Chapter 2
Representing and Exchanging 3D City Models with CityGML

Thomas H. Kolbe

Abstract. CityGML is an open data model and XML-based format for the representation and exchange of virtual 3D city models. It is based on the Geography Markup Language version 3.1.1 (GML3). Both CityGML and GML3 are international standards issued by the Open Geospatial Consortium (OGC). CityGML not only represents the shape and graphical appearance of city models but specifically addresses the object semantics and the representation of the thematic properties, taxonomies and aggregations. The paper gives an overview about CityGML, its modelling aspects and design decisions, recent applications, and its relation to other 3D standards like IFC, X3D, and KML.

2.1   Semantic 3D City Models

Virtual 3D city models have been used in the past mainly for the visualisation or graphical exploration of cityscapes. Nowadays, an increasing number of applications like environmental and training simulations, urban planning and facility management, disaster management and homeland security, and personal navigation require additional information about the city objects given in a standardised representation. Semantic 3D city models comprise besides the spatial and graphical aspects particularly the ontological structure including thematic classes, attributes, and their interrelationships. Objects are decomposed into parts due to logical criteria (and not due to graphical considerations!) which follow structures that are given or can be observed in the real world. For example, a building will be decomposed into different (main) building parts, if they have different roof types and their own entrances like a house and the garage.

The semantic modelling of cities requires the appropriate qualification of 3D data. This can be done by an automated process in some cases or by manual interpretation.

Institute for Geodesy and Geoinformation Science
Technische Universität Berlin
kolbe@igg.tu-berlin.de

Anyway, it increases the efforts needed to create and maintain the 3D city model. From an economic viewpoint, the semantic modelling of cities only makes sense, if the data (in particular the semantic information) can be used by different customers within multiple applications. This, however, would require to find a common information model over the different users and applications.

This is what CityGML is all about. The aim of the development of CityGML was to reach a common definition and understanding of the basic entities, attributes, and relations within a 3D city model. By providing a core model with entities which are relevant to many disciplines the city model can become a central information hub to which different applications can attach their domain specific information (see Fig. 2.1). Information exchange between different disciplines can then be aligned with the objects of the city model. Similar ideas were discussed in the towntology project [1] and the nD modelling project [2].
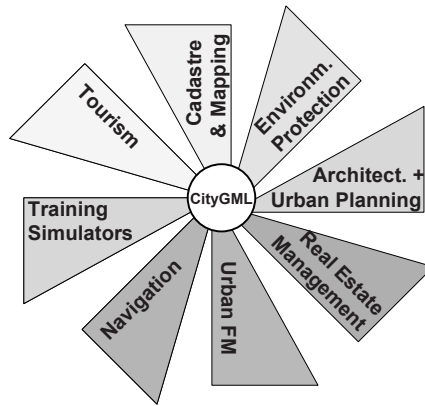

Fig. 2.1. The semantic 3D city model as the information hub which connects different disciplines by aligning information exchange with the city model entities.

2.2 CityGML and its Modelling Aspects

CityGML is an international standard for the representation and exchange of semantic 3D city and landscape models recently adopted by the Open Geospatial Consortium (OGC) [3, 4]. The data model behind CityGML is based on the ISO 19100 standards family and is implemented as an application schema for OGC's Geography Markup Language (GML 3.1.1) [5].

CityGML has been developed by the Special Interest Group 3D (SIG 3D) of the initiative Geodata Infrastructure North-Rhine Westphalia, Germany. The group consists of more than 70 members from industry, academia, and public administration. The different application backgrounds of the members include cadastre, urban planning, environmental and training simulations, civil engineering, architecture, computer graphics, geoinformation sciences, tourism, and telecommunication. Thus, a broad range of requirements, but also of expertise and experiences was brought into

the consensus process for the definition of the data model. This process took a considerable amount of time. For example, it took more than 2 years with a regular full day meeting every six weeks of about 20 people to define the building model. Afterwards, the model was presented and discussed on international conferences and within the OGC. Much of the feedback was considered and led to further improvements. Although this does not guarantee to achieve a perfect data model, there is a good chance of not having overseen important issues. In general, all modelling suggestions (feature classes and attributes) have been rated with respect to their multifunctional use and were only considered if they were needed in various application domains.

CityGML represents four different aspects of virtual 3D city models, i.e. semantics, geometry, topology, and appearance. Above, all objects can be represented in up to five different, well-defined levels-of-detail (LOD0 to LOD4 with increasing accuracy and structural complexity). The following subsections will discuss each of these aspects as well as spatio-semantic coherence and extensibility of CityGML in more detail.

Since version 1.0.0 CityGML is functionally partitioned into modules (cf. Fig. 2.2). The vertical modules provide the definitions of the different thematic models like building, relief (i.e. digital terrain model), city furniture, land use, water body, and transportation etc. The horizontal modules (CityGML core, appearance, and generics) define structures that are relevant or can be applied to all thematic modules. This structure allows for partial CityGML implementations on the one hand and easy extension by further thematic models on the other hand.
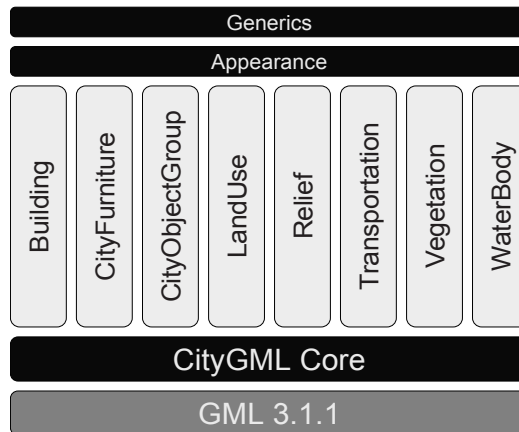

Fig. 2.2. Modularization of CityGML 1.0.0. Vertical modules contain the semantic modelling for different thematic areas.

2.2.1 Multi-scale representation

CityGML differentiates between five consecutive levels of detail (LOD), where objects become more detailed with increasing LOD regarding both geometry and thematic differentiation. Different LODs often arise from independent data collection processes and facilitate efficient visualization and data analysis. In a CityGML

dataset, the same object may be represented in different LODs simultaneously, enabling the analysis and visualization of the same object with regard to different degrees of resolution.

The coarsest level LOD0 is essentially a two and a half dimensional digital terrain model (DTM). LOD1 is the well-known blocks model, without any roof structures. A building in LOD2 has distinctive roof structures and larger building installations like balconies and stairs. LOD3 denotes architectural models with detailed wall and roof structures, doors, windows and bays. LOD4 completes a LOD3 model by adding interior structures like rooms, stairs, and furniture. Beyond buildings, the LOD concept applies to the other object classes as well. Fig 2.3 illustrates the five different LODs in CityGML.
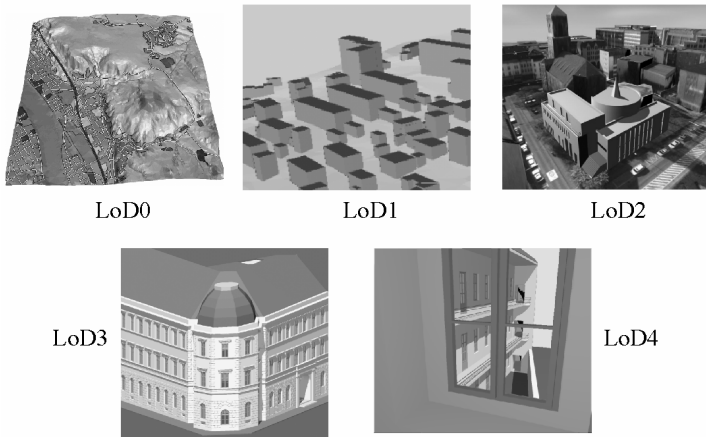

Fig. 2.3. Illustration of the five Levels-of-Detail (LOD) defined by CityGML.

The definition of the 5 LODs was based on previous work by different research groups on the usage of levels of detail in the representation of 3D city models [6, 7, 8]. LOD1 to LOD3 are inspired by similar classifications from [6] and [8]. LOD0 and LOD4 are originally defined by the SIG 3D.

The classification of models according to these 5 LODs may sometimes seem to be too coarse or restrictive (see [9]). However, the fixed number of LODs together with their specified information contents and accuracies establish quality classes. Just by naming the LOD of a CityGML dataset (or of objects within a CityGML dataset), one can quickly indicate the quality class for the 3D city model dataset. The LOD category makes datasets comparable and both the provider and customer / user get an idea of the data granularity, complexity, and accuracy.

2.2.2 Semantics

The semantic model of CityGML employs the ISO 19100 standards family framework for the modelling of geographic features. According to ISO 19109 geographic features are abstractions of real world objects [10]. They are modelled by classes which are formally specified using UML notation [11]. Geographic features may have

an arbitrary number of spatial and non-spatial attributes. Object oriented modelling principles can be applied in order to create specialisation and aggregation hierarchies.

CityGML provides class definitions, normative regulations, and explanations of the semantics for the most important geographic features within virtual 3D city models including buildings, DTMs, water bodies, vegetation, and city furniture.

Fig. 2.4 depicts the top level class hierarchy of CityGML. The base class of all thematic classes is the abstract class *CityObject*. It inherits the attributes *name*, *description*, and *gml:id* from the GML superclass *Feature* and provides the additional attributes *creationDate* and *terminationDate* in order to model different object states over time periods (please note, that attributes are not shown in Fig. 2.4 for the sake of readibility). Above, every CityObject may be linked to objects in other datasets or external databases by an arbitrary number of *ExternalReferences*. These references may point to other representations of an object in cadastres, enterprise resource management systems or other application specific datasets. This is especially useful to maintain back links to original objects from which the 3D models might have been derived. *CityObjects* can be aggregated to build a *CityModel* which is a subclass of the GML superclass *FeatureCollection*.
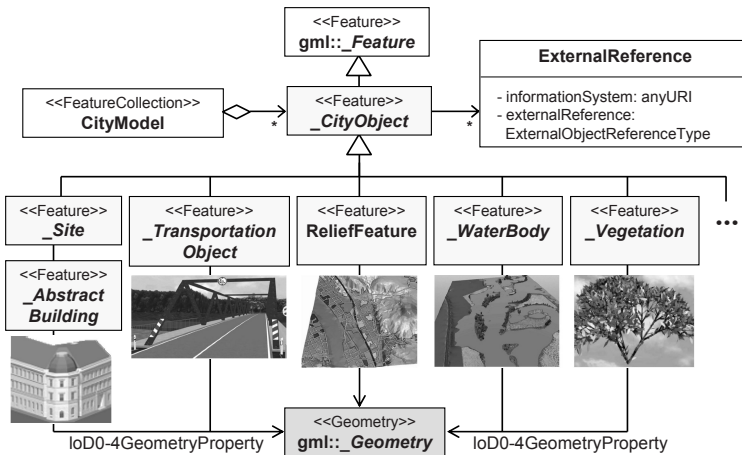

Fig. 2.4. UML diagram [11] of the top level class hierarchy of CityGML. All thematic objects are considered geographic features (following [10]) and their classes are derived from the abstract superclass CityObject. Attributes are omitted here for readability.

The subclasses of *CityObject* belong to different thematic areas and are defined within their corresponding modules (cf. Fig. 2.2). In the following, the building model will be discussed in more detail to illustrate the general modeling principles in CityGML. A complete description and normative reference is given in the CityGML specification document [3].

The pivotal class of the building model is *AbstractBuilding* from which the two non-abstract classes *Building* and *BuildingPart* are derived. These three classes follow the general composite design pattern [12]: a *Building* may contain *BuildingParts*, and as the latter class is also derived from *AbstractBuilding* a (recursive) aggregation hierarchy of arbitrary depth may be realized (see Fig. 2.5).

A *Building* or *BuildingPart* is described by optional attributes inherited from *AbstractBuilding* (not shown in Fig. 2.5): function, usage, and class, year of construction and demolition, roof type, measured height, number and individual heights of storeys above and below ground. *Building* and *BuildingParts* can be assigned multiple postal addresses.
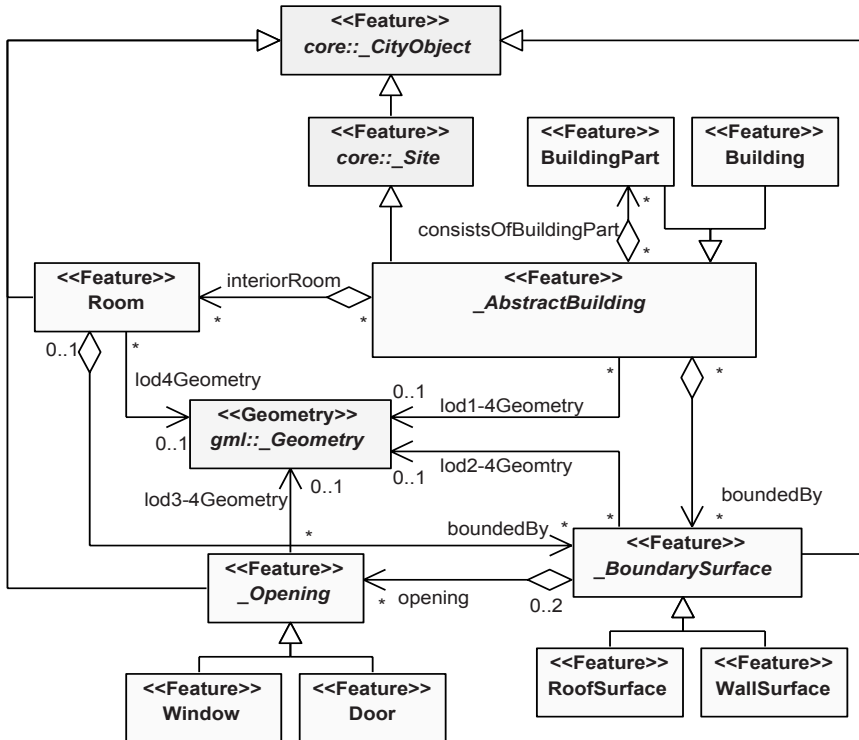

Fig. 2.5. UML diagram showing a simplified excerpt from the CityGML building model. Attributes are omitted for readability. Please note, that feature classes have multiple relations to geometry classes (e.g. *lod1Geometry*, *lod2Geometry* etc.).

Starting from LOD2 the boundary surfaces of *Buildings* and *BuildingParts* may be represented as semantic objects. *BoundarySurface* is the abstract superclass of these thematic objects having *RoofSurface*, *WallSurface*, *GroundSurface*, *ClosureSurface* etc. as subclasses (only the former two are shown in Fig. 2.5). As they are also derived from class *CityObject* they inherit its attributes and relations. In LOD3 and LOD4 openings in boundary surfaces may be represented by explicit thematic features of the classes *Window* or *Door*. Only in LOD4, buildings are allowed to have *Rooms*. Please note, that all feature classes have additional thematic attributes and further relations to different geometry classes not shown here.

The definition of this data model restricts the way buildings can be represented in a city model. In conjunction with the normative definitions of the feature classes this has an impact on the way how concrete data would have to be registered and structured. While this may be seen as restrictive concerning the freedom of modelling on

the one side, it establishes a bindingness or liability of the producer with respect to potential users or customers on the other side. Users and their applications can expect buildings to be structured in a well-defined way. This makes it profitable to develop new applications that exploit the semantic information and structure of the *CityObjects*.

2.2.3 Geometry

CityGML uses a subset of the GML3 geometry model which is an implementation of the ISO 19107 standard [13]. According to ISO 19107 and GML3, geometries of geographic features are represented as objects having an identity and further (geometric) substructures. GML3 provides classes for 0D to 3D geometric primitives, 1D-3D composite geometries, and 0D-3D geometry aggregates. Composite geometries like *CompositeSurface* must be topologically connected and isomorphic to a primitive of the same dimension (e.g. *Surface*). Aggregate geometries like *MultiSurface* or *MultiSolid* do not impose topological constraints and thus their components can also permeate each other or be disjoint.

Volumetric geometries are modeled according to the well-known Boundary Representation (BRep, see [14]) where each solid is bounded by a closed surface (typically a *CompositeSurface*). In contrast to scene graphs or Constructive Solid Geometry (CSG) [14], all coordinates must belong to a world coordinate reference system (CRS) and no local transformations are allowed. 3D CRS refer to geographic or projected world reference systems. Above, GML3 also supports composite 2D+1D CRS with different reference systems for planimetry (x, y) and vertical coordinates (z). In general, the CRS define the frame for real world coordinates and they often differ for different countries, states, or even regions. However, the explicit association of each geometry with a CRS facilitates the integration of datasets by the application of the respective geodetic datum and coordinate transformations.

The advantage of having only absolute coordinates is that each geometry object belongs to exactly one fixed place in space. This allows to easily create and maintain spatial indexes in geodatabases or geoinformation systems. In fact, commercial and OpenSource RDBMS like Oracle Spatial or PostGIS both natively support (most of) the GML geometry model and different CRS. A big disadvantage is, however, that shape definitions cannot be reused like with CSG and scene graphs. For example, if a city model would contain 100 street-lamps or trees, 100 different, but equally shaped geometries would have to be created. In order to overcome this drawback, CityGML provides an extension to the GML3 geometry model called *ImplicitGeometry*. *ImplicitGeometries* refer to a shape geometry in a local coordinate system (to be reusable) and additionally provide a transformation matrix and an anchor point in a world CRS. The real world coordinates are computed by the multiplication of the local shape coordinates with the transformation matrix and then adding the coordinates of the anchor point. *ImplicitGeometries* are called like this, because the anchor point locates a place on earth where the associated shape 'unfolds' and thus implicitly describes the occupied space in real world. Features with implicit geometries can be retrieved from a GIS or spatial database by spatial selections on the anchor points.

In order to ensure a broad system support, CityGML is also restricted to non-curved geometries, as these often cannot be handled by GIS or spatial database

management systems. Therefore, curved surfaces of real world objects must be approximated by facetted surfaces of planar patches.

According to ISO 19109 geographic features can be assigned more than one spatial property. This is generally being used in CityGML where most feature classes like *AbstractBuilding* or *Room* are assigned individual geometries for the different LODs by multiple associations (e.g. *lod1Solid*, *lod2Solid*, *lod3Solid*; c.f. Fig. 2.5) to the same geometry class (e.g. *Solid*).

For a given city model it is often not known, whether the data is topologically sound. For example, the boundary surfaces of buildings may not be closed or contain additional surfaces, which do not belong to the boundary of the volume (e.g. a roof overhang). For this reason, the geometry of nearly all thematic features can be represented by either *MultiSurfaces* or *Solids*. While *MultiSurfaces* are an unconstrained collection of surfaces in 3D space, *Solids* must be bounded by a closed (composite) surface (c.f. [3]).

2.2.4 Topology

For many applications topological correctness of the object geometries is crucial. For example, the surfaces bounding a building must be closed in order to be able to compute its volume. Also the solids of adjacent objects like *BuildingParts* must touch but their interiors are not allowed to permeate each other, because space can only be occupied by one physical object. In the application field of indoor navigation, rooms should be topologically connected as this facilitates the computation of a connectivity graph which then can be transformed to a 3D geometry network for route planning [15, 16].

In the past, different frameworks for the representation of 3D topology have been presented [17, 18, 19]. Nevertheless, most of them propose a geometric-topological structure where coordinates are stored only within the nodes or the points associated with the nodes. Higher dimensional primitives are then constructed by connecting primitives of lower dimensions. This way, nodes, edges, and faces can be shared by different primitives in order to construct complex shapes. By sharing primitives an explicit connection between higher dimensional shapes can be established and gaps or permeations can be avoided. This is illustrated in Fig. 2.6.

The topology model of ISO 19107 and GML3 also follows the line of full decomposition of n-dimensional topological primitives into $(n\text{-}1)$-dimensional primitives, which again are decomposed down to the level of nodes (0D). Please note that each primitive becomes an object (with ID). Furthermore, the topological representation in GML would require appropriate topological properties in the CityGML application schema. That means, that besides the geometry properties like *lod1Solid* to *lod4Solid* of class *Building* also corresponding topology properties like *lod1TopoSolid* to *lod4TopoSolid* would have to be added to the data model.
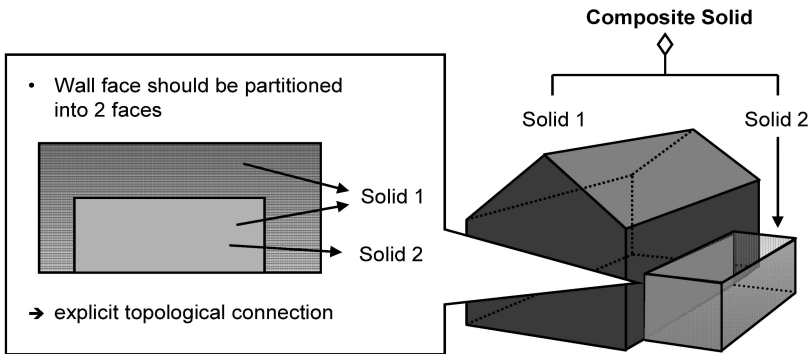
Fig. 2.6. A building with a connected garage. The topological adjacency between the solids representing the building extent and the garage extent can be made explicit by sharing the common wall surface.

As CityGML should be able to represent purely geometric models on the one hand and geometric-topological models on the other hand usage of the GML3 topology model would significantly increase the complexities of both the data model and concrete instances, i.e. datasets. If we take a closer look at Fig. 2.6 then we can also see, that it is not necessary to decompose the 3D building shape down to the level of nodes to establish the topological connection between the house and the garage. Indeed, it is sufficient to simply reuse the common wall surface of the house and the garage. This can be implemented in GML by providing the definition of the surface geometry inline within the specification of the solid geometry (bounded by a composite surface) of either the house or the building. In the representation of the other solid this surface is then included by reference (and not by value) which creates the connection between both solids. The following Listing shows an excerpt of a CityGML file which illustrates the referencing of a polygon 2.1 using the XLink mechanism [5]:

```
<bldg:BuildingPart>
  ...
  <bldg:lod2Solid>
    ...
    <gml:surfaceMember>
      <gml:Polygon gml:id="wallSurface4711">
        <gml:exterior>
          <gml:LinearRing>
            <gml:pos srsDimension="3">32.0 31.0 2.5</gml:pos>
            ...
          </gml:LinearRing>
        </gml:exterior>
      </gml:Polygon>
    </gml:surfaceMember>
  </bldg:lod2Solid>
  ...
</bldg:BuildingPart>
...
<bldg:BuildingPart>
  ...
```

```
    <bldg:lod2Solid>
        ...
        <gml:surfaceMember xlink:href="#wallSurface4711"/>
        ...
    </bldg:lod2Solid>
    ...
</bldg:BuildingPart>
```

Listing 2.1. Establishing a topological connection between two solids by referring to the same polygon within the outer shells (composite surfaces) of the two solids.

2.2.5 Appearance

Information about a surface's appearance, i.e. observable properties of the surface, are considered an integral part of virtual 3D city models in addition to semantics and spatial properties. *Appearances* capture characteristics of surfaces as they appear to specific sensors like RGB or infrared (IR) cameras. Appearances are not limited to visual data but represent arbitrary categories called *themes* such as infrared radiation, noise or sunlight emission, or even earthquake-induced structural stress. Consequently, appearances can serve as input for both visualisation and analysis tasks.

Each surface is allowed to have assigned multiple appearances (e.g. one RGB texture for a building facade recorded in summer and another one from winter). All appearances must be assigned to themes like e.g. the "spring RGB theme" or the "winter IR theme". These themes are useful if one wants to visualize a 3D scene differently for winter or summer time. Switching of themes in a viewer client would mean a replacement of the surface materials and textures. By using themes, also multi-texturing can be supported by CityGML. One example could be the precomputation of shadow maps for each surface which would all be assigned to a theme that might be called "shadows". The shadow textures could then be mapped together with the RGB texture images to the corresponding surfaces.

The appearance data is given by texture images or by material definitions (adopted from X3D [20] and COLLADA). CityGML provides different ways how the raster data is mapped to the surfaces: *GeoreferencedTextures* can be used to describe the appearance of non-vertical surfaces. This way, e.g. an aerial orthophoto can be used together with its georeferencing information to easily describe the appearances of an arbitrary number of e.g. roof surfaces. It is up to a viewer to compute the proper extents of the textures from the 2D footprint of the corresponding 3D surface.

ParameterizedTextures use either explicit texture coordinates for each surface they should be mapped onto or a projection matrix. This can be computed from the exterior camera orientation and focal length associated with the texture image. 3D surfaces can simply refer to the picture in which their surface is best observed. The rest can be computed from these information by a viewer (or a 3D portrayal web service like the Web 3D Service [21]).

2.2.6   Spatio-semantic coherence

The ISO 19109 standard for the modelling of geographic features implies a dual structure of thematic object classes (the feature classes) on the one side and spatial properties (using the geometry and topology classes of ISO 19107) on the other side. In CityGML, objects like buildings may be decomposed both regarding their thematic structure into *BuildingParts*, *Rooms*, *WallSurfaces*, etc. and regarding their geometric structure into *CompositeSolids* consisting of *Solids* and these again be bounded by *CompositeSurfaces*.

In general, it is desirable to have a coherent thematic and spatial structuring of the objects. This means, that each complex thematic object is associated with a complex geometry object and each of the thematic components are also assigned to geometric components (being substructures of the original complex geometry). For example, a building may be assigned a solid geometry in LOD2. If the building is further decomposed into thematic surfaces like *WallSurface*, *RoofSurface* etc. their associated geometries should refer to those surface geometry objects which are part of the outer shell of the building's solid geometry. A fully coherent dataset has the advantage that each geometry object 'knows' what thematic role it plays and that each thematic feature 'knows' its location and spatial extent.

CityGML is flexible about the spatio-semantic coherence. It is possible to represent fully coherent datasets but also those types of data, in which either the semantic objects or the geometry objects are deeper structured. This can occur, for example, in a situation where the building geometry is reconstructed or acquired as a complex aggregation hierarchy which, however, is assigned to a simple Building object without any further thematic decomposition. A detailed discussion of spatio-semantic coherence and the typical cases for 3D city models is given in [22].

2.2.7   Extensibility

CityGML has been designed as a universal topographic information model that defines feature classes and attributes which are useful for a broad range of applications. However, in practical applications it is often necessary to store and exchange extra attributes or even 3D objects which do not belong to any of the predefined classes.

For these cases, CityGML generally provides two different ways of extension. The first is the usage of generic city objects and generic attributes, both defined within the module 'generics' (Fig. 2.2). Any *CityObject* may have an arbitrary number of additional generic attributes. For each generic attribute of an object the name, type, and value has to be given within the CityGML dataset. Supported data types are string, integer, real, date, and URI. *GenericCityObjects* may be assigned arbitrary geometries or *ImplicitGeometries* for each LOD. As they are derived from *CityObject* they may also be assigned generic attributes.

The second concept for extending CityGML are the so-called Application Domain Extensions (ADE). An ADE specifies systematic extensions of the CityGML data model, see Fig. 2.7. These comprise the introduction of new properties to existing CityGML classes like e.g. the number of habitants of a building or the definition of new feature classes. The difference between ADEs and generic objects and attributes is, that an ADE has to be defined within an extra XML schema definition file with its

own namespace. This file has to explicitly import the XML Schema definition of the extended CityGML modules. The advantage of this approach is that the extension is formally specified. Extended CityGML instance documents can be validated against the CityGML and the respective ADE schema. ADEs can be defined (and even standardised) by information communities which are interested in specific application fields. More than one ADE can be actively used in the same dataset.
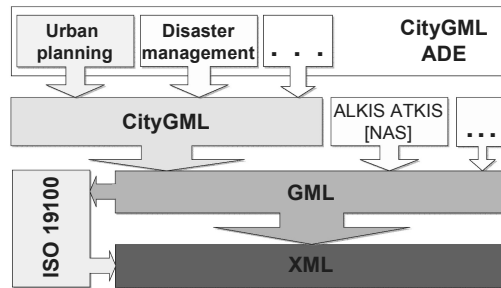

Fig. 2.7. CityGML Application Domain Extensions are XML schema definitions based on the CityGML schema. They extend CityGML by new feature classes and additional attributes for existing classes. CityGML is an application schema of GML (like others, e.g. the German national cadastre standard ALKIS).

2.3   Relation to other 3D Standards

Virtual 3D city models are not only investigated in the context of geoinformation systems. The field of architecture, engineering, construction, and facility management (AEC/FM) as well as the field of computer graphics provide their own standards for the representation and exchange of 3D models. The following subsections discuss the relations of CityGML to these standards.

2.3.1   Building Information Modelling / IFC

Building Information Modelling (BIM) means the semantic modelling of objects and processes in the the field of AEC/FM and CAAD. Like in CityGML, thematic objects are represented with their 3D spatial properties and interrelationships. Data are typically exchanged using the Industry Foundation Classes (IFC, see [23]), an ISO standard describing a product model and data exchange format for the built-up environment developed by the International Alliance for Interoperability (IAI).

   IFC provides a very detailed semantic model for 3D building representations using constructive elements like beams, walls etc. Like in GML, IFC geometries are spatial properties of semantic objects. IFC has a very flexible geometry model (CSG, BRep, and Sweep representations), but does not provide support for CRS. Thus, georeferencing is not possible (yet).

Since the scope of IFC is restricted to buildings and sites, no topographic feature classes like terrain, vegetation, water bodies etc. are included. These would have to be modelled and exchanged in IFC as generic objects (i.e. IFC_Proxy).

IFC is a semantic model like CityGML, but with a different scope and at a different scale. IFC models can be converted to CityGML in different LODs preserving most of their semantic information [24]. CSG and sweep geometries have to be converted to BRep though. This way, IFC objects can be brought into the context of a city model within a GIS or spatial database and could then become subject to spatial and thematic queries (see [25]).

The derivation of IFC objects from CityGML data is a topic of future research, because from CityGML's surface models 3D volumetric components would have to be reconstructed. However, CityGML may be a good intermediate step in the (semi-)automatic acquisition of IFC models, because the CityGML object classes like *Wall-Surface*, *RoofSurface* etc. are much closer to photogrammetric observations or geodetic measurements (including laser scanning) than the component models of IFC (cf. [26]).

2.3.2 3D Computer Graphics and Geovisualisation / X3D & KML

CityGML is complementary to 3D computer graphics standards like X3D [20], VRML, or COLLADA and geovisualization standards like KML [27].

These standards address the way how to efficiently and nicely visualize 3D models and how to interact with them. Both in X3D and KML georeferenced coordinates can be used. Although it is in principle possible to also exchange semantic information using X3D (see [28]) or KML, both specifications do not standardise the way how to represent complex geographic features and their interrelationships.

CityGML should be considered a rich source format from which X3D or KML can easily be derived. CityGML is not optimized with respect to efficient visualization. However, the semantic information given by the explicit association of CityGML objects to thematic classes like buildings, doors, plants, and the provision of thematic attributes can be exploited to filter objects and to create 3D graphical shapes, appearance properties and materials accordingly.

2.4 Applications

CityGML is being employed already in different application areas. For example, the mapping of environmental noise pollution for the whole state of North-Rhine Westphalia in Germany is done based on CityGML data for about 8.6 million buildings in LOD1, road and railway networks in LOD0, and the digital terrain model in LOD1. In order to facilitate noise immission simulation, a specific CityGML Noise ADE was created which extends CityGML feature classes by noise data like the number of cars during daytime and nighttime or the type of roadbed and road material. Also a new feature type representing noise protection walls was modelled. The complete dataset comprises buildings, roads, railways, DTM, and noise barriers with most of them including noise specific extensions. It is forwarded to the noise simulation application as one CityGML file, accessed via a Web Feature Service over the Internet [29].

Municipalities nowadays create and exchange their 3D city models according to the CityGML standard. They use their models mainly for urban planning, city business development, and tourism. For example, the cities of Berlin [30], Stuttgart, Cologne, Munich, Düsseldorf, Recklinghausen, Frankfurt, and many more are providing CityGML compliant 3D city models. The VEPS 3D project has investigated different ways to employ 3D city models (and CityGML) in urban planning and public participation [31].

Within the OGC Open Web Services Testbed no. 4 (OWS-4), CityGML was employed together with IFC in a homeland security scenario. Here it was shown how an emergency operator can find a suitable location and building for a field hospital after the explosion of a dirty bomb [25]. Generally, CityGML can provide important information for disaster management [32]. This includes emergency route planning and indoor navigation [15, 33] as well as the extension of CityGML by dynamic water surfaces within a specific ADE for flooding scenarios [34].

CityGML may also become an important data source for training simulators. In [35] the requirements on 3D city models for training simulators are discussed in general, and in [36] concerning emergency car-driver training in particular.

The field of 3D cartography is not well developed yet. One important reason has been the lack of semantic 3D models in the past, because class information and thematic attributes are prerequisites for cartographic styling, symbolisation and generalisation. CityGML provides rich data that can be exploited for non-photorealistic rendering [37]. User defined styling may be specified using an appropriate 3D extension of the OGC Styled Layer Descriptors and Symbology Encoding (SLD/SE) standards [16].

2.5 Conclusions

CityGML is both a semantic model (specified by a formal data model) and an exchange format for virtual 3D city and landscape models. Rules for the acquisition and structuring of urban objects follow implicitly from this semantic modelling. The long discussions with the big group of people during the development of CityGML is (of course) not a guarantee, but a reasonable basis for the definition of these structures. It is now a topic of future work to bring CityGML to a wider adoption and discuss and learn from the experiences of the much broader user base.

The data model of CityGML balances between strictness and genericity. For this purpose it consists of three main parts: 1) the core thematic model with well-defined LODs, classes, spatial and thematic attributes, and relations; 2) *GenericCityObjects* and generic attributes allow the extension of CityGML data 'on-the-fly'; and 3) ADEs facilitate the systematic extension for specific application domains.

CityGML also balances between simple objects and objects with complex thematic and spatial structures. Data is given high flexibility to grow with respect to their spatial and semantic structuring as well as their topological correctness in different stages of data acquisition or along a city model processing chain (see [26]).

Finally, CityGML is complementary to visualization standards like X3D or KML. While these address presentation, behaviour, and interaction of 3D models, CityGML is focused on the exchange of the underlying urban information behind the 3D objects.

It also complements building information models (BIM) and the IFC standard on a smaller scale and by topographic feature classes and georeferencing.

2.6 Acknowledgements

References

1. Teller J. , Keita A. K. , Roussey C. , Laurini R., 2005. Urban Ontologies for an improved communication in urban civil engineering projects. In: Proc. of the Int. Conference on Spatial Analysis and GEOmatics, Research & Developments, SAGEO 2005 Avignon, France, June, 20th-23rd.
2. Hamilton, A., Wang, H., Tanyer, A. M., Arayici, Y., Zhang, X., Song, Y., 2005. Urban information model for city planning, ITcon Vol. 10, Special Issue From 3D to nD modelling
3. Gröger, G., Kolbe, T.H., Czerwinski, A., Nagel, C., 2008. OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Version 1.0.0, OGC Doc. No. 08-007r1, Open Geospatial Consortium
4. CityGML Homepage: http://www.citygml.org
5. Cox, S., Daisy, P., Lake, R., Portele, C., Whiteside, A., 2004. OpenGIS Geography Markup Language (GML3), Implementation Specification Version 3.1.0, OGC Doc. No. 03-105r1.
6. Köninger, A., Bartel, S., 1998. 3D-GIS for Urban Purposes. Geoinformatica, 2(1), March 1998
7. Coors, V., Flick, S., 1998. Integrating Levels of Detail in a Web-based 3D-GIS, In: Proc. of ACM GIS '98 in Washington D.C., USA, ACM Press
8. Schilcher, M., Guo, Z., Klaus, M., Roschlaub, R., 1999. Aufbau von 3D-Stadtmodellen auf der Basis von 2D-GIS (in German only). Zeitschrift für Photogrammetrie und Fernerkundung (ZPF), Vol. 67, No. 3
9. Döllner, J., Buchholz, H., 2005. Continuous level-of-detail modeling of buildings in 3D city models. In: Proc. of ACM GIS 2005, ACM Press
10. ISO 19109, 2005. Geographic information — Rules for application schema.
11. Booch, G., Rumbaugh., J. & Jacobson, I. 1997. Unified Modeling Language User Guide. Addison-Wesley.
12. Gamma, E., Helm, R., Johnson, R.E., 1995. Design Patterns. Elements of Reusable Object-Oriented Software, Addison-Wesley Longman
13. Herring, J., 2001. The OpenGIS Abstract Specification, Topic 1: Feature Geometry (ISO 19107 Spatial Schema). OGC Document Number 01-101.

14. Foley, J., van Dam, A,. Feiner, S. & Hughes, J. 1995. Computer Graphics: Principles and Practice. Addison Wesley, 2nd Ed.
15. Lee, J., Zlatanova, S., 2008. A 3D data model and topological analyses for emergency response in urban areas. In: Zlatanova, Li (eds.), Geospatial Information Technology for Emergency Response, Taylor & Francis
16. Neubauer, S., Zipf, A., 2007. Suggestions for Extending the OGC Styled Layer Descriptor (SLD) Specification into 3D – Towards Visualization Rules for 3D City Models. In: Proc. of 26th Urban and Regional Data Management, October 2007 in Stuttgart, Taylor & Francis
17. Molenaar, M., 1992. A topology for 3D vector maps. ITC Journal 1992-1.
18. Zlatanova, S. 2000. 3D GIS for Urban Development. PhD Thesis, ITC Dissertation Series No. 69, The International Institute for Aerospace Survey and Earth Sciences, The Netherlands
19. Oosterom, P., Stoter, J., Quak, W. & Zlatanova, S. 2002. The balance between geometry and topology. In: Proc. of 10th Int. Symp. SDH 2002, Springer
20. ISO 19775, 2005. X3D International Specification Standards, Web 3D Consortium.
21. Quadt, U., Kolbe, T. H., 2005. Web 3D Service Implementation Specification, OGC Discussion Paper, Doc. No. 05-019
22. Stadler, A., Kolbe, T.H., 2007. Spatio-semantic Coherence in the Integration of 3D City Models. In: Proceedings of the 5th International Symposium on Spatial Data Quality, Enschede 2007.
23. Liebich, T., Adachi, Y., Forester, J., Hyvarinen, J., Karstila, K., Reed, K., Richter, S., Wix, J., 2007. Industry Foundation Classes IFC2x Edition 3 – Technical Corrigendum 1, International Alliance for Interoperability, http://www.iai-international.org.
24. Benner, J., Geiger, A., Leinemann, K., 2005. Flexible generation of semantic 3D building models. In: Gröger, Kolbe (eds.), Intern. ISPRS / EuroSDR / DGPF-Workshop on Next Generation 3D City Models. Bonn, Germany, EuroSDR Publication No. 49.
25. Lapierre, A., Cote, P., 2007. Using Open Web Services for urban data management: A testbed resulting from an OGC initiative for offering standard CAD/GIS/BIM services. In: Coors, V., Rumor, M., Fendel, E. M., Zlatanova S. (eds): Urban and Regional Data Management. Proceedings of the 26th UDMS, October 10-12, 2007, Stuttgart, Taylor & Francis
26. Kolbe, T. H., Nagel, C., Stadler, A., 2008. CityGML – A Framework for the Representation of 3D City Models from Geometry Acquisition to full Semantic Qualification. In: Proc. of ISPRS Congress 2008 in Bejing, China
27. Wilson, T., 2008. OGC KML. OGC Encoding Standard, Version 2.2.0, OGC Doc. No. 07-147r2, Open Geospatial Consortium
28. Pittarello, F., De Faveri, A., 2006. Semantic Description of 3D Environments: a Proposal Based on Web Standards. In: Proceedings of the Web 3D Symposium 2006 in Columbia, Maryland, 18-21 April 2006, ACM Press.
29. Czerwinski, A., Kolbe, T. H., Plümer, L., Stöcker-Meier, E., 2006. Spatial data infrastructure techniques for flexible noise mapping strategies. In: Tochtermann, Scharl (eds.), Proc. of the 20th International Conference on Environmental Informatics - Managing Environmental Knowledge. Graz 2006.
30. Döllner, J., Kolbe, T. H., Liecke, F., Sgouros, T., Teichmann, K., 2006. The Virtual 3D City Model of Berlin - Managing, Integrating, and Communicating Com-

plex Urban Information. In: Proceedings of the 25th Urban Data Management Symposium UDMS 2006 in Aalborg, DK, May 15-17. 2006.

31. VEPS 3D Homepage: http://www.veps3d.org

32. Kolbe, T. H., Gröger, G., Plümer, L., 2008. CityGML – 3D City Models and their Potential for Emergency Response. In: Zlatanova, Li (eds.), Geospatial Information Technology for Emergency Response, Taylor & Francis

33. Mäs, S., Reinhardt, W., Wang, F., 2006. Conception of a 3D Geodata Web Service for the Support of Indoor Navigation with GNSS. In: Proc. of 3D GeoInfo 2006 in Kuala Lumpur, LNG&C, Springer

34. Schulte, C., Coors, V., 2008. Development of a CityGML ADE for dynamic 3D flood information. Joint ISCRAM-CHINA and GI4DM Conference on Information Systems for Crisis Management 2008 in Harbin, China

35. Bildstein, F., 2005. 3D City Models for Simulation and Training – Requirements on Next Generation 3D City Models. In: Proc. of the Int. ISPRS Workshop on Next Generation 3D City Models in Bonn, Germany. EuroSDR publication no. 49

36. Randt, B., Bildstein, F., Kolbe, T. H., 2007. Use of Virtual 3D Landscapes for Emergency Driver Training. In: Proc. of the Int. Conference on Visual Simulation IMAGE 2007 in Scottsdale, Arizona 2007

37. Döllner, J., Walther, M., 2003. Real-Time Expressive Rendering of City Models. In: Proc. of the 7th International Conference on Information Visualization in London, IEEE Press