Modeling Cities and Landscapes in 3D with CityGML

Arroyo Ohori, Ken; Biljecki, Filip; Kavisha, Kavisha; Ledoux, Hugo; Stoter, Jantien

DOI
10.1007/978-3-319-92862-3_11

Publication date
2018

Document Version
Final published version

Published in
Building Information Modeling

Chapter 11
Modeling Cities and Landscapes in 3D with CityGML

Ken Arroyo Ohori, Filip Biljecki, Kavisha Kumar, Hugo Ledoux, and Jantien Stoter

Abstract CityGML is the most important international standard used to model cities and landscapes in 3D with extensive semantics. 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 standardized levels of detail. CityGML serves as an exchange format and as a data source for visualizations, either in dedicated applications or in a web browser. It can also be used for a wide range 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.

11.1 Introduction

Municipalities and other governmental organizations are increasingly using 3D city and landscape models to maintain and plan the environment (see Fig. 11.1 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 exposure of roof surfaces to sunlight to assess the potential of installing solar panels. An overview of applications of 3D city

K. Arroyo Ohori (✉) · K. Kumar · H. Ledoux · J. Stoter
3D Geoinformation, Faculty of the Built Environment and Architecture, Delft University of Technology, Delft, The Netherlands
e-mail: g.a.k.arroyoohori@tudelft.nl; k.kavisha@tudelft.nl; h.ledoux@tudelft.nl; j.e.stoter@tudelft.nl

F. Biljecki
Department of Architecture, School of Design & Environment, National University of Singapore, Singapore, Singapore
e-mail: filip@nus.edu.sg

A. Borrmann et al. (eds.), *Building Information Modeling*,
https://doi.org/10.1007/978-3-319-92862-3_11

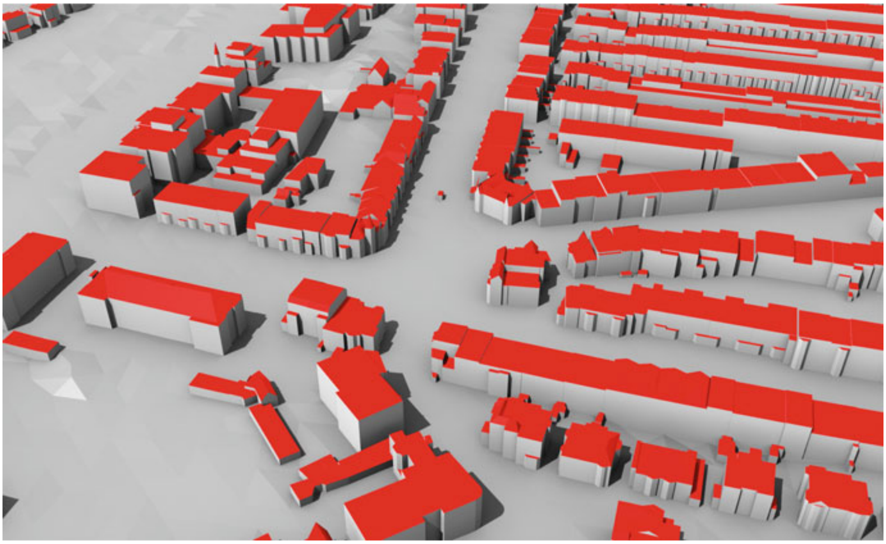

Fig. 11.1 A subset of The Hague in CityGML, containing terrain and buildings. Cities are increasingly investing in CityGML datasets and are releasing them as open data. (Data courtesy of the City of The Hague. © F. Biljecki, reprinted with permission)

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.

This chapter gives an explanation of the standard while addressing its main principles. The chapter is structured as follows:

- Brief overview of the main principles of the standard (Sect. 11.2)
- The principle of Level of Detail (LoD) in CityGML (Sect. 11.3)
- Validation of CityGML datasets (Sect. 11.4)
- Viewing CityGML data over the web (Sect. 11.5)
- Applications of 3D city models (Sect. 11.6)
- BIM and 3D GIS Integration: IFC and CityGML (Sect. 11.7)
- BIM and 3D GIS Integration: gbXML and CityGML (Sect. 11.8)
- Concluding remarks (Sect. 11.9)

11.2 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 colors 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 composed of 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 make it the possible to represent objects for different applications and purposes (Sect. 11.3).

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 Sect. 11.6.

11.2.1 Implementation

In its most common implementation, which is the one generally used to disseminate and exchange data, CityGML datasets consist of a set of 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 extends down to individual objects and their attributes.

Each of these objects have a geometry described using the Geography Markup Language (GML) 3.2.1 (OGC 2012).

Another important implementation of CityGML is the 3D City Database (3D City DB) (3D City Database 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 visualization in a broad range of applications such as Google Earth, ArcGIS, and the WebGL-based Cesium Virtual Globe.

11.2.2 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 only linear geometries. Areal features are represented as triangles and polygons, while volumetric geometries are represented as 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 modeled this way, there are buildings whose exterior envelope is a self-tangent. For these, a `gml:Solid` should not be used, and its exterior boundary must instead be stored as a `gml:MultiSurface`, i.e. an unstructured set of 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).

11.3 LoD in CityGML

3D city models may be derived at different levels of detail (LoDs), depending on the acquisition technique and intended application of the data (Kolbe 2009). CityGML supports storing multiple representations, and differentiates between them by defining five LoDs depending on the geometric and semantic complexity of the model (Fig. 11.2).

For buildings, the following LoDs are described. **LoD0** is a footprint containing its elevation and optionally a polygon representing the roof edges. Such models represent the transition from 2D to 3D GIS but 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). LoD1 models are used for a wide range of applications, such as computational fluid dynamics (Amorim et al. 2012). and can be acquired automatically with a number of different techniques, such as using existing data in cadastral databases or analyzing point clouds derived from

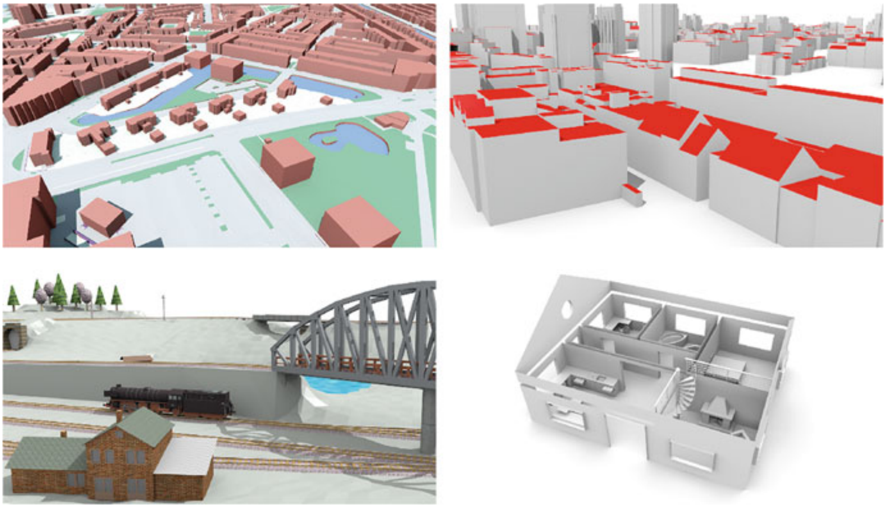

Fig. 11.2 CityGML datasets at different LoDs: LoD1 (top left), LoD2 (top right), LoD3 (bottom left), and LoD4 (bottom right). (Data courtesy of: Kadaster, AHN, City of Rotterdam, and Karlsruhe Institute of Technology. © F. Biljecki, reprinted with permission)

airborne laser scanning. Due to their favorable balance between usability and easy of acquisition, LoD1 models are popular and widely available (Biljecki et al. 2018). **LoD2** includes a generalized roof shape and larger roof superstructures, making them useful, for example, for rooftop solar potential estimations (Bremer et al. 2016). They are usually obtained using 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 by converting data from BIM models or using terrestrial laser scanning (Donkers et al. 2016). The presence of windows and other details makes them useful for applications such as energy simulations (Previtali et al. 2014; Monien et al. 2017). The most detailed LoD in CityGML is **LoD4**, which is an LoD3 containing indoor features such as rooms and furniture. LoD4 marks the boundary between GIS and BIM. Datasets modeled at LoD4 are useful for spatial analyses that integrate both outdoor and indoor features, for example the simulation of floods for predicting damage to buildings (Amirebrahimi et al. 2016), or for navigation purposes (Vanclooster et al. 2016; Kim and Wilson 2014).

While many spatial analyses are possible with any of these LoDs, data at finer LoDs is usually of a higher accuracy and produces more reliable results in a spatial analysis (Biljecki et al. 2018). However, these benefits come at a cost, as datasets modeled at high LoDs require more laborious acquisition approaches.

In CityGML, alongside the geometric content, each LoD 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 necessitate semantics, e.g. an LoD2 with only geometry and no semantic differentiation is still valid (Biljecki et al. 2016).

11.4 Validation of CityGML Datasets

Collecting geographical data about existing physical objects, which can be done with different acquisition devices (laserscanners, cameras, total-stations), is prone to errors. These errors often propagate to errors in the constructed 3D objects, e.g. objects missing part of a roof, a bridge not connected to the shore, two houses slightly overlapping, houses "floating" a few centimeters above the ground, etc. Such errors are problematic for various reasons: (1) they hinder interoperability as non-watertight solids can make it impossible to convert from one format to another; (2) several spatial operations require valid datasets, e.g. the volume of a non-watertight solid cannot be computed making it unusable for some applications (Steuer et al. 2015); (3) errors such as duplicate surfaces or wrongly oriented surfaces in visualizations of datasets cause artifacts that distract the user.

The validation of a CityGML dataset ensures that it conforms to the standardized 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 – verifying 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 tool that can be used is *Apache Xerces*.[1]

Validating geometry means checking whether a given 3D primitive respects the standardized definitions. For a typical volumetric primitive, a `Solid`, several errors are possible, e.g. duplicate bounding surfaces, non-watertight boundary, intersecting surfaces, etc. This too has been solved and details of the methodology are available in Computer-Aided Civil and Infrastructure Engineering 28(9) (Ledoux 2013), along with an open-source implementation.[2] However, (City)GML datasets contain more 3D primitives, since primitives can be combined into either *aggregates* or *composites*; see Fig. 11.3.

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

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

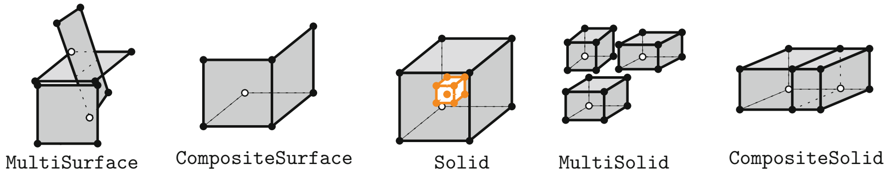

Fig. 11.3 3D geometric primitives used in CityGML. (© H. Ledoux, reprinted with permission)

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*`), of which the most relevant 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. At present software implementations that are capable of validating such 3D primitives are lacking.

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 a concept, expressed in natural language, into verifiable functions. An example is that if a building is one homogeneous volume it should be represented as one `Building`, but different `BuildingParts` should be used if the roof types or if the number of stories differ, or if the addresses are different. The validation of these requirements requires either extra knowledge (information about the addresses in the area) or specifying what different roof types means.

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 Sect. 11.6) 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 modeled by a solid (with a ground floor). CityGML specifies that buildings can be represented as a `MultiSurface`, but in such cases all applications requiring volumes will not be possible without additional processing. Another example is to ensure consistent attributes (e.g. codes) of buildings when estimating their energy demand. Such inconsistencies may result in errors when the data is used across different software packages.

11.5 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 visualizing 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 visualization of CityGML models on the web is still a challenging area since CityGML is designed for the representation of 3D city models and not for presenting or visualizing 3D city models directly on the web.

Among other issues, large CityGML XML files often cannot be rendered directly in a web browser due to memory constraints. Sometimes 3D data cannot be visualized because the user does not have the right browser plug-ins.

Visualizing 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 visualize the model. Several 3D graphical standards such as X3D,[3] KML[4]/COLLADA,[5] etc. can be used but it should be noted that when CityGML data is converted to those formats for visualizing data over the web, the rich semantics of CityGML are often lost.

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

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

[4] https://developers.google.com/kml/

[5] https://www.khronos.org/collada/

[6] http://gun.teipir.gr/VRML-amgem/spec/index.html

is a better choice for presenting 3D city models online due to its compatibility with HTML and good support in popular browsers such as Firefox or Chrome.

With advances in the development of 3D web-based applications, virtual globes have emerged as a new medium for visualizing and interacting with geographic information. They offer users the ability to freely move around in a 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, such as Cesium JS,[7] OpenWebGlobe[8] or WebGLEarth.[9] *Cesium*, for example, 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. In a preprocessing step, CityGML can be converted to KML using 3D City DB, which is used for visualization in the Cesium globe (Chaturvedi et al. 2015). With 3D City DB, it is possible to export the geometric information of the 3D city models to an interoperable format such as KML/COLLADA. This is more suitable for visualization purposes than CityGML (Fig. 11.4). Semantic information can be retrieved from the 3D City DB using a Web Feature Service. Cesium also supports rendering 3D models in its native format glTF[10] (GL Transmission

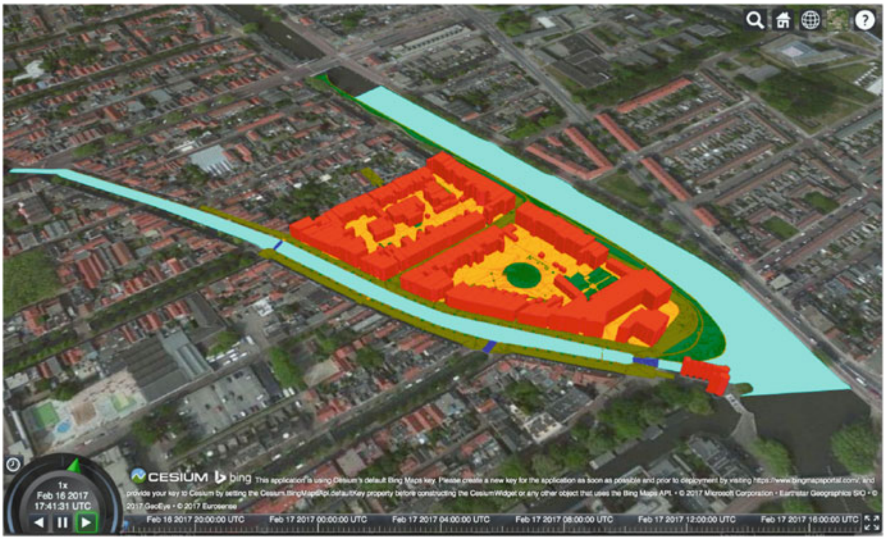

Fig. 11.4 3D city model of a part of Delft rendered over Cesium in KML/COLLADA format. (© K. Kumar, reprinted with permission)

[7] http://cesiumjs.org/

[8] http://www.openwebglobe.org

[9] http://www.webglearth.org/

[10] https://github.com/KhronosGroup/glTF

Format). Collada2gltf & obj2glft[11] are two tools that convert COLLADA & OBJ models to glTF for use with Cesium.

11.6 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 at the level of buildings. 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. One such application is the analysis of building heating energy consumption, which requires data such as building function, number of occupants, and refurbishment information (Nouvel et al. 2017). Due to its structure and support for such semantic information, CityGML constitutes a powerful platform for use in 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 modeling topographic features. Hence, it is not always possible to store semantic information required by certain applications.

Such domain-specific information can be modeled in CityGML either by generic classes or by the definition of an extra formal schema based on the CityGML schema definitions. These schemas are called CityGML Application Domain Extensions (ADE). The approach of defining an extra formal schema makes it possible to define new classes, their relationships and attributes and is ideal for applications that require a large number of new features.

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 modeled 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 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 modeled in the 2D large-scale map. It also adds additional attributes, see Fig. 11.5.

[11] https://cesiumjs.org/convertmodel.html

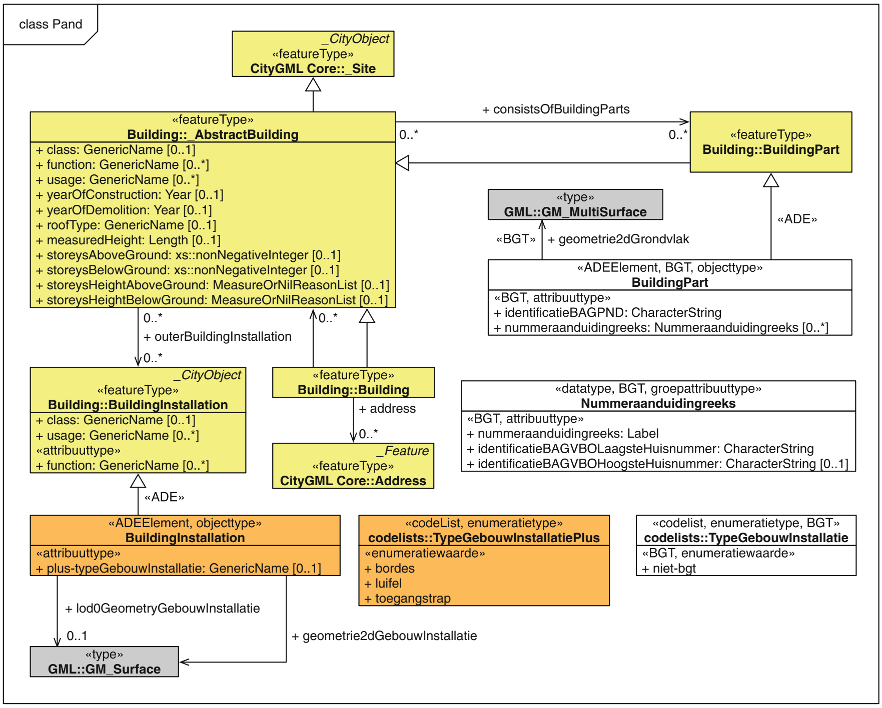

Fig. 11.5 The UML diagram of IMGeo ADE for the CityGML class Building (*Pand* in Dutch). The yellow parts are from the CityGML standard; the rest is additional information in the application domain extension. (© Geonovum, reprinted with permission)

11.7 BIM and 3D GIS Integrations: 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 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, it is widely acknowledged in both GIS and BIM that the integration of their data is mutually beneficial and a crucial step forward for future 3D city modeling. 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 as context and georeference for BIM data, enabling architects and managers to see how a building relates to its surroundings. 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).

However, BIM and 3D GIS data differ significantly in their modeling paradigms and software tools, as 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 modeled together using the same coordinate system. IFC geometries are mostly representations of a set of *volumes* but CityGML generally models the visible *surfaces* of a building (Fig. 11.6). IFC models are often created during the building design phase, which can differ significantly from how it is eventually constructed, while CityGML models are usually created by measuring an already existing building. These differences are just a few that illustrate the very different modeling paradigms of IFC and CityGML, and in turn 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) generalization 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). Up to now, solutions for BIM and 3D GIS data integration have only been partial since it is very complex to reconcile all their differences. Even standard GIS

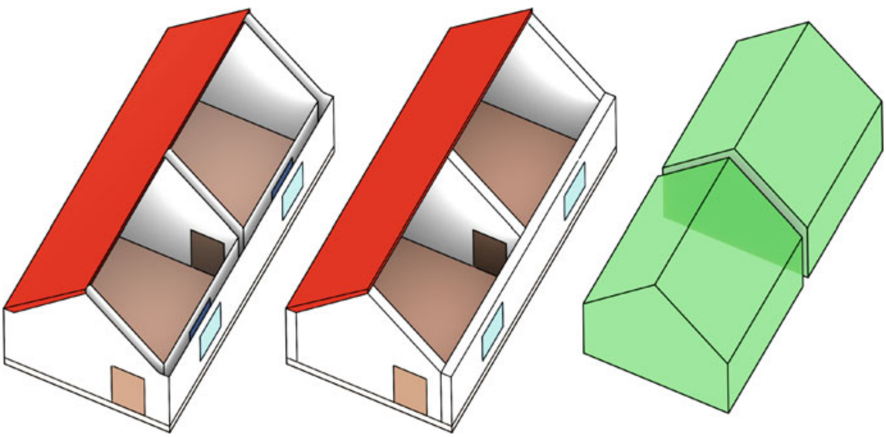

Fig. 11.6 Two modeling paradigms: (left) boundary representation as used in CityGML, (center and right) space-filling representation as used in IFC. (© K. Arroyo Ohori, reprinted with permission)

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 becoming more detailed, Smart City concepts require integrated approaches to city infrastructure, and sustainability objectives require approaches that operate at multiple levels of detail. This will focus further attention to the many yet unresolved 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.

11.8 BIM and 3D GIS: BIM gbXML and CityGML

At present, IFC and CityGML are the two most popular standards for modeling 3D objects in the BIM & 3D GIS domains. As mentioned in Sect. 11.7, a lot of work has already been done in transforming IFC to CityGML and vice versa. But there is also another BIM standard that is relevant for the BIM/3DGIS integration: gbXML.

gbXML[12] (green building XML) is a comparatively new BIM standard that 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 and 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 in which a space represents a volume enclosed by surfaces. In a building, every closed volume is an analytical space which is modeled as a shell geometry (see Fig. 11.7b). Building components such as walls, roofs, and floors are modeled as analytical surfaces (see Fig. 11.7c).

While CityGML is presently the best standard for modeling the geometric-semantic relations of 3D city objects, it cannot, unlike gbXML, be used directly as input by energy simulation tools. An interesting topic for future research will therefore be 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 CityGML to gbXML, significant time can be saved during energy simulations as it will not be necessary to recreate the building geometry within the simulation interface. In current practice, gbXML-based BIM

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

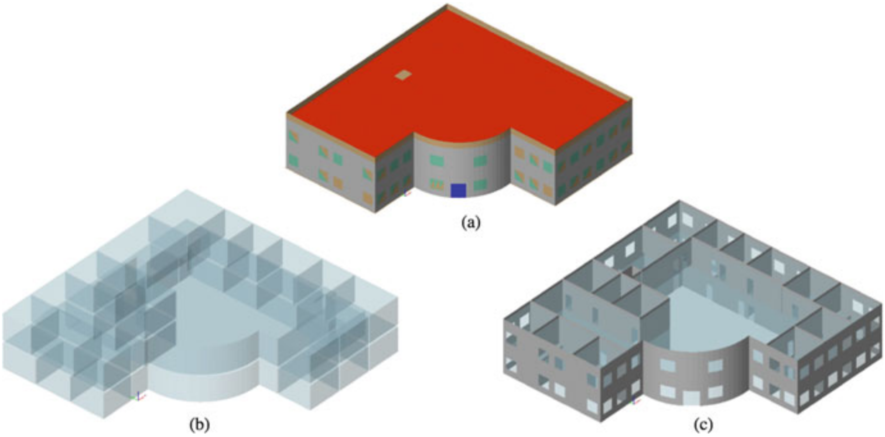

Fig. 11.7 (**a**) gbXML building model (Source:gbxml.org) (**b** and **c**) Spaces in a gbXML building model with and without exterior walls. (© K. Kumar, reprinted with permission)

models are used exclusively to derive the thermal properties of building elements (e.g. thermal conductivity and specific heat), which are then directly used by energy simulation tools.

11.9 Summary

This chapter provided an explanation of and background information on CityGML as an international standard for modeling cities and landscapes. It is the dominant standard for 3D city and landscape models, and is widely adopted by researchers and industry alike. An important feature of CityGML is that it models 3D data so that it can be used beyond 3D visualization. As such, the data can be used in spatial analyses, e.g. to better understand the physical environment or to better predict the impact of interventions on the environment, whether foreseen (such as a new road) or unforeseen (emissions from a toxic cloud). Since CityGML models similar features to BIM standards, it will be interesting to see how both standards could 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 modeling in the coming years will be data integration: not only between BIM and CityGML, but also above and below ground, between voxel and vector, sensors, bathymetry and digital terrain models, etc. This can

potentially result in one digital view of the built environment that can support a wide variety of applications: a point on the horizon that many governmental organizations are looking towards. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 677312 UMnD).

References

3D City Database. (2017). *(3D city DB: The CityGML database)*. Retrieved from http://www.3dcitydb.org/

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

Amirebrahimi, S., Rajabifard, A., Mendis, P., & Ngo, T. (2016). 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.

Amorim, J. H., Valente, J., Pimentel, C., Miranda, A. I., & Borrego, C. (2012). Detailed modelling of the wind comfort in a city avenue at the pedestrian level. In: Leduc, T., Moreau, G., Billen, R. (Eds.), *Usage, usability, and utility of 3D city models – European COST action TU0801* (pp. (03,008)1–6). EDP Sciences, Nantes.

Arroyo Ohori, K., Ledoux, H., & Stoter, J. (2015). A dimension-independent extrusion algorithm using generalised maps. *International Journal of Geographical Information Science, 29*(7), 1166–1186.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., & Çöltekin, A. (2015). Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information, 4*(4), 2842–2889.

Biljecki, F., Ledoux, H., & Stoter, J. (2016). An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems, 59*, 25–37.

Biljecki, F., Heuvelink, G. B. M., Ledoux, H., & Stoter, J. (2018). The effect of acquisition error and level of detail on the accuracy of spatial analyses. *Cartography and Geographic Information Science, 45*(2), 156–176. https://doi.org/10.1080/15230406.2017.1279986

Boeters, R., Arroyo Ohori, K., Biljecki, F., & Zlatanova, S. (2015). Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry. *International Journal of Geographical Information Science, 29*(12), 2248–2268.

Brasebin, M., Perret, J., Mustière, S., & Weber, C. (2012). Measuring the impact of 3D data geometric modeling on spatial analysis: Illustration with Skyview factor. In: T. Leduc, G. Moreau, & R. Billen (Eds.), *Usage, usability, and utility of 3D city models – European COST action TU0801* (pp. (02,001)1–16). EDP Sciences, Nantes.

Bremer, M., Mayr, A., Wichmann, V., Schmidtner, K., & Rutzinger, M. (2016). 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.

Çağdaş, V. (2013). 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.

Chaturvedi, K., Yao, Z., & Kolbe, T. H. (2015). 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* (Vol. 3).

Donkers, S., Ledoux, H., Zhao, J., & Stoter, J. (2016). Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings. *Transactions in GIS, 20*(4), 547–569.

El-Mekawy, M., Östman, A., & Hijazi, I. (2012). A unified building model for 3D urban GIS. *ISPRS International Journal of Geo-Information, 1*(3), 120–145.
Geiger, A., Benner, J., & Haefele, K. H. (2015). Generalization of 3D IFC building models. In M. Breunig, M. Al-Doori, E. Butwilowski, P. V. Kuper, J. Benner, & K. H. Haefele (Eds.), *3D geoinformation science* (pp. 19–35). Cham: Springer.
Gröger, G., & Plümer, L. (2012). CityGML – interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing, 71*, 12–33.
Haala, N., & Kada, M. (2010). An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing, 65*(6), 570–580.
Kim, K., & Wilson, J. P. (2014). Planning and visualising 3D routes for indoor and outdoor spaces using CityEngine. *Journal of Spatial Science, 60*(1), 179–193.
Kolbe, T. H. (2009). Representing and exchanging 3D city models with CityGML. In: S. Zlatanova & J. Lee (Eds.), *3D geo-information sciences* (pp. 15–31). Berlin/Heidelberg: Springer.
Ledoux, H. (2013). On the validation of solids represented with the international standards for geographic information. *Computer-Aided Civil and Infrastructure Engineering, 28*(9), 693–706.
Mao, B., & Ban, Y. (2011). Online visualization of 3D city model using CityGML and X3DOM. *Cartographica: The International Journal for Geographic Information and Geovisualization, 46*(2), 109–114.
Monien, D., Strzalka, A., Koukofikis, A., Coors, V., & Eicker, U. (2017). Comparison of building modelling assumptions and methods for urban scale heat demand forecasting. *Future Cities and Environment, 3*(2). https://doi.org/10.1186/s40984-017-0025-7
Nouvel, R., Kaden, R., Bahu, J. M., Kaempf, J., Cipriano, P., Lauster, M., Benner, J., Munoz, E., Tournaire, O., & Casper, E. (2015). Genesis of the CityGML energy ADE. In: J. L. Scartezzini (Ed.), *Proceedings of the International Conference on CISBAT 2015 Future Buildings and Districts – Sustainability from Nano to Urban Scale*, LESO-PB, EPFL (Lausanne) (pp. 931–936).
Nouvel, R., Zirak, M., Coors, V., & Eicker, U. (2017). The influence of data quality on urban heating demand modeling using 3D city models. *Computers, Environment and Urban Systems, 64*, 68–80.
OGC. (2012). OGC geography markup language (GML) – Extended schemas and encoding rules 3.3.0. Open Geospatial Consortium.
OGC. (2016). OGC CityGML quality interoperability experiment. Open Geospatial Consortium inc., document OGC 16-064r1.
Open Geospatial Consortium. (2012). OGC city geography markup language (CityGML) encoding standard 2.0.0. Technical report.
Pedrinis, F., Morel, M., & Gesquiére, G. (2015). Change detection of cities. In M. Breunig, M. Al-Doori, E. Butwilowski, P. V. Kuper, J. Benner, & K. H. Haefele (Eds.), *3D geoinformation science* (pp. 123–139). Cham: Springer.
Previtali, M., Barazzetti, L., Brumana, R., Cuca, B., Oreni, D., Roncoroni, F., & Scaioni, M. (2014). Automatic façade modelling using point cloud data for energy-efficient retrofitting. *Applied Geomatics, 6*(2), 95–113.
Prieto, I., Izkara, J. L., & del Hoyo, F. J. D. (2012). Efficient visualization of the geometric information of CityGML: Application for the documentation of built heritage. In B. Murgante, O. Gervasi, S. Misra, N. Nedjah, A. M. A. C. Rocha, D. Taniar, & B. O. Apduhan (Eds.), *International Conference on Computational Science and Its Applications* (pp. 529–544). Berlin/Heidelberg: Springer.
Sokolov, I., & Crosby, J. (2011). *Utilizing gbXML with AECOsim building designer and speedikon*.
Stadler, A., & Kolbe, T. H. (2007). Spatio-semantic coherence in the integration of 3D city models. *The ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXVI-2/C43*, 8.
Steuer, H., Machl, T., Sindram, M., Liebel, L., & Kolbe, T. H. (2015). Voluminator—approximating the volume of 3D buildings to overcome topological errors. In F. Bacao, M. Y. Santos, & M. Painho (Eds.), *AGILE 2015* (pp. 343–362). Cham: Springer.

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

van den Brink, L., Stoter, J., & Zlatanova, S. (2013b). UML-based approach to developing a CityGML application domain extension. *Transactions in GIS, 17*(6), 920–942.

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

Vanclooster, A., Van de Weghe, N., & De Maeyer, P. (2016). Integrating indoor and outdoor spaces for pedestrian navigation guidance: A review. *Transactions in GIS, 20*(4), 491–525.

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

Wrózyński, R., Sojka, M., & Pyszny, K. (2016). The application of GIS and 3D graphic software to visual impact assessment of wind turbines. *Renewable Energy, 96*, 625–635.

Zucker, G., Judex, F., Blöchle, M., Köstl, M., Widl, E., Hauer, S., Bres, A., & Zeilinger, J. (2016). A new method for optimizing operation of large neighborhoods of buildings using thermal simulation. *Energy and Buildings, 125*, 153–160.