Modelling cities and landscapes in 3D with CityGML

Ken Arroyo Ohori Filip Biljecki Kavisha Kumar
Hugo Ledoux Jantien Stoter

This is an author's version of the paper. The authoritative version is:

Modeling cities and landscapes in 3D with CityGML. Ken Arroyo Ohori, Filip Biljecki, Kavisha Kumar, Hugo Ledoux and Jantien Stoter. In André Borrmann, Markus König, Christian Koch and Jakob Beetz (eds.), *Building Information Modeling: Technology Foundations and Industry Practice*, Springer, September 2018, pp. 199–215. ISBN: 978-3-319-92861-6 (Print) 978-3-319-92862-3 (Online).
DOI: 10.1007/978-3-319-92862-3_11

CityGML is the most important international standard that is used to model cities and landscapes in 3D. Compared to BIM standards such as IFC, CityGML models are usually less detailed but they cover a much greater spatial extent. They are also available in any of five standardised levels of detail. CityGML serves as an exchange format and as a data source for visualisations, either in dedicated applications or in a web browser. It can also be used for a great number of spatial analyses, such as visibility studies and solar potential. Ongoing research will improve the integration of BIM standards with CityGML, making improved data exchange possible throughout the life-cycle of urban and environmental processes.

1 Introduction

Municipalities and other governmental organisations are increasingly using 3D city and landscape models to maintain and plan the environment (see Figure 1 for an example). These models contain 3D data about urban objects such as buildings, roads, and waterways; and the data is collected, maintained and used in applications for urban planning and environmental simulations. Examples of such applications are estimating the shadows cast by buildings and vegetation, simulations of floods and noise propagation, and predicting how much a roof is exposed to sun for assessing the feasibility of installing a solar panel. An overview of applications of 3D city models is available in Biljecki et al. [2015]. The most prominent international standard to define the content of 3D city and landscape models is CityGML Open Geospatial Consortium [2012]; Gröger and Plümer [2012]. The standard was established in 2008 by the Open Geospatial Consortium (OGC) and is an application independent information model and exchange format for 3D city and landscape models. It models semantics, geometry, topology and the appearance of objects. The standard is supported by an increasing number of vendors who provide import and export functionalities as well as viewers. CityGML database implementations are also available.

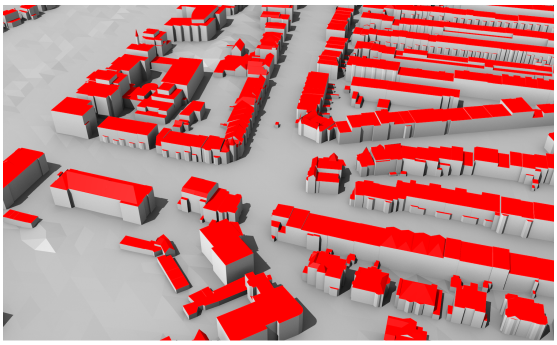

Figure 1: A subset of the Hague in CityGML, containing terrain and buildings. Cities are increasingly investing in CityGML datasets and they are releasing them as open data. Data courtesy of the City of the Hague.

This chapter will give an explanation of the standard while addressing its main principles. The overview of the chapter is as follows:

- Brief overview of the main principles of the standard (Section 2)
- The principle of the Level of Detail (LOD) of CityGML (Section 3)
- Validation of CityGML datasets (Section 4)
- Viewing CityGML data over the Web (Section 5)
- Applications of 3D city models (Section 6)
- Integration between BIM and 3D GIS: IFC and CityGML (Section 7)
- Integration between BIM and 3D GIS: gbXML and CityGML (Section 8)
- Concluding remarks (Section 9)

2 What is CityGML? A short introduction

CityGML was originally developed by the members of the Special Interest Group 3D (SIG 3D) of the initiative Geodata Infrastructure North-Rhine Westphalia (GDI NRW) in Germany. However, it is now an open standard developed and maintained by the Open Geospatial Consortium (OGC).

CityGML defines ways to describe the geometry and attributes of most of the common 3D features and objects found in cities, such as buildings, roads, rivers, bridges, vegetation and city furniture. These can be supplemented with textures and/or colours to give a better impression of their appearance. Specific relationships between different objects can also be stored using CityGML, for example that a building is decomposed into three parts, or that a building has a both a carport and a balcony. CityGML defines different standard levels of detail (LODs) for 3D objects. These provide the possibility to represent objects for different applications and purposes (Section 3).

The types of objects stored in CityGML are grouped into different modules. These are:

- **Appearance**: textures and materials for other types
- **Bridge**: bridge-related structures, possibly split into parts
- **Building**: the exterior and possibly the interior of buildings with individual surfaces that represent doors, windows, etc.
- **CityFurniture**: benches, traffic lights, signs, etc.
- **CityObjectGroup**: groups of objects of other types
- **Generics**: other types that are not explicitly covered
- **LandUse**: areas that reflect different land uses, such as urban, agricultural, etc.
- **Relief**: the shape of the terrain
- **Transportation**: roads, railways and squares
- **Tunnel**: tunnels, possibly split into parts
- **Vegetation**: areas with vegetation or individual trees
- **WaterBody**: lakes, rivers, canals, etc.

It is possible to extend this list with new classes and attributes by defining Application Domain Extensions (ADEs). See Section 6.

2.1 Implementation

In its most common implementation, which is the one generally used to disseminate and exchange data, CityGML datasets consist of a set of plain text files (XML files) and possibly some accompanying image files that are used as textures. Each text file can represent a part of the dataset, such as a specific region, objects of a specific type (such as a set of roads), or a predefined LOD. The structure of a CityGML file is a hierarchy that ultimately reaches down to individual objects and their attributes. These objects have a geometry that is described using the Geography Markup Language (GML) 3.2.1 OGC [2012].

Another important implementation of CityGML is 3D City Database (3D City DB) 3dc [2017]. It is an open source database schema that implements the CityGML standard on top of a standard spatial relational database (Oracle and PostGIS). The 3D City DB content can be exported into KML, COLLADA, and glTF formats for the visualisation in a broad range of applications such as Google Earth, ArcGIS, and the WebGL-based Cesium Virtual Globe.

2.2 Geometry

Since CityGML is an application schema for GML, all geometries supported by GML are supported by CityGML with one exception: while GML allows the use of non-linear geometries, CityGML uses linear ones only. Areal features are represented by triangles and polygons, while volumetric geometries are represented with a boundary representation scheme (*b-rep*) using triangles/polygons.

For representing the exterior of a building, the natural choice is a `gml:Solid` (without interior shells) because it is a volumetric object that must be watertight. Using a `gml:Solid` however implies that the exterior envelope is a 2-manifold, and while the vast majority of buildings can be modelled this way, there are buildings whose exterior envelope is self-tangent. For these, a `gml:Solid` should not be used, and its exterior boundary must instead be stored with a `gml:MultiSurface`, i.e. a set of disorganised surfaces. Another important rule is that the orientation of the surfaces of a `gml:Solid` must be consistent. A complete list of properties can be found in Ledoux [2013].

3 LOD in CityGML

3D city models may be derived in different levels of detail (LODs), depending on the acquisition technique and intended application of the data Kolbe [2009]. CityGML supports storing multiple representations, and it differentiates them by defining five LODs depending on the geometric and semantic complexity of the model (Figure 2).

For buildings, the following LODs are described. **LOD0** is a footprint containing its elevation and optionally a polygon representing the roof edges. Such models present the transition from 2D to 3D GIS, and they do not contain volumetric features. **LOD1** is a block model that is usually derived by extruding a footprint to a uniform height Arroyo Ohori et al. [2015]. Nowadays they are used for a wide range of applications, such as computational fluid dynamics Amorim et al. [2012]. LOD1 models may be acquired automatically with a number of different techniques, such as using existing data in cadastral databases or analysing point clouds derived from airborne laser scanning. Such a favourable balance between usability and easy of acquisition makes LOD1 models popular and widely available Biljecki et al. [2017]. **LOD2** mandates a generalised roof shape and larger roof superstructures. As such, LOD2 models are useful for rooftop solar potential estimations Bremer et al. [2016]. They are usually obtained with photogrammetric techniques, and may be derived automatically Haala and Kada [2010]. **LOD3** is a detailed architectural model containing roof overhangs, openings, and other façade details. Models at LOD3 are usually obtained with a conversion from BIM models or from terrestrial laser scanning Donkers et al. [2016]. The presence of windows and other details makes them useful in applications such as energy simulations Previtali et al. [2014]; Monien et al. [2017]. The LOD taxonomy of CityGML is completed by **LOD4**, which is an LOD3 containing indoor features such as rooms and furniture. LOD4 marks the boundary of GIS and BIM. Datasets modelled at LOD4 are useful for spatial analyses that integrate both outdoor and indoor features. An example of such analyses is simulating floods for predicting damage of buildings Amirebrahimi et al. [2016] and for navigation Vanclooster et al. [2016]; Kim and Wilson [2014].

While many spatial analyses are possible with any of these LODs, data in finer LODs usually comes at a higher accuracy and it may bring more reliable results in a spatial analysis Biljecki et al. [2017]. However, these benefits come at a cost, as datasets modelled at high LODs require more laborious acquisition approaches.

In CityGML, besides the geometric content, each LOD also implies a certain level of semantic information Stadler and Kolbe [2007]. For example, in LOD2 the geometry may be classified into *RoofSurface*, *GroundSurface*, and *WallSurface* among others, which is not possible at LOD1. Nevertheless, CityGML is flexible and it does not mandate semantics, e.g. an LOD2 with only geometry and no semantic differentiation is valid Biljecki et al. [2016].

4 Validation of CityGML datasets

Collecting geographical data about existing physical objects, which is done with different acquisition devices (laserscanners, cameras, total-stations), is prone to errors. These errors often propagate to errors in the 3D objects reconstructed, e.g. part of a roof missing, a bridge not connected to the shore, two houses slightly overlapping, houses "floating" a few centimetres above the ground, etc. Such errors are problematic for different reasons: (1) they hinder interoperability since it can be impossible to convert one format to another if for instance a solid is not watertight; (2) several spatial operations require valid datasets, e.g. if the non-watertight solid is to be used in an application where its volume is necessary, it will be impossible to compute it Steuer et al. [2015]; (3) even for simply visualising a datasets, errors such as duplicated surfaces or surfaces wrongly orientation will cause artefacts that distract the user.

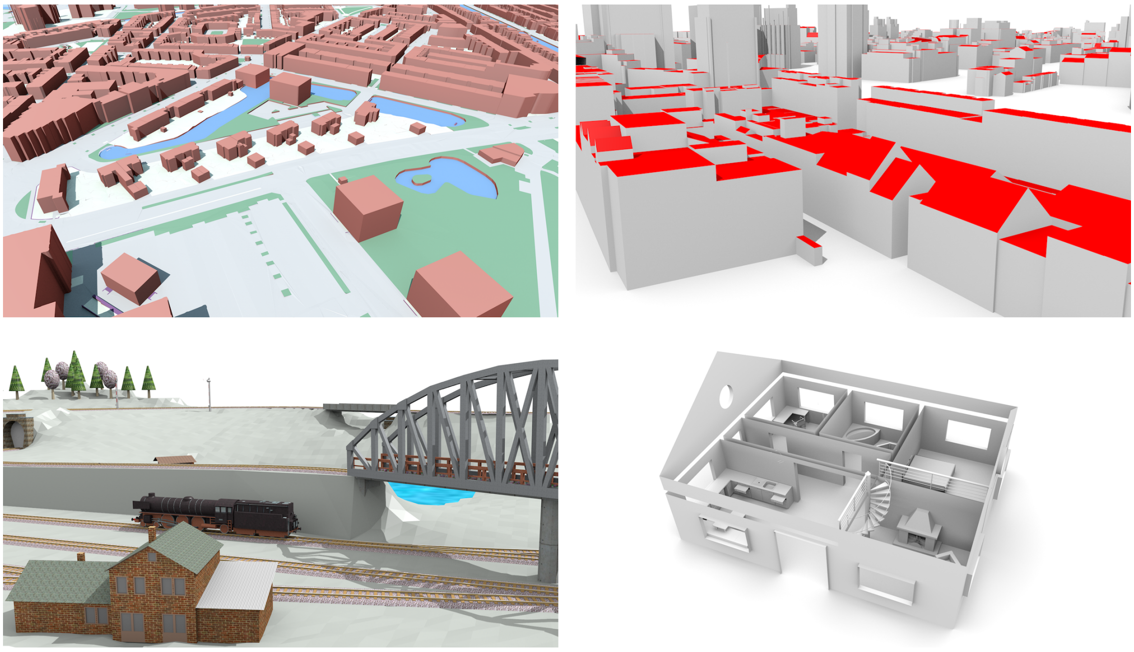

Figure 2: CityGML datasets at different LODs: LOD1 (top left), LOD2 (top right), LOD3 (bottom left), and LOD4 (bottom right). Data courtesy of: Kadaster, AHN, City of Rotterdam, and Karlsruhe Institute of Technology.

The validation of a CityGML dataset means that one must ensure that it conforms to the standardised specifications and definitions as given in Open Geospatial Consortium [2012]. In general, five aspects of data quality should be ensured OGC [2016]; van Walstijn [2015]:

1. schema conformance;
2. geometry;
3. semantics;
4. conformance requirements;
5. application-specific rules.

Tools for the first aspect, which means to verify whether the structure of a GML file conforms to the schemas, are readily available, and this can be considered a solved problem in practice. An open-source tooling that can be used is *Apache Xerces*[1].

For the geometry, the validation means that we need to check whether a given 3D primitive respects the standardised definitions. For the volumetric primitive `Solid`, several errors are possible, e.g. duplicated bounding surfaces, non-watertight boundary, intersecting surfaces, etc. The validation of solids is solved: details of the methodology are available in Ledoux [2013], and there is an open-source implementation available[2]. However, (City)GML datasets contain more 3D primitives, since primitives can be combined into either *aggregates* or *composites*; see Figure 3. An aggregate is an arbitrary collection of primitives of the same dimensionality that is simply used to bundle together geometries; the topological relationships between the primitives are not prescribed. GML has classes for each dimensionality (`Multi*`), the most relevant ones in our context are `MultiSurface` (often used for the geometry of a building) and `MultiSolid`. A composite of dimension d is a collection of d-dimensional primitives whose union forms a valid d-dimensional primitive. The most relevant example in our context is a `CompositeSolid`, which is often used to represent the volumetric part of a building in CityGML. We are not aware of software implementations that are capable

[1] `http://xerces.apache.org`

[2] https://github.com/tudelft3d/val3dity

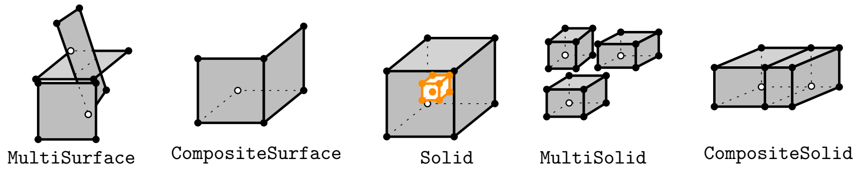

Figure 3: The 3D geometric primitives used in CityGML.

of validating such 3D primitives.

The features in CityGML can have semantics, for instance each of the surfaces used to represent a building can be a semantic class (e.g. roof, wall, window, etc.), which defines its real-world meaning. Depending on the LOD, a semantic surface in a building can be one of nine classes. While it is impossible to validate with 100% certainty the semantics of the surfaces of a building, it is possible to infer it from the orientation of a surface Boeters et al. [2015]; Wagner et al. [2015].

Conformance requirements refer to statements made in the international standard document Open Geospatial Consortium [2012] that cannot be directly implemented. They require the translation of the concept, stated in natural language, into verifiable functions. An example is that if a building is one homogeneous part it should be represented as one `Building`, but different `BuildingParts` should be used if the roof types or if the number of storeys differ, or if the addresses are different. The validation of these requirements require either extra knowledge (information about the addresses in the area) or require specifying what different roof types mean.

Application-specific rules are rules that are not specified in the standard, but that are required in practice. One example is that a building can be required to have a ground floor to form a volume.

Applications of 3D city models (see Section 6) may be affected by missing information and/or inconsistencies in the data, which are not specified in the standard. For instance, that a volume of a building can only be computed if it is modelled by a solid (with a ground floor). CityGML specifies that buildings can be represented with a `MultiSurface`, but if this is the case all applications requiring volumes are not possible without additional processing. Another example is to have consistent attributes (e.g. codes) of buildings when estimating their energy demand. Such inconsistencies may propagate errors when the data is used across different software packages.

5 Viewing CityGML data over the Web

CityGML presents an appealing solution for the storage and exchange of 3D city models because it combines geometry and semantics in a single data model. However, efficiently visualising 3D geometries and semantic information stored in CityGML is complex. A number of desktop viewers are available for the local visualisation of CityGML data such as *FZK Viewer*, *FME Data Inspector* and *azul*. However, the visualisation of CityGML models on the web is however still a challenging area since CityGML is designed for the representation of 3D city models and not for presenting or visualising the 3D city models directly on web.

Among other issues, large CityGML XML files often cannot be rendered directly on a web browser due to memory constraints. Sometimes 3D data cannot be visualised as the user did not install the right browser plug-ins.

Visualising CityGML over the web requires separating the geometric information from the semantic information in the commonly used 3D graphics formats and using these formats to visualise the model. Several 3D graphical standards like X3D[3],

[3] http://www.web3d.org/x3d/what-x3d

KML[4]/COLLADA[5], etc. can be used. It should be noted that when CityGML data is converted to those formats to be able to visualise the data over the Web, often the rich semantics of CityGML are lost.

X3D (Extensible 3D) is an XML-based, open 3D data format that is used for representing 3D scenes in a web environment. It is the successor of VRML[6] (Virtual Reality Modelling Language). Several studies have been conducted to visualise CityGML data over the web browser using X3D. *Mao et al.* Mao and Ban [2011] developed a framework for the online visualisation of CityGML models. In his approach, 3D scenes are generated from the CityGML data based on the geometric and semantic information, which are then viewed in the web browser using X3DOM. Supporting the importance of X3D, *Prieto et al.* Prieto et al. [2012] introduced a framework for the visualisation of CityGML data over the web (without any dependency of plug-ins) using X3D and W3DS (Web 3D Service).

KML (Keyhole Markup Language) is a file format used to display geographic data in an Earth browser such as Google Earth. KML focuses on geographic visualisation, including annotation of maps and images, and version 2.2 has been adopted as an OGC implementation standard. Although KML is not designed for 3D visualisation, it uses COLLADA for 3D modelling. *COLLADA* (COLLAborative Design Activity) is an XML-based open standard for the representation and exchange of 3D assets between applications. It focuses on the exchange of geometric data and 3D scenery. KML/COLLADA is designed for an Earth browser, while X3D is a better choice to present online 3D city models because of its compatibility with HTML and wide support from popular browsers such as Firefox or Chrome.

With the advances in the development of 3D web-based applications, virtual globes have emerged as a new medium for visualising and interacting with geographic information. They provide the user ability

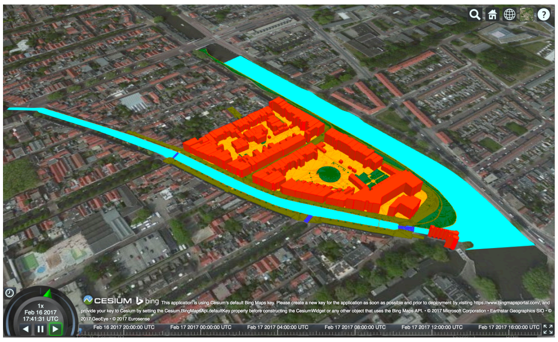

Figure 4: 3D city model of a part of Delft, the Netherlands rendered over Cesium in KML/COLLADA format.

to freely move around in the virtual environment by changing the viewing angle and position. To develop cross-platform and cross-browser applications, several WebGL based virtual globes have been developed like Cesium JS[7], OpenWebGlobe[8] or WebGLEarth[9], etc.. A virtual globe worth mentioning is *Cesium*. Cesium is an open-source JavaScript library to create 3D virtual globes as well as 2D maps on a web browser. However, Cesium does not directly support rendering of CityGML data. As part of preprocessing, CityGML can be converted to KML using 3D City DB, which is used for visualisation on the Cesium globeChaturvedi et al. [2015]. With 3D City DB, it is possible to export the geometric information of the 3D city models to interoperable format such as KML/COLLADA. This is more suitable for visualisation purpose as compared to CityGML (Figure 4). The semantic information can be retrieved from the the 3D City DB using a Web Feature Service. Cesium also supports rendering 3D models in its native format glTF[10] (GL Transmission Format). Collada2gltf & obj2glft[11] are two tools that convert COLLADA & OBJ models to glTF for use with Cesium.

[4]https://developers.google.com/kml/
[5]https://www.khronos.org/collada/
[6]http://gun.teipir.gr/VRML-amgem/spec/index.html
[7]http://cesiumjs.org/
[8]http://www.openwebglobe.org
[9]http://www.webglearth.org/
[10]https://github.com/KhronosGroup/glTF
[11]https://cesiumjs.org/convertmodel.html

6 Applications of 3D city models

3D city models are nowadays used for many different purposes. A recent study identified 29 use cases in dozens of application domains where 3D city models are used Biljecki et al. [2015]. These use cases range from large-scale studies to micro analyses focused on the building level. For example, 3D city models stored in CityGML (but also other formats) may be used in energy planning Agugiaro [2016], change detection Pedrinis et al. [2015], facilitating property taxation Çağdaş [2013], calculating the sky view factor Brasebin et al. [2012], visibility studies Wróżyński et al. [2016], and thermal simulations Zucker et al. [2016].

Each of these applications may require specific semantic data. A case of such an application is analysing building heating energy consumption, which requires data such as building function, number of occupants, and refurbishment information Nouvel et al. [2017]. Owing to its structure and support for such semantic information, CityGML constitutes a powerful platform to support applications.

While CityGML enables storing a number of generic attributes such as the year of construction of a building, it is meant as a generic standard for modelling topographic features. Hence, it is not always possible to store semantic information required by certain applications.

Such domain specific information can be modelled in CityGML either by generic classes or by the definition of an extra formal schema based on the CityGML schema definitions. Such a schema is called a CityGML Application Domain Extension (ADE). The approach of defining an extra formal schema allows definition of new classes, their relationships and attributes and is recommended for applications that require a large number of new features to be defined.

Examples of ADEs to support particular applications are the Immovable Property Taxation Çağdaş [2013], Noise Open Geospatial Consortium [2012], and Energy Nouvel et al. [2015] ADEs. ADEs can also be modelled to support the needs of a specific domain or context like the IMGeo (Information Model for large-scale Geographical Information) ADE in The Netherlands van den Brink et al. [2013a,b]. This ADE models additional attributes to all CityGML classes for specific use as national 3D standard. The IMGeo ADE also adds a 2D geometry to each class to establish a link to the 2D reference data set, i.e. the geometries in 3D extend features that are modelled in the 2D large-scale map. It also adds additional attributes, see Figure 5.

7 Integration of BIM and 3D GIS: IFC and CityGML

BIM and 3D GIS have some overlap as they both model buildings. However, BIM focuses on the range from a building down to the individual components that are used in its construction, while 3D GIS focuses on anything from a single building up to entire cities and countries, including both man-made and natural features. This means that BIM data almost always contains much more detail than GIS data but it also has a much more limited extent.

Because both domains model buildings and constructions, in both GIS and BIM it is widely acknowledged that the integration of their data is mutually beneficial and a crucial step forward for future 3D city modelling. Detailed BIM data can be used to feed GIS data, providing comprehensive data for the interior of buildings—including parts that would otherwise be hidden—and avoiding having to create new building models from scratch when data already exist. At the same time, the extensive coverage and free availability of GIS data is helpful to provide context and georeference BIM data, enabling architects and managers to see how a building related to the surrounding area. In addition, both types of models can be used to perform a very large number of spatial analyses (e.g. water, noise, air quality, energy, building and construction).

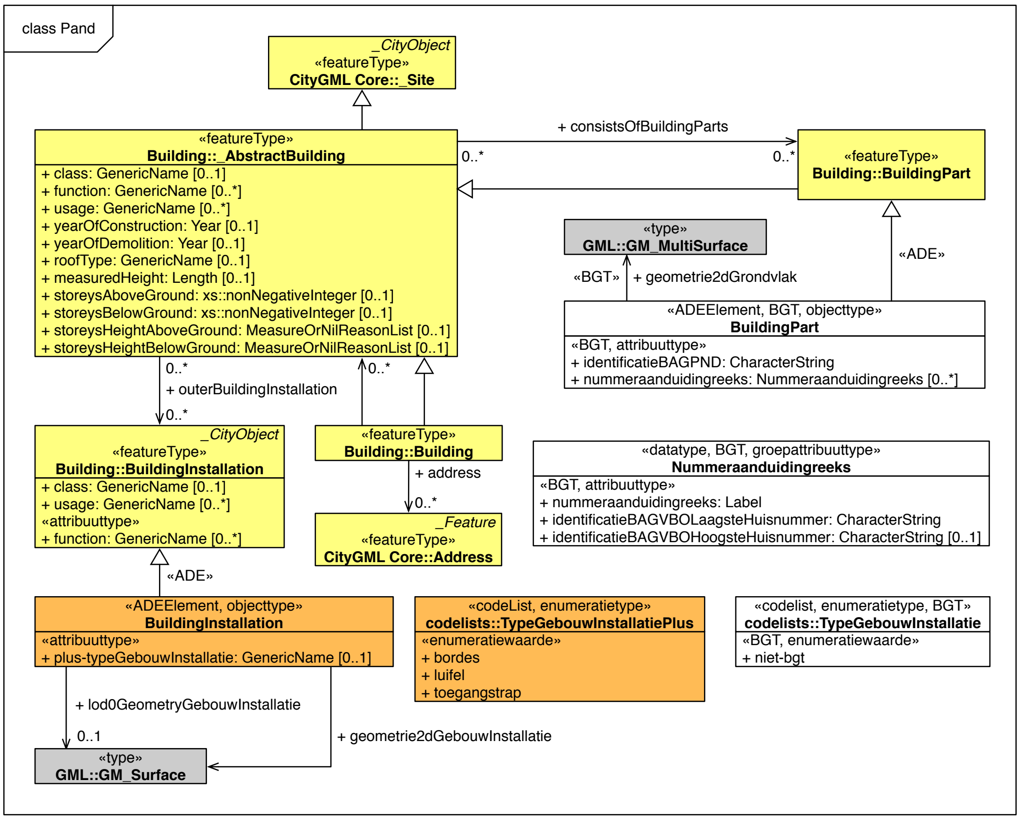

Figure 5: The UML diagram of IMGeo ADE for the CityGML class Building (*Pand* in Dutch). The yellow parts are from the CityGML standard; the rest is the additional information in the application domain extension.

However, BIM and 3D GIS data differ significantly in their modelling paradigms and software tools, which is exemplified by their main open standards: IFC and CityGML. These differ in their approach to model geometry and semantics as well as their level of detail.

For instance, IFC geometries follow three different representation paradigms (i.e. CSG, Sweep Volumes and *b-rep*), while volumetric geometries in CityGML are solely represented with *b-rep*. Individual objects in an IFC file (i.e. entities) are usually designed individually and have their own coordinate system, while objects in a CityGML file are usually modelled together and in the same coordinate system. IFC geometries are mostly representations of a set of *volumes* but CityGML generally models the visible *surfaces* of a building (Figure 6). IFC models are often created during the building design phase, which can differ significantly from how a building was constructed, while CityGML models are usually created by measuring an already existing building. These sort of differences are only a few that show the very different modelling paradigms followed by IFC and CityGML, and in turn by BIM and 3D GIS.

Many researchers and practitioners have studied how to best share information between BIM and GIS, including models that combine both approaches El-Mekawy et al. [2012], the (automatic) generalisation of detailed BIM data for GIS use Geiger et al. [2015], adding more detail to GIS 3D datasets Boeters et al. [2015], and the creation of automatic converters between IFC and CityGML Donkers et al. [2016]. However, solutions on BIM and 3D GIS data integration are so far only partial since it is very complex to reconcile all their differences. Even standard GIS software features such as georeferencing can be a problem in practice with IFC files. This makes it very hard to share 3D information among different users throughout the life-cycle of urban and environmental processes (from planning, design and construction to maintenance).

The two domains of 3D GIS and BIM are increasingly intersecting: BIM methodologies are applied to infrastructural works; city models are getting more detailed; Smart City concepts ask for an integrated reasoning on city infrastructure; and objectives towards sustainability urge for an approach on multiple levels of detail. This will bring added attention to the many open challenges in integrating 3D GIS and BIM data, such as the automatic conversion of models, the inclusion of appropriate semantics, and the preparation of models for various types of spatial analyses.

8 BIM and 3D GIS: BIM gbXML and CityGML

At present, IFC and CityGML are the two most popular standards used for modelling 3D objects in the BIM & 3D GIS domains. As mentioned in Section 7, a lot of work is already done in transforming IFC to CityGML and vice versa. But there is another BIM standard that is relevant for the BIM/3DGIS integration: gbXML.

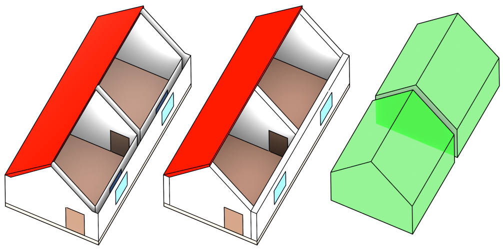

Figure 6: Two modelling paradigms: (left) boundary representation as used in CityGML, (middle+right) space-filling representation as used in IFC.

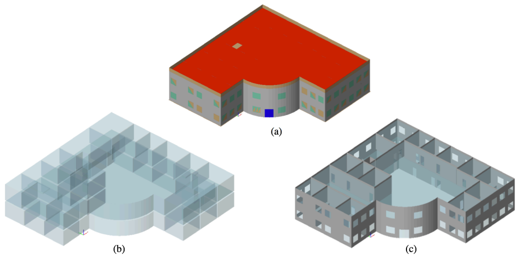

Figure 7: (a) gbXML building model (Source:gbxml.org) (b+c) Spaces in gbXML building model with and without exterior walls.

gbXML[12] (green building XML) is still a new BIM standard and is gaining industry support from leading BIM authoring and analysis software vendors like Bentley and Autodesk. It is an XML-based BIM standard that facilitates the transfer of building information between different BIM models and engineering environmental analysis tools. It provides extensive coverage of the characteristics required for the building energy domain. The gbXML schema comprises nearly 400 elements and attributes for storing information related to building geometry, weather data, spaces, thermal zones, surface adjacency information, etc. Sokolov and Crosby [2011]. The schema is based on the notion of Analytical Space wherein a space represents a volume enclosed by surfaces. In a building, every closed volume is an analytical space which is modelled as shell geometry Figure 7(b). The building components like walls, roofs, and floors are modelled as analytical surfaces Figure 7(c).

CityGML seems to be the best standard for modelling the geometric-semantic relations of 3D city objects. But unlike gbXML, it cannot be directly used as input by energy simulation tools. It is therefore an interesting topic for future research to develop a formal framework for the geometric-semantic transformation of 3D city objects between the two standards, gbXML and CityGML. By transforming 3D objects from

[12] http://www.gbxml.org/

CityGML to gbXML, significant time can be saved during energy simulations as it will not be required to recreate the building geometry within the simulation interface. In today's practice, the gbXML-based BIM models are exclusively being used to derive the thermal properties of building elements (e.g. thermal conductivity and specific heat), which are then directly used by energy simulation tools.

9 Concluding remarks

This chapter provided an explanation as well as background of the international standard that is used to model city and landscapes in 3D: CityGML. It is the dominant standard for 3D city and landscape models, and it is widely adopted by researchers and industry alike. An important characteristic of CityGML is that it models 3D data so that it can be used beyond 3D visualisation. Therefore the data can be used in spatial analyses, e.g. to better understand the physical environment or to better predict the impact on the environment in case of interventions, whether foreseen (like a new road) or unforeseen (emission of a toxic cloud). Since CityGML models similar features as BIM standards, it is interesting to see how both standards can be better aligned to make improved data exchange possible. For a successful integration, it is important to acknowledge the differences in each domain, semantically, geometrically and in their level of detail. Overcoming these differences is still a challenge. This is also true for other domains: it is expected that the main challenge for 3D city modelling the coming years will be the data integration: not only between BIM and CityGML, but also above and underground, voxel and vector, sensors, bathymetry and digital terrain models, etc. This will provide one digital view on the built environment that can support a wide variety of applications, the spot on the horizon of many governmental organisations.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 677312 UMnD).

References

3D City Database (3D city DB: the CityGML database), 2017. URL http://www.3dcitydb.org/.

Giorgio Agugiaro. Energy planning tools and CityGML-based 3D virtual city models: experiences from Trento (Italy). *Applied Geomatics*, 8(1):41–56, March 2016.

Sam Amirebrahimi, Abbas Rajabifard, Priyan Mendis, and Tuan Ngo. A BIM-GIS integration method in support of the assessment and 3D visualisation of flood damage to a building. *Journal of Spatial Science*, 61(2):317–350, April 2016.

J H Amorim, J Valente, C Pimentel, A I Miranda, and C Borrego. Detailed modelling of the wind comfort in a city avenue at the pedestrian level. In T Leduc, G Moreau, and R Billen, editors, *Usage, Usability, and Utility of 3D City Models – European COST Action TU0801*, pages (03008)1–6, Nantes, France, 2012. EDP Sciences.

Ken Arroyo Ohori, Hugo Ledoux, and Jantien Stoter. A dimension-independent extrusion algorithm using generalised maps. *International Journal of Geographical Information Science*, 29(7):1166–1186, 2015.

Filip Biljecki, Jantien Stoter, Hugo Ledoux, Sisi Zlatanova, and Arzu Çöltekin. Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information*, 4(4):2842–2889, 2015.

Filip Biljecki, Hugo Ledoux, and Jantien Stoter. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*, 59:25–37, 2016.

Filip Biljecki, Gerard B M Heuvelink, Hugo Ledoux, and Jantien Stoter. The effect of acquisition error and level of detail on the accuracy of spatial analyses. *Cartography and Geographic Information Science*, Advance online publication, 2017.

Roeland Boeters, Ken Arroyo Ohori, Filip Biljecki, and Sisi Zlatanova. Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry. *International Journal of Geographical Information Science*, 29(12):2248–2268, December 2015.

M Brasebin, J Perret, S Mustière, and C Weber. Measuring the impact of 3D data geometric modeling on spatial analysis: Illustration with Skyview factor. In T Leduc, G Moreau, and R Billen, editors, *Usage, Usability, and Utility of 3D City Models – European COST Action TU0801*, pages (02001)1–16, Nantes, France, October 2012. EDP Sciences.

M Bremer, A Mayr, V Wichmann, K Schmidtner, and M Rutzinger. A new multi-scale 3D-GIS-approach for the assessment and dissemination of solar income of digital city models. *Computers, Environment and Urban Systems*, 57: 144–154, 2016.

Volkan Çağdaş. An Application Domain Extension to CityGML for immovable property taxation: A Turkish case study. *International Journal of Applied Earth Observation and Geoinformation*, 21:545–555, April 2013.

Kanishk Chaturvedi, Zhihang Yao, and Thomas H Kolbe. Web-based exploration of and interaction with large and deeply structured semantic 3d city models using html5 and webgl. In *Wissenschaftlich-Technische Jahrestagung der DGPF und Workshop on Laser Scanning Applications*, volume 3, 2015.

Sjors Donkers, Hugo Ledoux, Junqiao Zhao, and Jantien Stoter. Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings. *Transactions in GIS*, 20(4):547–569, August 2016.

Mohamed El-Mekawy, Anders Östman, and Ihab Hijazi. A Unified Building Model for 3D Urban GIS. *ISPRS International Journal of Geo-Information*, 1(3):120–145, December 2012.

Andreas Geiger, Joachim Benner, and Karl Heinz Haefele. Generalization of 3D IFC Building Models. In *3D Geoinformation Science*, pages 19–35. Springer International Publishing, 2015.

Gerhard Gröger and Lutz Plümer. CityGML – Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71:12–33, July 2012.

Norbert Haala and Martin Kada. An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6):570–580, 2010.

KyoHyouk Kim and John P Wilson. Planning and visualising 3D routes for indoor and outdoor spaces using CityEngine. *Journal of Spatial Science*, 60(1):179–193, May 2014.

T H Kolbe. Representing and exchanging 3D city models with CityGML. In Sisi Zlatanova and Jiyeong Lee, editors, *3D Geo-Information Sciences*, pages 15–31. Springer Berlin Heidelberg, 2009.

Hugo Ledoux. On the validation of solids represented with the international standards for geographic information. *Computer-Aided Civil and Infrastructure Engineering*, 28(9):693–706, 2013.

Bo Mao and Yifang Ban. Online visualization of 3d city model using citygml and x3dom. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 46(2):109–114, 2011.

Dirk Monien, Aneta Strzalka, Athanasios Koukofikis, Volker Coors, and Ursula Eicker. Comparison of building modelling assumptions and methods for urban scale heat demand forecasting. *Future Cities and Environment*, 3(1):2842, January 2017.

Romain Nouvel, Robert Kaden, Jean-Marie Bahu, Jerome Kaempf, Piergiorgio Cipriano, Moritz Lauster, Joachim Benner, Esteban Munoz, Olivier Tournaire, and Egbert Casper. Genesis of the citygml energy ade. In Jean-Louis Scartezzini, editor, *Proceedings of International Conference CISBAT 2015 Future Buildings and Districts - Sustainability from Nano to Urban Scale*, pages 931-936. LESO-PB, EPFL (Lausanne), 2015. doi: 10.5075/epfl-cisbat2015-931-936.

Romain Nouvel, Maryam Zirak, Volker Coors, and Ursula Eicker. The influence of data quality on urban heating demand modeling using 3D city models. *Computers, Environment and Urban Systems*, 64:68–80, July 2017.

OGC. OGC Geography Markup Language (GML) — Extended schemas and encoding rules 3.3.0. Open Geospatial Consortium, 2012.

OGC. OGC CityGML quality interoperability experiment. Open Geospatial Consortium inc., 2016. Document OGC 16-064r1.

Open Geospatial Consortium. OGC City Geography Markup Language (CityGML) Encoding Standard 2.0.0. Technical report, April 2012.

F Pedrinis, M Morel, and G Gesquière. Change Detection of Cities. In *3D Geoinformation Science*, pages 123–139. Springer International Publishing, 2015.

M Previtali, L Barazzetti, R Brumana, B Cuca, D Oreni, F Roncoroni, and M Scaioni. Automatic façade modelling using point cloud data for energy-efficient retrofitting. *Applied Geomatics*, 6 (2):95–113, 2014.

Iñaki Prieto, Jose Luís Izkara, and Francisco Javier Delgado del Hoyo. Efficient visualization of the geometric information of citygml: application for the documentation of built heritage. In *International Conference on Computational Science and Its Applications*, pages 529–544. Springer, 2012.

I Sokolov and J Crosby. Utilizing gbxml with aecosim building designer and speedikon, 2011.

A Stadler and T H Kolbe. Spatio-semantic coherence in the integration of 3D city models. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XXXVI-2/C43:8, 2007.

Horst Steuer, Thomas Machl, Maximilian Sindram, Lukas Liebel, and Thomas H Kolbe. Voluminator—Approximating the Volume of 3D Buildings to Overcome Topological Errors. In *AGILE 2015*, pages 343–362. Springer International Publishing, 2015.

L. van den Brink, J. Stoter, and S. Zlatanova. Establishing a national standard for 3d topographic data compliant to citygml. *International Journal of Geographical Information Science*, 27(1):92–113, 2013a. doi: 10.1080/13658816.2012.667105. URL http://dx.doi.org/10.1080/13658816.2012.667105.

Linda van den Brink, Jantien Stoter, and Sisi Zlatanova. UML-Based Approach to Developing a CityGML Application Domain Extension. *Transactions in GIS*, 17(6):920–942, December 2013b.

Lucas van Walstijn. Requirements for an integral testing framework of CityGML instance documents. Master's thesis, Institute of Geodesy and Geoinformation Science, Technische Universitaet Berlin, 2015.

Ann Vanclooster, Nico Van de Weghe, and Philippe De Maeyer. Integrating Indoor and Outdoor Spaces for Pedestrian Navigation Guidance: A Review. *Transactions in GIS*, 20(4):491–525, 2016.

D. Wagner, N. Alam, M. Wewetzer, M. Pries, and V. Coors. Methods for geometric data validation of 3D city models. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XL-1-W5:729–735, 2015.

Rafał Wróżyński, Mariusz Sojka, and Krzysztof Pyszny. The application of GIS and 3D graphic software to visual impact assessment of wind turbines. *Renewable Energy*, 96:625–635, October 2016.

Gerhard Zucker, Florian Judex, Max Blöchle, Mario Köstl, Edmund Widl, Stefan Hauer, Aurelien Bres, and Jyoti Zeilinger. A new method for optimizing operation of large neighborhoods of buildings using thermal simulation. *Energy and Buildings*, 125:153–160, August 2016.