Towards defining a framework for automatic generation of buildings in CityGML using Building Information Models

Umit Isikdag and Sisi Zlatanova

Abstract

Increased demand for tools that allow merging of Building Information Models with GIS models is observed in the last several years. Professionals from both domains are looking for solutions to seamlessly integrate such models for various purposes such as, building and construction analysis, urban planning, tourism, cadastre, homeland security, etc. Researchers suggested that the best approach for such integration is harmonised semantics, which will allow formal mappings between the design (BIM) and real world (GIS) models. Although many geometric models have been developed in both domains, the number of semantic models is relatively few. Two most prominent semantic models in the design and real worlds are currently IFC and CityGML. Several studies demonstrate the transfer of information from IFC models into the CityGML but the literature is lacking a formal and descriptive framework for automatic generation of buildings in CityGML using the IFC models. This paper presents preliminary ideas for defining a semantic mapping, which will allow automatic transformations between the two models.

1 Introduction

Several ways and methods exist to acquire geometrical and semantic information from building models and to represent this information within the 3D geospatial environment. Detailed geometric information about buildings can be obtained from digital building models (i.e. CAD drawings), by measuring existing buildings using laser scanning methods, surveying or photogrammetric techniques. Semantic information can be acquired by, querying various databases where information about a building is stored or using the semantic information stored in Building Information Models (BIM). This -acquired information- can later be transformed into the geospatial environment and stored in form of a geometrical and/or topological geospatial model, in a physical file or in a geospatial database. In this paper we concentrate on transformation of information from BIM (digital models) to the 3D geospatial models (and specifically to CityGML).

Since many years, the AEC industry is benefiting from Computer Aided Design (CAD) systems for creating digital building models. CAD systems are developed with the aim of modelling objects for assisting production/construction process. Models defined with CAD systems usually exist before the final product (or building), and are designed for representing the maximum level of detail in terms of geometry of the model. In CAD systems one or a group of building elements can be modelled in two or three dimensions, within a complex geometrical representation. The element geometries can be modelled by using CSG, Sweeping or BRep methods and geometries can contain curves, splines and surface patches (Lattuada, 2006). Until recent years, most of the CAD models created were in two dimensions, were not object oriented and rich in semantic information, and spatial relations between the building elements were not preserved within the models. Storing semantic information was not the focus of CAD systems, but with the advent of Product Lifecycle Management (PLM) in production management and Building Information Modelling (BIM) in the AEC industry this situation is now radically changing. Several Building Information Models have been reported in the literature, most commonly known ones are CIS/2 (CIS2,2007) and IFC models (IFC, 2008), which offer means to define objects with geometry and semantics.

On the other hand, geospatial information systems are developed to represent objects that already exist around us. They are defined for representing the objects in a simplified but efficient way (specifically in terms of geometry). Geospatial models represent large number of objects mostly in 2D/2,5D with more simple geometric representations and building geometries are represented by BRep and Sweeping methods. In addition the geometries are mostly created with straight lines (i.e. consisting of polygons, polyhedrons). Attribute information is an important aspect of geospatial information models and usually stored in databases. As Pu and Zlatanova (2006), Zlatanova *et al* (2006), Breuning and Zlatanova (2006) indicated that CAD software supports a broad range of 3D primitives and free-form curves, while these primitives and free-form curves are not present in the

GIS world. Recently 3D Information models were developed for representing the real world objects, i.e. CityGML (CityGML 2008, Kolbe *et al* 2003), in fact the geometry is still limited to simple representations.

Many researchers have investigated the differences and the similarities between CAD and GIS and suggested approaches for transforming information from one to other. As mentioned by van Oosterom *et al* (2006) the lack of object definitions in the CAD files, different scale representations, transformation of the local (CAD) coordinates into a geospatial coordinate system, the existence of parametric shapes in CAD files that can not be converted into GIS objects, and different levels of detail between CAD models and their representation in the geospatial environment appeared as main barriers that prevent CAD-GIS data transformation. Van Oosterom *et a*l (2006) draws an attention to the need of integrated geometric models and harmonised semantics between two domains in order to tackle with the interoperability problems between AEC and geospatial information domains. Zlatanova *et al* (2006) pointed out that developing uniform data types for both CAD and geospatial information models would eliminate the need for conversion between different formats.

Several attempts have been made to simplify BIM models and integrate them with GIS. For example in a recent effort, Isikdag (2006) demonstrated the transfer of information from an industry standard BIM (IFC) to the (ESRI) Shapefiles and Geodatabases. In parallel, commercial software for conversion from IFC to CityGML and vice versa is in development (i.e. IfcExplorer, 2008; Safe Software, 2008). OGC has completed tests on the integration of CityGML and IFC models in OWS-4 testbed (Lapierre, 2008). However a formal framework for strict (semantic and geometry) conversion is not available yet.

This paper suggests that information from (semantic) design and real-world models can automatically be exchanged if a formal framework can be made available. Following the background sections, we present our preliminary ideas for establishing framework for transformation of information between IFC and CityGML. Section 2 presents an introduction to Building Information Modelling, the following (third) section presents the background on the representation of buildings within 3D geospatial models. The 4th section presents an overview on information mapping needs which will serve as a basis when establishing the framework for transformation of information between IFC and CityGML models.

2 Building Information Modeling

The fragment nature of the AEC industry has resulted in significant barriers to communication between the various stakeholders, which in turn has significantly affected the efficiency and performance of the industry. Gallaher *et al.* (2004) indicated that, US$15.8B is lost annually in the U.S Capital Facilities Industry due to the lack of interoperability. In recent years, Building Information Modelling has become an active research area in order to tackle the problems related to information integration and interoperability. Today, Building Information Models (BIMs) are promising to be the facilitators of integration, interoperability and collaboration in the future of the construction industry. According to NBIMS (2006) a BIM is a computable representation of all the physical and functional characteristics of a building and its related project/life-cycle information, which is intended to be a repository of information for the building owner/operator to use and maintain throughout the life-cycle of a building.

2.1 Industry Foundation Classes (IFC)

Today, the current key efforts in the area of BIM are the Industry Foundation Classes (IFC) and the CIMSteel Integration Standards 2(CIS/2). The distinctive characteristics of the BIMs can be identified as follows:

1. *Object Oriented:* most of the BIMs are defined in an object-oriented nature.
2. *Data-rich / Comprehensive:* BIMs are data rich and comprehensive as they cover all physical and functional characteristics of the building.
3. *Three dimensional:* BIMs always represent the geometry of the building in three dimensions.
4. *Spatially-related:* Spatial relationships between building elements are maintained in the BIMs in a hierarchical manner.
5. *Rich in semantics:* BIMs maintain a high amount of semantic (functional) information about the building elements.
6. *Supports view generation:* The model views are subsets or snapshots of the model that can be generated from the base information model. BIMs therefore support view generation.

Today IFC is seen as being the strongest player of the BIM world. IFC is the effort of IAI/buildingSMART whose goal is to specify a common language for technology to improve the communication, productivity, delivery time, cost, and quality throughout the design, construction and maintenance life cycle of buildings. Each specification (called a ‘class’) is used to describe a range of things that have common characteristics. These

IFC-based objects aim to allow AEC/FM professionals to share a project model, while allowing each profession to define its own view of the objects contained within the model. In 2005, IFC became an ISO Publicly Available Specification (as ISO 16739).

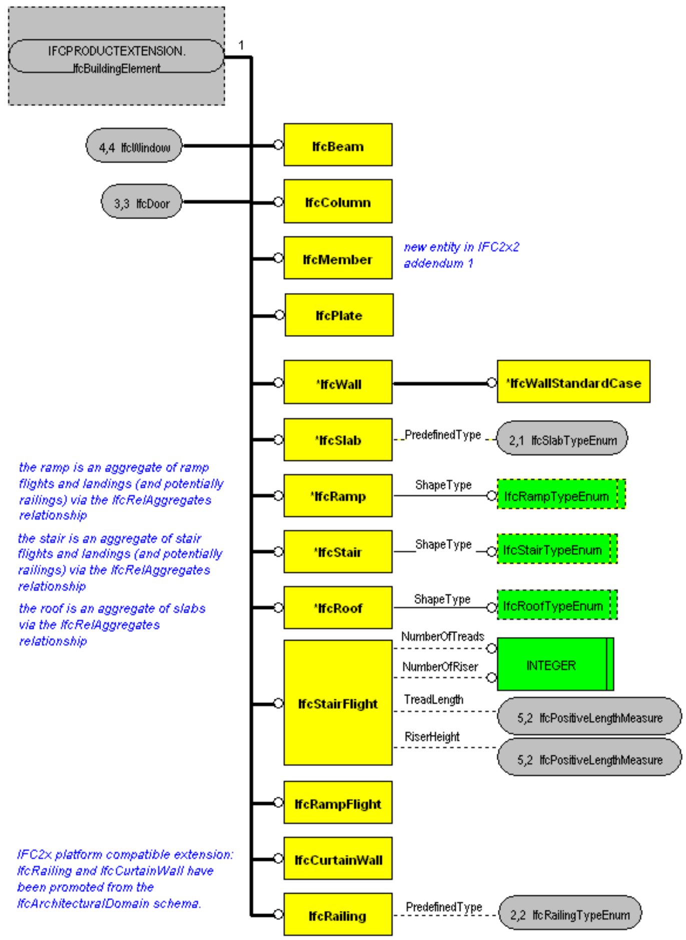

Fig.1 EXPRESS-G Representation of IFCSHAREDBUILDINGELEMENTS data model

In IFC model geometries of the building elements are always represented in 3D using one of/or combination of CSG, Sweeping and BRep methods. IFC model usually represents a building element by multiple geometric representations (i.e. a Sweeping Representation and a BRep). The spatial relations between the building elements are preserved within the models' spatial structure. The information about the geographic location of the building is stored as an attribute of *IfcSite* object of the model. The main structure of the building is represented within IFCSHAREDBUILDINGELEMENTS data model. This data model defines the entities for representing the basic components of the building. Each entity in this data model is a subtype of *IfcBuildingElement* entity. The EXPRESS-G Representation of IFCSHAREDBUILDING ELEMENTS data model is shown in Fig.1.

2.2 Model Views

In order to support several phases and the stakeholders of the construction life-cycle several views of the BIM needs to be generated (Eastman and Jeng, 1999). The BIM model views are generated by a declaration of a model that is a subset of another model, or by declaration of a model that is derivable from a BIM (in this case the view should not be a superset of a the BIM).These views can be generated from files or databases by using application interfaces and web interfaces. These views can either be transient or persistent depending on the need.

If the model view is persistent, it will be stored in a physical file or a database, otherwise if it is transient the physical storage of the view is not necessary. The model views are updated using EXPRESS-X (Eastman and Jeng, 1999) or XSL (IFC Model View Definition, 2007) languages. The Model Views can be used for eliminating semantic differences between the BIM and another domain model, and can also help in the model simplification process, i.e. Isikdag (2006) demonstrated the use of BIM model views while mapping information from the IFC model to an application specific view, which is developed to interact with a GIS.

3 Buildings within 3D Geospatial Models

In geospatial model a 3D geospatial object can be defined by its geometric or topologic representation or by using both representations (van Oosterom *et al* 2002, Breuning and Zlatanova 2006). As explained in Lake et al. (2004) topologic objects can be mapped into geometric objects.

3.1 3D Geometrical Representations

Requicha (1980) summarised six main approaches for representing 3D objects by computers. These approaches are primitive instancing, spatial occupancy enumeration, cell decomposition, constructive solid geometry, sweeping and representing the object using boundaries (Boundary Representation, BRep). Two of these geometrical representations sweeping and BRep are used commonly to represent 3D objects in 2,5D vector models. Swept or extruded 3D geometries are created by GIS applications on demand. Research in the area demonstrated the use of cell decomposition and 3D spatial occupancy enumeration (voxels), to represent 3D objects and also outlined the need for using CSG representations.

In ISO 19107 (Standard for Geographic Information: Spatial Schema, compliant with OGC Abstract Specifications Topic 1: Spatial Schema) geometric model, a primitive solid object (GM_Solid) is constructed by its boundaries (bounding surfaces) thus its 3D representation method is BRep. Other solid types such as Cone, Cylinder, Sphere are defined as a subclasses of Surface Patch (GM_SurfacePatch), Parametric Curve Surface (GM_ParametricCurveSurface) and Gridded Surface (GM_GriddedSurface) and are interpreted as surfaces. According to ISO 19107 a composite solid (GM_CompositeSolid) can be is generated by using a set of solids that join in pairs on common boundary surfaces to form a single solid. The resulting model will also be in the form of a BRep model.

3.1 Buildings within CityGML

CityGML Implementation Specification (2007) defines CityGML as a common semantic information model for the representation of 3D urban objects that can be shared over different applications. The current version of CityGML refers to only earth surface objects and above earth surface objects, but research and first tests has been reported on incorporating underground objects (Emgard and Zlatanova, 2007, 2008). CityGML is designed as an open data model and XML-based format for the storage and exchange of virtual 3D city models. It is implemented as an application schema of the Geography Markup Language 3 (GML3). CityGML defines the classes and relations for the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantic and appearance properties. CityGML is applicable for large areas and small regions and can represent the terrain and 3D objects in different levels of detail simultaneously. In CityGML 5 levels of detail (LOD) were defined in order to represent city objects. In terms of representing buildings 4 out of 5 LODs are used. As explained by CityGML Implementation Specification (2007) as follows:

1. LOD0 is essentially a two and a half dimensional Digital Terrain Model, over which an aerial image or a map may be draped.
2. LOD1 is the well-known blocks model comprising prismatic buildings with flat roofs.
3. A building in LOD2 has differentiated roof structures and thematically differentiated surfaces.
4. LOD3 denotes architectural models with detailed wall and roof structures, balconies, bays and projections. High-resolution textures can be mapped onto these structures.
5. LOD4 completes a LOD3 model by adding interior structures for 3D objects. For example, buildings are composed of corridors, rooms, interior doors, stairs, and furniture.

In LODs 2-4 of CityGML the building façade is defined in form of Boundary Surfaces i.e. WallSurface, Roof Surface, GroundSurface or ClosureSurface. The BuildingInstallation class is used for representing building elements like balconies, chimneys, dormers or outer stairs, which affect the outer appearance of a building. In LODs 3-4 the openings in the BoundarySurfaces can be represented as Doors and Windows. In LOD4, the Room class is used to represent the interior of the building. A Room can be defined by its Boundary Surfaces i.e., Interior Wall Surface or Floor Surface. The movable objects i.e. lamps, table and chairs are represented with Building Furniture class. Building Installation class is used to model other immovable building elements such as pillars and stairs.

4 The Framework

The IFC model consists of 4 conceptual layers as Resource, Core, Interoperability and Domain. The Interoperability layer defines concepts (or classes) common to two or more domain models (i.e. Architecture, Construction management) defined in IFC. As explained in IFC Technical Guide (2000) it is through the schemata defined at the interoperability layer that, multiple domain models can be plugged into the common IFC core. The entities in the Interoperability Layer and specifically in the 'Shared Building Elements' part of IFC specification can form an interesting starting point for the development of the transformation framework between IFC and CityGML models. The 'Shared Building Elements' part of the IFC specification, contains entities that represent the basic components of a building such as a beam, column, wall or slab (Fig.1).Each instance of these entities corresponds to an element of a building in a BIM. The '*Representation*' attribute of each entity in this set (i.e. IfcColumn, IfcBeam) will refer to an 'IfcProductRepresentation' entity which defines the geometry of that building element. Similarly the orientation and absolute location of a building element can be acquired from the 'IfcObjectPlacement' entity referred from the 'ObjectPlacement' attribute. In summary, a building elements' geometry is referred from a BIM object that contain semantic information about that element.

Transforming information from IFC to CityGML requires a two-step approach: transforming semantic information and transforming geometries. Since the objects (classes) in these two models are very diverse, the two steps cannot be performed separately. An object in one of the models might be mapped to a group of objects (and vice versa), which requires a careful consideration on the order or converting geometries and semantics.

In order to perform a successful transformation operation;

- A set of rules (a rule base) needs to be clearly defined in the first stage, in order to define the semantic mappings between the classes of two models for each LOD of CityGML.
- The second stage will be building up the rules/algorithms for geometric model simplification. A BIM model view can facilitate model simplification in this stage.
- The final stage will be defining the information that will be transformed to form the attributes of the CityGML objects for each LOD.

The following sections present a general overview of (semantic and geometric) information transformation from IFC to form CityGML models in different LODs of CityGML, mainly concentrating on the transfer of geometric information.

4.1 Transfer of Semantic Information from IFC to *_AbstractBuilding* class

The information required for populating the attributes of the *_AbstractBuilding* class in CityGML model can be acquired from several different IFC entities. For example an object count of the *IfcStorey* entity will provide the number of stories of the building, and the *IfcBuilding* entity will help in determining the number of stories above the groud. On the other hand the information for determining the CityGML *RoofType*, can be acquired from *IfcRoof* entity.The year of construction of the building can be acquired from *IfcWorkSchedule* entity.

4.2 Generation of Buildings in CityGML LOD 1

In the LOD 1 of the CityGML model, the geometry of a building is represented as a prismatic object with a flat top (Fig.2). The geometry of the building can either be represented with *gml:SolidType* as a volumetric object or the exterior surface (façade) of the building is represented with *gml:MultiSurfaceType*. In fact, in LOD 1 every wall is represented with a single face (flat surface).

In order to generate a building model in LOD 1, the façade of the building can be acquired from the *IfcWall* entities (representing the façade walls) and *IfcSlab* entities(representing the building roof). In this situation, the façade needs to be simplified using a BIM model view, in such a way that each wall (and roof slab) will only be represented with a single face (flat surface). An alternative way of getting a simplified geometry is using bounding box representation of the elements. The extensions to building such as balconies also need to be disregarded when constructing the LOD 1 model. In addition, if there is any curvature in the geometry of walls and roof slab, these should also be eliminated.

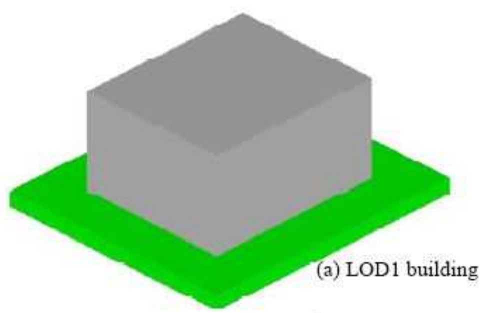

Fig.2 : Geometrical Representation of a building in CityGML LOD 1 (CityGML Implementation Specification, 2007)

On the other hand, as the model in LOD 1 represents the building in a simple form, another approach for generating the building representation would be acquiring the footprint of the building from the *IfcSlab* entity(representing the ground or the top floor) and the height of the building from the *IfcSlab* entity that are referenced from the *IfcRoof* entity (in other words the Roof Slabs). In this case, the BRep model of the building can easily be constructed using the coordinates of the building footprint and the height of the building. The Terrain Intersection Curve (TIC) can be generated either using the building footprint acquired from the IfcSlab entity or using *IfcWall* entities that form the building façade.

4.3 Generation of Buildings in CityGML LOD 2

In LOD 2 of the CityGML model, the outer façade of the building can be represented in a greater detail. The biggest differences between the LOD 1 and LOD 2 (in terms of geometrical representation) are, i.) the outer walls of the building can be represented with multiple faces and ii.) the curve geometries of the building façade can be represented within the model structure (Fig 3). On the other hand LOD 2 enables the representation of outer building installations such as chimneys, balconies, dormers, outer stairs etc. These installations can be represented within the *BuildingInstallation* class as *gml:Geometry* (as an aggregate of multiple geometrical types). Starting from LOD 2 (in order to represent the semantic differentiation between the building elements) different parts of the outer façade of the building can be denoted within different classes of CityGML. These different classes are aggregated under *_BoundarySurface* class and can be used to explicitly differentiate Roof Surface, Wall Surface, Ground Surface and Closure Surfaces. Each of these surfaces is represented with *gml:MultiSurfaceType.* (i.e as an aggregate of multiple surfaces).It should be noted here that the openings on the façade are only represented in LOD 3 and LOD 4, thus the façade representation in LOD 2 will not include the representations of openings such as door and windows (but there is no obligation against the representation of openings as a part of a face, i.e. wall surface, in the façade.).

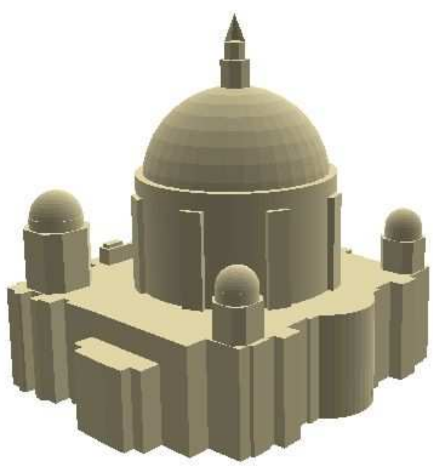

Fig.3: Building façade with curved geometries represented in CityGML LOD 2 (CityGML Dataset, City of Berlin)

The façade of the buildings in LOD 2 can be wrapped by raster-based 2D textures (Fig.4).

In CityGML LOD 2, the façade of the building can be represented as a result of an IFC model simplification. In order to generate a building model in LOD 2, *IfcWall* and *IfcSlab* entities need to be used as a primary resource. Entities such as *IfcColumn*, *IfcBeam*, *IfcCurtainWall*, can also help in forming the building façade at LOD 2. In order to preserve the semantic information about the building elements it can be advised to map the information from the IFC entities to the classes aggregated under the *_BoundarySurface* class.

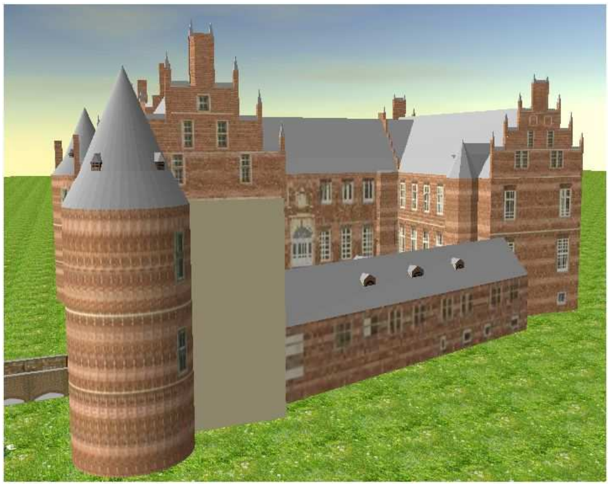

Fig.4: CityGML LOD 2 model showing a building façade with raster texture (CityGML Dataset, Castle Herten)

In such a case, the *WallSurface* objects in the CityGML model will be generated by acquiring the outer wall face from the *IfcWall* enitiy. The geometric structure of the *IfcWall* entity is usually defined in form of a Sweeping and CSG Solid, thus a Sweeping and CSG to BRep conversion will usually be needed before the simplification process. After this conversion process, the model simplification will serve for eliminating the sides of the wall that will not be transferred into the CityGML model. If there are curtain walls in the building, geometric information from *IfcCurtainWall* entity can also be mapped to *WallSurface* class of CityGML. In this mapping process the openings in the *IfcWall* and *IfcCurtainWall* classes can be omitted, as they can not be represented in this LOD, or the surface geometry of the Door and Windows can be represented within the *Wall Surface* class, but in this case semantic information will be lost (i.e. a

Door/Window will be interpreted as a face of the wall).The curved surfaces in the *IfcWall* can be represented by the *WallSurface* class, but the BRep model will contain more objects to represent the curvature of the wall surface.

The *RoofSurface* objects can be generated by simplifying the geometry of *IfcSlab* entities that are referenced from the *IfcRoof* entity. Similar to the walls, the BRep structure of the roof slab first needs to be constructed (by Sweeping to BRep conversion) before the simplification process.

The *GroundSurface* object can be generated either by acquiring the geometric information from the *IfcSlab* entity that is used to represent the ground floor of the building, or by joining the centrelines of the façade walls (represented by *IfcWall* entities) at the ground level, for forming a closed polygon, but some line clipping operations can be required in this latter approach if the centrelines of the walls cross each other at some points.

Columns and Beams that are visible from the outside of a building can be represented within *BuildingInstallation* class (as a *gml:Solid* or *gml:MultiSurface*). Geometric information regarding these building elements can be acquired from *IfcColumn* and *IfcBeam* entities. Similarly the stairs of the building can be represented in a LOD 2 model, by the *BuildingInstallation* class. In such a case, the geometry of the stairs needs to be acquired from the *IfcStair* entity. The geometrical representation of building elements, which are represented without a semantic definition in IFC, can be obtained from *IfcBuildingElementProxy* entities.

4.4 Generation of Buildings in CityGML LOD 3

In a LOD 3 CityGML model, the façade of the building can include openings such as doors and windows (Fig 5). The openings in the building are represented with *Door* and *Window* classes which are defined as the subclasses of the (abstract) class *Opening.* Each opening is represented with *gml:MultiSurface* geometry. The *Window* class is used to represent the windows inside and outer façade of the building. Similarly the *Door* class is used for modelling the doors that are between the adjacent -inside-spaces and located at the outer façade of the building.

Fig.5: Geometrical Representation of a building in CityGML LOD 3 (CityGML Dataset, Four Buildings in LOD 3)

Similarly the door and window classes in the IFC model (*IfcDoor* and *IfcWindow*) are referred from the (IFC) opening element (*IfcOpeningElement*) but, in contrast to the CityGML object model, in IFC model, neither *IfcOpeningElement* is an abstract class, nor the *IfcDoor* and *IfcWindow* are the subclasses of the *IfcOpeningElement.* The *IfcOpeningElement* is an element that is used the describe the geometry and semantics of an opening which can contain multiple door and windows, thus an *IfcOpeningElement* can refer to multiple *IfcDoor* and *IfcWindow* elements.

The doors and windows (in the IFC model) are represented with the help of two different relationships. First, the opening is referred from its spatial container (i.e. a wall) by

IfcWall→IfcRelVoidsElement →IfcOpeningElement

relationship (*IfcRelVoidsElement* is the element that enables the semantic relationship between a building element and an opening). As the next step, the door or window is referred from the opening by

IfcOpeningElement→ IfcRelFillsElement→IfcWindow/IfcDoor

relationship (in this case *IfcRelFillsElement* is used for establishing the semantic relationship between the opening and the element that fills that opening)

The *Door* and *Window* objects in CityGML can be generated by the information acquired from the *IfcDoor* and *IfcWindow* classes. In order to acquire geometric information from these IFC classes (i.e *IfcWindow*) using its related container (i.e. *IFcWall*), the following path of relationships needs to be followed:

IfcWall→*IfcRelVoidsElement*→*IfcOpeningElement*→
IfcRelFillsElement→*IfcWindow*

The coarse geometric representation of doors and windows in IFC, is very similar to the wall/slab representation (i.e. they are represented as Sweeping and CSG models), but a (finer) geometric representation of these elements is usually presented in form of a BRep. If the coarse geometric representation would be used when acquiring information from the IFC model, a Sweeping /CSG to BRep conversion will become a need. In IFC model, additional semantic information such as the panel type/number and the opening direction is also provided for the door and window objects in *IfcDoor* and *IfcWindow* classes, but CityGML *Door* and *Window* classes do not contain any specific (non-generic) attributes to store such information.

4.5 Generation of Buildings in CityGML LOD 4

The LOD 4 of CityGML provides the option to represent the interior structure of a building. The main CityGML classes that enable this representation are *Room* and *IntBuildingInstallation*. The *Room* is described as the semantic object for modelling the free space inside the building. According to CityGML specification the *Room* should be closed topologically (i.e. by using a *ClosureSurface* when necessary). The geometry of the room is generally represented by *gml:Solid* , but if the topologic correctness of the boundary can not be guarantied its geometry can also be represented by *gml:MultiSurface* . In order to preserve the semantic information related to a room (in a LOD 4 model) different parts of the *Room* can be represented within the classes aggregated under the *_BoundarySurface* class. These classes can be used to semantically differentiate the *CeilingSurface*, *InteriorWallSurface*, and *FloorSurface* (Fig. 6).

In CityGML LOD 4 model, the *Room* (*gml:Solid*) objects are topologically connected by surfaces at the doorways. The *Room*s are defined as being adjacent if the share an *Opening* or a *ClosureSurface*. The *Room*s can contain, movable objects such as furniture which are represented with *BuildingFurniture* class. A building can also contain other installations such as interior stairs, railings and pipes. These are represented by *IntBuildingInstallation* class in CityGML LOD 4 model. The geometry of both *BuildingFurniture* and *IntBuildingInstallation* objects are represented by *gml:Geometry* type.

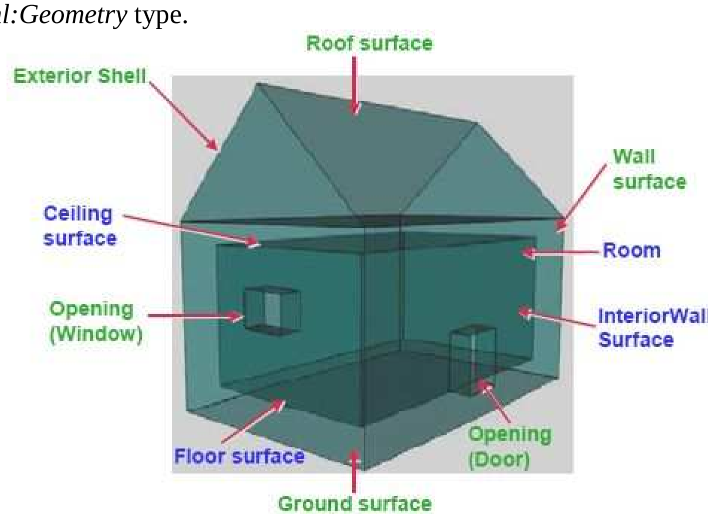

Fig. 6: Indoor Representation of a building in CityGML LOD 4 (CityGML Implementation Specification, 2007)

The IFC model does not have a specific class for denoting room spaces; in fact, the model has a generic class (*IfcSpace*) for defining the spaces in the building. *IfcSpace* class represents an area or volume bounded actually or theoretically. A space (*IfcSpace*) is generally associated to a building storey (*IfcBuildingStorey*) or in case of exterior spaces to a site (*IfcSite*). As explained in IFC 2x3 documentation (2007), a space may span over several connected spaces, therefore a space group provides for a collection of spaces included in a storey. A space can also be decomposed in parts, where each part defines a partial space. *IfcSpace* is represented within multiple geometric representations (usually by a combination of Sweeping and CSG and rarely with a BRep). As mentioned earlier, a Sweeping/CSG to BRep conversion can also become a need (in this case) to transfer information from the *IfcSpace* entity for generating the *Room* object in CityGML model. Other approach for mapping room geometries can be acquiring information from *IfcWall* and *IfcSlab* entities that form room boundaries. In this approach the *FloorSurface* and *CeilingSurface* objects for a *Room* object at storey N of a building, will be generated by the information acquired from *IfcSlab* entities located in storey N and N+1. The *IfcSlab* entity at storey N will be used for generating the *FloorSurface* and *IfcSlab* entity at storey N+1 will be used for generating the *CeilingSurface*. Information acquired from *IfcWall* entities that are representing the interior walls will help in generating the *InteriorWallSurface* objects of CityGML. On the other hand, information acquired from the *IfcWall* entities that are representing the exterior (façade) walls of the building can be used to generate both the *Wall Surface* objects (for representing the building façade) and *InteriorWallSurface* objects (for representing the interior walls of the exterior rooms of the building). The interior *Door* and *Window* objects can be generated by the information acquired from *IfcDoor* and *IfcWindow* entities by using the approach explained in the previous section.

Structural elements such as beams and columns and other installations such as stairs, ramps and railings are semantically differentiated and represented in separate classes (as *IfcBeam*, *IfcColumn*, *IfcStair*, *IfcRailing*, *IfcRamp*) in the IFC model. The information acquired from these elements and installations can only be represented as *IntBuildingInstallation* in CityGML LOD 4. A Sweeping/CSG to BRep conversion also becomes a need specifically for representing the beams and columns in CityGML, as IfcBeam and *IfcColumn* entities are usually represented as Swept or CSG solids.

Furnitures in the building is represented with *IfcFurnishingElement* entity, and the geometry of the furniture is usually represented by BRep in the IFC model. Information acquired from the *IfcFurnishingElement* can be used for generating the *BuildingFurniture* in CityGML LOD 4 model.

5 Conclusion

Information from design and real-world models can automatically be exchanged if a formal framework can be made available. Following a background literature review, the paper presents a general overview of (semantic and geometric) information transformation from BIM (specifically from IFC models) into CityGML models. The information provided in the paper can contribute to the efforts towards such a building a formal framework. The focus of this work was on transforming the building information as buildings are the key elements of the city fabric. On the other hand in terms of information modelling, the building elements explicitly described in different LODs and the building representation in higher LODs of CityGML, provides the possibility for building up rules for semantic matching between BIMs and CityGML.

Although there are explicit semantic classifications on both models a transformation needs to tackle problems due to semantic mismatches between the classes in two models. For example, an opening on a slab is represented with the *IfcOpening* entity in the IFC model, in contrast the CityGML model does not provide a class for representing an opening that does not contain a door or window. Similarly, CityGML model does not provide a class for representing the storey of a building. On the other hand, an entity of the IFC model can correspond to different objects of CityGML also due to the semantic mismatches between two models. For example, the CityGML model provides two different objects for representing the *FloorSurface* and *CeilingSurface* of a room, but these surfaces are represented with a single entity in the IFC model (i.e. *IfcSlab*).

Another problem in the transformation process might exist due to differences in granularity of both models. A BIM (i.e IFC model) is a finer-grained model, in terms of representation of building elements, when compared with the CityGML model. This difference in levels of granularity can cause problems when exchanging information between two models, i.e a *Window* object in CityGML might correspond to 3 or 4 different '*IfcWindow*' entities in the IFC model. In this case 1-to-1 object matching will cause the creation of redundant CityGML objects when transforming information from IFC to CityGML model, on the other hand if a need for transforming the information vice versa appears, the resultant IFC model might become very coarse grained, i.e. 3 or 4 windows might be represented with a single *IfcWindow* entity.

The overview presented in this paper, only concentrated on unidirectional information transformation (i.e. from a BIM to the CityGML models) as the need for such a transformation appears more eminent today, in fact bidirectional transformation might also be required to support renovation related tasks where an information model of a building usually does not exist.

The findings of the study indicate that:

- IFC model contains all necessary information for representation of buildings in different LODs of CityGML model.
- It is possible to define rules for transforming geometrical information from IFC entities to CityGML objects.
- It is possible to define rules for facilitating semantic matching between two models.

The results of the previous research illuminate that the model views (BIM sub-models) can play role in the semantic matching process in addition to simplifying the model before the geometrical transformation.

The ongoing research will focus of formally defining the stages of the transformation framework, constructing the rule base for semantic matching, developing model views and algorithms for achieving geometric/semantic model simplification, and testing the developed models views and algorithms. The future research will also investigate the possible benefits and investigate methods of bidirectional transformation.

References

Breuning, M. and S. Zlatanova, 2006, in: Zlatanova&Prosperi (Eds.) 3D GeoDBMS, in 3D large scale data integration: challenges and opportunities, CRC Press, Taylor&Francis Group, pp. 87-115

CIS2, 2007, Model documentation webpage. CIMSteel Integration Standards Relase 2, 2007, http://www.cis2.org [last accessed 10-2007]

CityGML Dataset, City of Berlin, http://www.citygml.org/ (last accessed 06-2008)

CityGML Dataset, Castle Herten, http://www.citygml.org/ (last accessed 06-2008)

CityGML Dataset, Four Buildings in LOD 3, http://www.citygml.org/ (last accessed 06-2008)

CityGML, Exchange and storage of Virtual 3D city Models 2008, available at http://www.citygml.org/ [last accessed 06-2008]

CityGML Implementation Specification,2007, Candidate OpenGIS Implementation Specification (City Geography Markup Language) https://portal.opengeospatial.org/files/?artifact_id=16675 [last accessed 12-2007]

Eastman, C and Jeng, T. S. 1999, A database supporting evolutionary product model development for design, Automation in Construction, 8(3):305-323

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

Emgard, L. and S. Zlatanova, 2008, Design of an integrated 3D information model. In Coors, Rumor, Fendel & Zlatanova (eds.), Urban and regional data management: UDMS annual 2007 (pp. 143-156), Taylor & Francis Group, London, UK

IAI/BuildingSmart, Web Site for International Alliance for Operability http://www.iai-international.org

IFC, 2008, Model documentation webpage. International Alliance for Operability, 2008, http://www.iai-international.org/Model/R2x3_final/index.htm [last accessed 06-2008]

IFC Explorer, 2008, Tool for viewing and conversion of IFC models http://www.iai.fzk.de/www-extern/index.php?id=1040&L=1[last accessed 06-2008]

IFC Model View Definition ,2007, IFC Model View Definition Format, http://www.iai-international.org/software/mvd.shtml [last accessed 05-2007]

IFC Technical Guide, 2000,,The Techincal Guide for Industry Foundation Classes http://www.iai-international.org/Model/documentation/IFC_2x_Technical_Guide.pdf [last accessed 12-2000]

Isikdag,U. , 2006, Towards the Implementation of Building Information Models in Geospatial Context, PhD Thesis, University of Salford, UK.

ISO 16739 Industry Foundation Classes, Release 2x, Platform Specification http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38056

ISO 19107 Geographic information -- Spatial schema http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26012

Kolbe, T. and G. Gröger, 2003, Towards unified 3D city models. Proceedings of the ISPRS Comm. IV Joint Workshop on Challenges in Geospatial Analysis Integration and Visualization II2, September 8-9, 2003 in Stuttgart, 8p.

Lake, R., Burggraf, D., Trninic, M., Rae, L., 2004, Geography Mark-up Language: Foundation for the Geo-Web, John Wiley and Sons

Lapierre, A. and P. Cote, 2008, Using Open Web Services for urban data management: a testbed resulting from an OGC initiative offering standard CAD/GIS/BIM services, in Coors, Rumor, Fendel & Zlatanova (eds.): Urban and Regional Data Management; UDMS Annual 2007, Taylor and Francis, London, pp. 381-393

Lattuada, R., 2006, Three-dimensional representations and data structures in GIS and AEC. In: Zlatanova& Prosperi (eds.): Large-scale 3D data integration - Challenges and Opportunities, Taylor&Francis, pp.57-86

NBIMS, 2006 National BIM Standard Purpose, US National Institute of Building Sciences Facilities Information Council, BIM Committee, http://www.nibs.org/BIM/NBIMS_Purpose.pdf [last accessed 02-2008]

Pu, S. and S. Zlatanova, 2006, Integration of GIS and CAD at DBMS level, In: Fendel&Rumor (Eds); Proceedings of UDMS'06 Aalborg, Denmark May 15-17, 2006, TU Delft, 2006, pp. 9.61-9.71

Requicha, A.A.G., 1980, Representations for Rigid Solids: Theory, Methods and Systems, ACM Computing Surveys,12(4):434-464

Safe Software, 2008, FME Desktop Translator/Converter Software http://www.safe.com/products/desktop/formats.php [last accessed 06-2008]

Van Oosterom, P. van, J. Stotter and e. Janssen, 2006, Bridging the worlds of CAD and GIS, In: Zlatanova& Prosperi (eds.): Large-scale 3D data integration -Challenges and Opportunities, Taylor&Francis, pp.9-36

van Oosterom, P, J. Stoter, W. Quak and S. Zlatanova, 2002, The balance between geometry and topology, in: Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling, D.Richardson and P.van Oosterom (Eds.), Springer-Verlag, Berlin, pp. 209-224

Zlatanova, S., S. Pu and W.F. Bronsvoort, 2006, Freeform curves and surfaces in DBMS- a step forward in spatial data integration, in Nayak, Pathan&Garg (Eds.), Proceedings of the ISPRS Commission IV Symposium on 'Geospatial Databases for Sustainable Development', 27-30 September, 2006, Goa, India; Archives of ISPRS Vol. 36, Part 4A, pp. 407-412