boundaries (e.g. the house) need very high resolution to achieve acceptable visual appearance. In contrary the shape of objects with undefined boundaries such as trees could be better represented. Some tiny objects (as the wind mills behind the house on the vector model) will disappear.

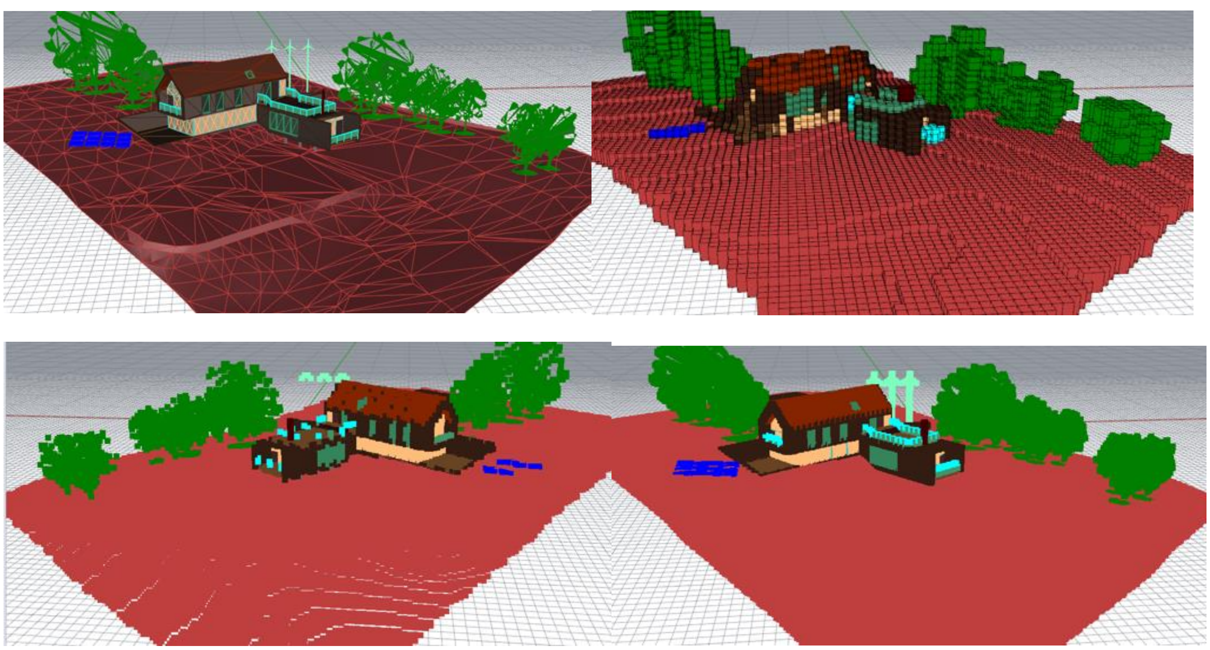


Figure 22: Illustration of objects represented as vector (up-left), 3D raster 50cm (up-right), 3D raster 40 cm (down-left) and 3D raster 10cm (down-right)

Current status and future research

The chapter overviews 3D representation and models developed to account for 3D geometry, semantic and topology. Presently, there is not a single 3D model that can support all applications. The models are still adapted, extended or newly designed to address specific application needs or general purposes (e.g. developing of a national 3D topographic standard). This process affects all components of the 3D representations in a variety of combinations. Most commonly the semantics is adapted and the geometry is kept unchanged, such as extensions of CityGML and IFC (Tegtmeijer et al 2014, Hijazi et al 2012, Isikdag et al 2013, van de Brink et al 2013). Geometric representation might be also changed, but it is mostly for temporary purposes such as integrating B-reps and 3D raster. Very often 3D topologically structured data are converted to non-topological data, but the semantics is preserved. Nevertheless, a strict internationally accepted mechanism for using, adapting and creating new concepts for digital representation of the real world is needed. Figure 23 schematically illustrates linking concepts from different models. To facilitate this process, definitions, validity rules and relations between objects in individual model have to be further clarified and strengthen in the description of the standards.

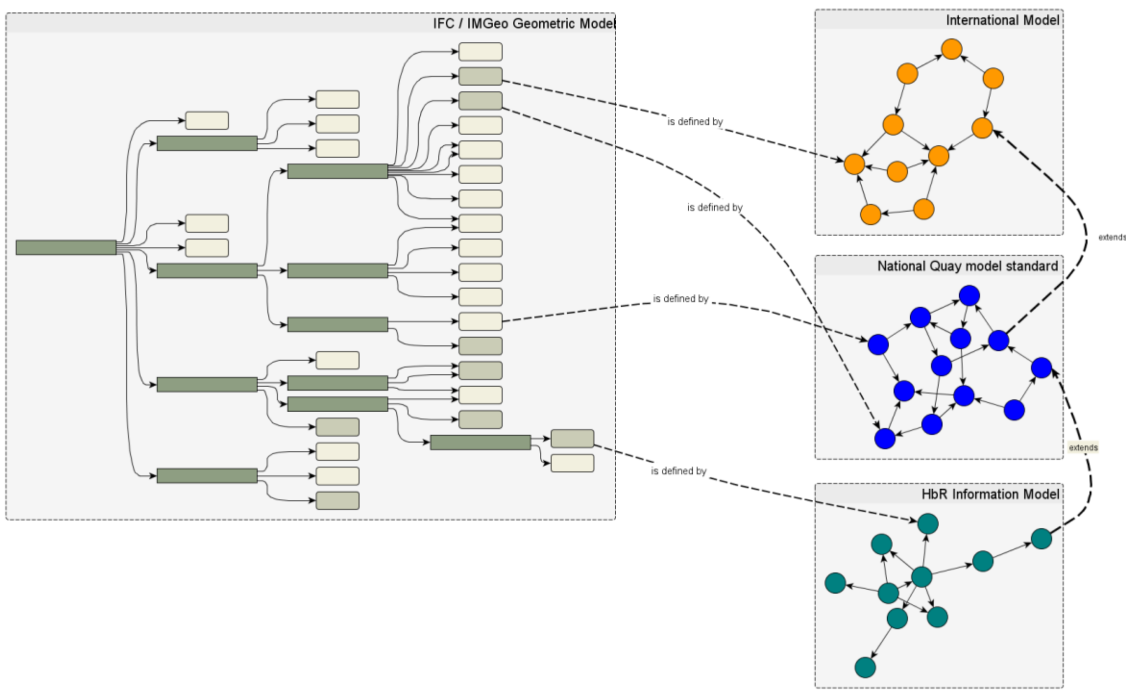


Figure 23: Schematic overview of enriching and re-using semantically rich data models captured in distributed, decentralized and easily extendible models (Zlatanova & Beetz, 2012)

The discussion on which component of 3D representation should be considered primary has been going on for decades. The most commonly accepted vision is that semantics is the very critical component, followed by geometry and topology. Below is a general workflow for 3D data modeling:

- Semantic definition: investigate which objects are needed including the required resolution, accuracy and visualization; apply as much as possible existing definitions and vocabulary; introduce new semantics specifications only when no suitable semantics is available.
- Geometric definition: decide which representation is more appropriate and if needed apply hybrid models or on-the-fly conversions; consider which representations are supported by DBMS. GIS-

based software vendors and mainstream DBMS do not support parametric representations, CAD/AEC – based software vendors heavily rely on them.

- Topological definition: decide to which extend 3D topological data structure is need; investigate options for building topology to check validity of objects or perform certain operation only; consider computational geometry libraries such as CGAL (The Computational Geometry Algorithms Library), which provide robust 3D operations on polyhedra (e.g. Nef-polyhedra) (http://www.cgal.org/)

To conclude, 3D modelling is entering a new phase, which requires more attention and cooperation of different disciplines than ever before. The ultimate goal is be to create a set of different core 3D representations and provide tools for mapping, integration and exchange. First steps in this direction are made by many international organizations that created CityGML and INSPIRE, CityGML (OGC) and IFC (BuildingSMART), and Web3D. Researchers face many new challenges in developing data structures, theoretical frameworks and algorithms. New domains and cross-domains are emerging as well. A typical example is 3D indoor modelling. Research in support of modelling interiors has been active for over thirty years, but it has been mostly for design purposes. Governmental and commercial enterprises as well as individuals are beginning to apply indoor models in their business process and applications. Is the technology developed for outdoor world readily applicable for indoor? 3D representations should be advanced to allow seamless transitions between indoor and outdoor spaces and between different application domains.

SEE ALSO:

Data model (object-oriented), Data structure (vector, raster), Ontology (domain application), Representation (indoor spaces, spatial hierarchy), Spatial database, Topological relationships, 3D structure (acquisition),

References

Abdul-Rahman, A and M. Pilouk, 2008, Chapter 3: 2D and 3D spatial data representations, Spatial Data modelling for 3DGIS, Springer, Berlin, pp 25-43.

Arens, C., J. Stoter and P.van Oosterom, 2005, Modelling 3D spatial objects in a Geo-DBMS using a 3D primitive, Computers &Geosciences, Vol 31. 2, pp. 165-177

Aronoff, S., 1995, Geographic information systems: a management perspective, WDL publications, Ottawa, Canada

Alam, N., V. Coors, S. Zlatanova and P.J.M. van Oosterom, 2012, Shadow effect on photovoltaic potentiality analysis using 3D city models, In: Shortis, Shimoda & Cho (Eds.); International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXIX-B8, XXII ISPRS Congress, August-September 2012, pp. 209-214

Billen, R. and S. Zlatanova, 2003, 3D spatial relationship model: a useful concept for 3D cadastre? In: Computers, Environment and Urban Systems, Vol. 27 (2003) pp. 411-425

Billen, R, A.-F. Cutting-Decelle, O. Marina, J.-P. de Almeida, M. Caglioni, G. Falquet, T. Leduc, C. Métral, G. Moreau, J. Perret, G. Rabino, R. San Jose, I. Yatskiv and S. Zlatanova, 2014, 3D City Models and urban information: Current issues and perspectives, European COST Action TU0801, EDP science, 130p.

Breunig, M. and S. Zlatanova, 2011, 3D geo-database research: Retrospective and future directions, In: Computers & Geosciences, Volume 37, 7, pp. 791-80

van de Brink, L. J. Stoter and S. Zlatanova, 2013, Establishing a national standard for 3D topographic data compliant to CityGML, International Journal of GIS, Issue 1, 2013, pp.92-113

Caglioni, M. 2006. Ontologies of Urban Models, Technical report n°4, Short Term Scientific Mission Report, Urban Ontologies for an improved communication in urban civil engineering projects Towntology Project. Cost Action 21http://www.cost.eu/COST_Actions/tud/Actions/C21

Döner, F., R. Thompson, J. Stoter, Ch. Lemmen, H. Ploeger, P. van Oosterom and S. Zlatanova, 2011, Solutions for 4D cadastre - with a case study on utility networks, In: International Journal of Geographical Information Science, Volume 25, 7, pp. 1173-1189

Emgård, L. and S. Zlatanova, 2008, Implementation alternatives for an integrated 3D information model, In: Van Oosterom, Zlatanova, Penninga&Fendel (Eds,), 2008, Advances in 3D Geoinformation Systems, Lecture Notes in Geoinformation and Cartography, Springer-Verlag, Heidelberg, pp. 313-329

Isikdag, U., S. Zlatanova and J. Underwood, 2013, A BIM-Oriented Model for supporting indoor navigation requirements, Computers, Environment and Urban Systems, Volume 4, September 2013, pp. 112-123

Hijazi, I., M. Ehlers and S. Zlatanova, 2012, NIBU: a new approach to representing and analyzing interior utility networks within 3D geo-information systems, In: International Journal of Digital Earth, Vol. 5. Issue 1, pp. 22-42

Laine, S., 2013. A Topological Approach to Voxelization. Computer Graphics Forum, 32(4), p. 77–86.

Latuada, R., 2006, Three-dimensional representations and Data Structures in GIS and AEC, in Zlatanova and Prosperi, 2006 (eds) Large-scale 3D data integration: Challenges and opportunities, CRCpress, Taylor & Francis Group, Boca Raton, pp. 57-86

Ledoux, H., K. Arroyo Ohori and M. Meijers, 2014, A triangulation-based approach to automatically repair gis polygons. Computers & Geosciences 66, 2014, pp. 121–131.

Longley, P A, M. F. Goodchild, D. J. Maguire, D. W. Rhind, 2005, Geographical Information Systems: Principles, Techniques, Management and Applications, 2nd Edition, Abridged, Wiley

Louwsma, J., S. Zlatanova, R van Lammeren, and P. van Oosterom, 2006, Specifications and implementations of constraints in GIS, In: GeoInformatica, Vol. 10, No. 4, pp. 531-550

Mäntylä, M., 1988, An introduction to solid modelling, Computer Science Press, New York, USA

Penninga, F and P.J.M. van Oosterom, 2008, A simplicial complex-based DBMS approach to 3D topographic data modelling In: International Journal of Geographical Information Science, Volume 22, 7, 2008, pp. 751-779

Pegl , L. and W. Tiller, 1997, The NURBS Book 2nd Edition, Springer-Verlag

Raper, J.F., 1989, The 3-dimentional geo-scientific mapping and modelling system: a conceptual design, Three-dimensional applications in geographic information systems, Taylor & Francis, London

Tegtmeier, W., S. Zlatanova, P.J.M. van Oosterom, H.R.G.K, Hack, 2014, 3D-GEM: Geo-technical extension towards an integrated 3D information model for infrastructural development, In: Computers&Geosciences, Volume 64, March 2014, pp.126-135

Worboys, F.M., 1995, GIS: A computing perspective, Taylor&Francis, London

Zlatanova, S., 2006, 3D geometries in DBMS, In: Rahman, Zlatanova & Coors (Eds.), Innovations in 3DgeoInformation systems, Springer, Berlin, Heidelberg pp.1-14

Zlatanova, S. and D. Prosperi, 2006, Large-scale 3D Data Integration, Challenges and Opportunities, CTCpress, Taylor and Francis, Boca Raton.

Zlatanova, S. and J. Beetz, 2012, 3D spatial information infrastructure: the case of Port Rotterdam. In: Ledux, Moreau & Billen (Eds.), Usage, Usability, and Utility of 3D City Models - European COST Action TU0801 (pp. 1-8). Les Ulis EDP Science

Zlatanova, S., L. Itard, M. S. Kibria and M. van Dorst, 2010, A user requirements study of digital 3D models for urban renewal, In: Open House International, Volume 35, 3, pp. 37-46

Zlatanova, S., S. Pu and W.F. Bronsvoort, 2006, Freeform curves and surfaces in DBMS- a step forward in spatial data integration, In:Nayak, Pathan Garg (Eds.), Proceedings of the ISPRS Commission IV Symposium on 'Geospatial Databases for Sustainable Development', 27-30 September, 2006, Goa, India, Archives of ISPRS Vol. 34, Part 3A, pp.407-412 (

Zlatanova, S., A. A. Rahman and W. Shi, 2004, Topological models and frameworks for 3D spatial objects, In: Journal of Computers & Geosciences, May, Vol. 30, No. 4, pp. 419-428

Key Words
Representation, Data modeling, Visualization, Geospatial, and Cartography