

Towards Interoperating CityGML and IFC Building Models: A Unified Model Based Approach

Mohamed El-Mekawy, Anders Östman, and Khurram Shahzad

Abstract CityGML represents 3D urban objects that can be shared over different applications, whereas, IFC provides a very detailed semantic model for 3D building representations using constructive elements like beams, walls, etc. Attempts have been made to interoperate CityGML and IFC for seeking useful common applications. However, these efforts use a unidirectional method (mostly from IFC to CityGML) for conversion processes. A bidirectional method can lead to development of unified applications in the areas of urban planning, building construction analysis, homeland security, etc. The benefits of these applications clearly appear at the operational level (e.g., cost reduction, unified data-view), and at the strategic level (e.g., crisis management and increasing the analyses capabilities). In this paper, we present an approach for interoperating CityGML and IFC based on development of a unified building model for converting IFC to CityGML and vice versa. The conversion is a two-steps process in which a model is firstly converted to the unified model and secondly to the target model. Finally, we demonstrate the approach and outcome of each step by a hospital building case that is located in Norrtälje City, north of Stockholm, Sweden.

M. El-Mekawy (✉)
Future Position X, Gävle, Sweden
e-mail: moam@kth.se

A. Östman
GIS Institute, University of Gävle, Gävle, Sweden
e-mail: Anders.Ostman@hig.se

K. Shahzad
Department of Computer and System Sciences, Royal Institute of Technology (KTH), Stockholm, Sweden
e-mail: mks@dsv.su.se

1 Introduction

By merging building models and geographic information models, public and private enterprises are attempting to fulfil the application demands in construction analysis, urban planning, homeland security, etc. (Benner et al. 2005; Isikdag and Zlatanova 2009a). Geometric models have been developed in both domains but semantic models are relatively few. The two most prominent semantic models are Industry Foundation Classes (IFC) and City Geography Markup Language (CityGML) (Isikdag and Zlatanova 2009a).

The goal of IFC is to specify a common language for building industry technology that improves communication, productivity, delivery time, cost, and quality throughout the design, construction and maintenance life cycle of buildings (Hallberg and Tarandi 2009). IFC is then used to assemble computer readable models that contain related information to different parts of a building (Karola et al. 2002; IFC Model, <http://www.iai-tech.org/ifc/IFC2x3/TC1/html/>, accessed 01-2011). CityGML is however used to define information related to topological and semantic properties of a geographical area including buildings (OGC, <http://www.opengeospatial.org/>; Gröger et al. 2008). On one hand, IFC has been developed as an ISO standard and it has been largely accepted for the building industry (Isikdag and Zlatanova 2009a). On the other hand, CityGML has been recently adopted as an international standard for modelling cities in the Open Geospatial Consortium (OGC) and the EU (OGC, <http://www.opengeospatial.org/>; Gröger et al. 2008). This has led to major increase in the development of new application areas, software tool kits and extensions to existing systems for supporting IFC as well as CityGML.

The integration of IFC and CityGML is seen today as a needed step for getting a more complete picture of 3D modelling at different levels of detail, i.e., sharing and exchanging information between building industry objects (represented in IFC) and geospatial object (represented in CityGML). Several efforts have been made to integrate CityGML and IFC. All these efforts are mainly in form of developing frameworks, extended discussion for addressing requirements, or developing conversion tools. For frameworks, the IFC for GIS (IFG) project was initiated by The Norwegian State Planning Authority (Statens Bygningstekniske Etat) and completed in 2007. This framework aimed to exchange building information between CAD systems and GIS using IFC. The project succeeded to create a mapping specification from XML version of IFG geometry to GML and vice versa (IFG, http://www.iai.no/ifg/index_history.html). Looking specifically on technical aspects, another framework was proposed by Nagel (2007) and aimed for algorithms that automatically transform IFC building models into CityGML models (Nagel 2007). In 2009, Isikdag and Zlatanova did complement Nagel's framework by proposing a framework for automatic generation of buildings in CityGML using BIM based on definition of building semantics and components (Isikdag and Zlatanova 2009a). Following the holistic view of 3D city modelling aspects, an

extended discussion on conceptual requirements for converting CityGML to IFC models was proposed in 2009 by a team led by Thomas Kolbe at the Technical University of Berlin (Nagel et al. 2009). They proposed a framework that integrates 3D graphics/data of buildings and urban areas (X3D, DXF, KML, COLLADA, etc.) with semantic data in a CityGML target schema. Additional to that, a few conversion environments can be seen in this area. Léon (2009) demonstrated his team's latest Application Domain Extensions (ADE) that integrates Building Information Model (BIM) data based on the open standard IFC into CityGML (Van Berlo 2009). Not only research efforts, but commercial software products for conversion from IFC to CityGML [e.g., IfcExplorer (http://www.ifcwiki.org/index.php/IfcExplorer_CityGML_Export) and Safe Software (<http://www.safe.com/products/desktop/formats.php>)] also contribute to the development of 3D city modelling integration. However, these attempts have either; (a) an approach for a unidirectional conversion with a focus on converting geometries (mostly from IFC to CityGML), (b) a discussion about what should be done in terms of integration, i.e., how it should be done is not sufficiently implemented yet, (c) focused on down-grading IFC to lower LoDs (LoDs) in CityGML, or (d) a discussion on the interest of rich semantics of IFC.

Several studies (Benner et al. 2005; Isikdag and Zlatanova 2009a; Nagel et al. 2009) have emphasized that a formal framework for strict semantic and geometry conversion is required for a complete integration of CityGML and IFC. As a consequence, the purpose of this paper is to propose and describe a unified model oriented approach that can be used for bidirectional conversion between IFC and CityGML. The proposed approach thereby contributes towards increasing integration of CityGML and IFC for extending 3D city model applications.

Geometry is highlighted as one of the main problematic concerns for integrating IFC and CityGML (Nagel et al. 2009; Isikdag and Zlatanova 2009b). However, most of the recent efforts have focused on the conceptual integration or conversion processes. Considering the vast amount of efforts for defining geometric differences and developing conversion algorithms (Lapierre and Cote 2008), this type of problem is not in our objectives. In this study we instead focus our discussion on the conceptual integration and mapping of different objects in both IFC and CityGML standards.

The rest of the paper is organized as follows: Sect. 2 discusses how we approach the research problem. Building models for CityGML and IFC are presented in Sect. 3, followed by detailed discussion on UBM in Sect. 4. In Sect. 5 the two-steps approach for conversion of IFC to CityGML and CityGML to IFC is presented along with an illustrative case study.

2 Research Approach

A unified model is here defined as a superset model concept that is extended to contain all the features and objects from both IFC and CityGML building models. It is an intermediate model relating objects from both models. The unified model

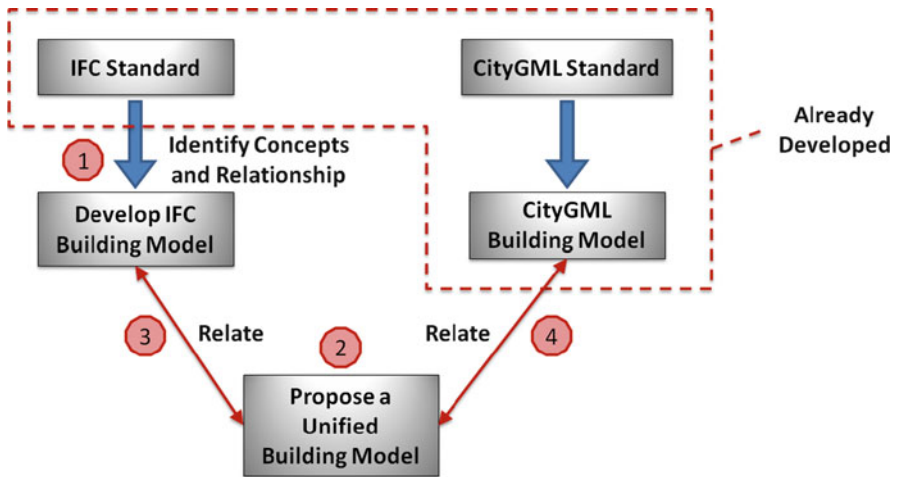


Fig. 1 The research approach

is originally derived from the superset concepts of mathematics in 1970s led by Thomas J. Jech and other researchers (Jech 1971; Miguel et al. 2002). It has also been used in software engineering from the mid 1990s.

Due to its wide acceptance, Unified Modelling Language (UML) has been selected as the modelling language for our Unified Building Model (UBM) approach. The requirements on the proposed UBM are that conversion is done with a minimum information loss and an efficient schema matching and mapping process. The approach consists of the following steps as shown in Fig. 1: (1) elicitation of IFC building model, (2) development of the UBM, (3) conversion between IFC building model and UBM, and (4) conversion between UBM and CityGML building model.

Prior to discussing CityGML and IFC conversion, it is necessary to discuss each model separately.

3 CityGML and IFC Building Models

In order to interoperate CityGML and IFC it is essential to develop building models for both. This section discusses the development of building models for IFC and CityGML.

3.1 IFC Building Model

IFC is an object oriented format developed by the International Alliance for Interoperability (IAI) (<http://www.iai-tech.org/>). It is used to facilitate interoperability in

the building industry and sharing information among participants. IFCs are used to assemble computer readable models that contain data elements that represent parts of buildings and their relevant information (Lapierre and Cote 2007). There is no universally accepted building model for IFC (Kolbe et al. 2008). In this section we, however, present an IFC building model that is primarily based on the work done by the IAI and ISO in form of IFC standard documentation (IAI, <http://www.iai-tech.org/>), the ISO 16739 standard (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38056) and Benner et al. (2005). From the standards we identify important concepts (Fig. 2). UML standard notations are used for developing the IFC building model.

A building should have at least one storey and may have multiple storeys. Each building storey may have zero or more spaces related to it, i.e., a building structure which has only one wall is a building with zero spaces. Building elements and opening elements are subtypes of structural element. Each building element has zero or more opening elements, i.e., a wall without any door or window has zero openings, whereas each opening element (like door, window) is attached to only one building element. Figure 2 shows 12 types of building elements that can represent a building structure in IFC standard. The scope of this study is limited to building models that only represent constructed parts of buildings.

3.2 CityGML Building Model

CityGML is an open standard that has been implemented as an application schema for GML3 (OGC, <http://www.opengeospatial.org/>; Gröger et al., Open Geospatial Consortium). GML3 is the extendible international standard for spatial data exchange that has been developed and issued by the Open Geospatial Consortium

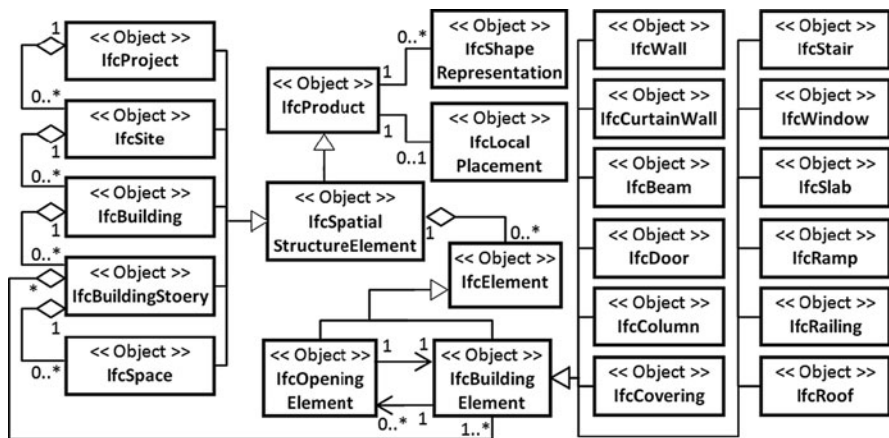


Fig. 2 IFC building model

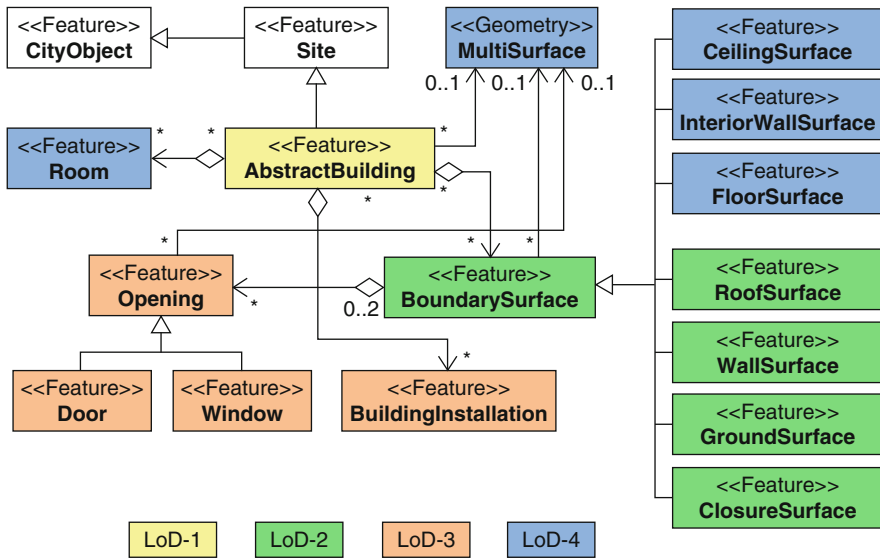


Fig. 3 CityGML building model

(OGC) (Cox et al. 2004) and the ISO TC211 (<http://www.isotc211.org/>). CityGML is developed as an open data model expressed by an XML schema. CityGML files can store and exchange virtual 3D objects and city models among applications (CityGML, <http://www.citygml.org>).

Based on ISO 19107 (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26012) and ISO 19109 (http://www.iso.org/iso/catalogue_detail.htm?csnumber=39891), a CityGML Building Model has been produced in the CityGML standard (Kolbe 2008). The building model (shown in Fig. 3) is an excerpted version from the CityGML standard in which only the used conversion concepts to IFC are represented, i.e., *BuildingFurniture*, and *IntBuildingInstallation* are not represented. UML standard notations are used for developing CityGML building model.

CityGML is developed in five levels of detail (LoDs) which are used to represent city model objects according to required details' level in different applications. LoDs are numbered from LoD0 to LoD4 and they have different accuracies and minimum dimensions of spatial objects that can be represented in each LoD. Individual buildings start to appear at LoD1. Therefore, we have presented CityGML building model from LoD1 to LoD4. Figure 3 represents the LoDs by different colours.

4 The Unified Building Model

IFC holds more detailed information about building objects than CityGML. Therefore, IFC to CityGML conversion is a less complex task as compared to CityGML

to IFC conversion. Due to that, most of previous work has aimed at IFC to CityGML conversion. However, two-ways conversion approach between CityGML and IFC has not been sufficiently explored.

In this section, we present the proposed Unified Building Model (UBM) and justify the inclusion of its elements. The UBM is capable of capturing information about spatial structures and building objects from both IFC and CityGML building models. The model is, therefore, used as an intermediate step for conversion of IFC to CityGML and vice versa.

UBM can be used as a starting point to support applications where information from both views (CityGML and IFC) is required for analyses. It can also facilitate modelling a database schema that is capable of capturing information that is required for all level of details. In the absence of UBM, if the application is designed at a specific LoD (e.g., LoD2) the information about higher level of detail (e.g., LoD4) cannot be acquired.

To build the UBM, all classes with their concepts were initially collected from both models while omitting their relationships. Overlapping concepts were merged and new objects were created in such a way that both indoor and outdoor objects are captured. Finally, relationships between the objects were redefined to produce our UBM. UML notations are used for representing its objects and relationships between them.

Figure 4 shows the UBM where different colours are used to represent different LoDs. The UBM is briefly described below.

As the study is limited to building structure, the objects beyond building (such as project and site) are not represented in the model. As building is a common object for both IFC and CityGML, *UBMbuilding* object has been used as a starting point for the model.

A building in CityGML consists of rooms, where a room is a space surrounded by different boundary surfaces. Storeys are not explicitly defined but they can be represented as an explicit aggregation of all building features on a

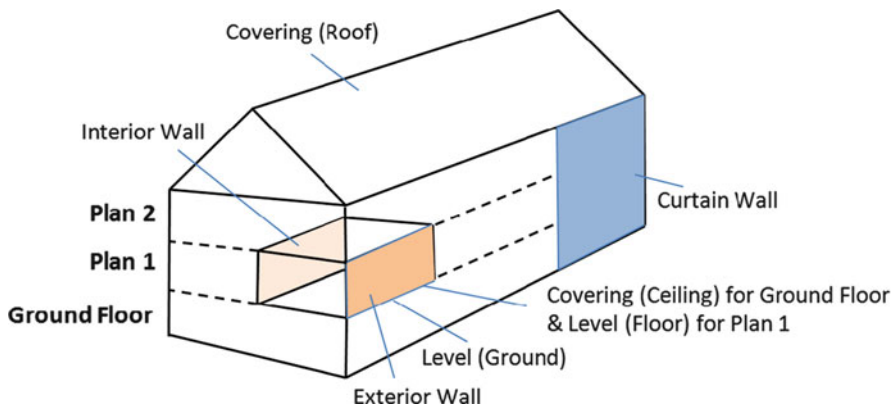


Fig. 4 The unified building model

certain height level (Gröger et al. 2008). However, in IFC, the structure is smoothly organized by breaking down a building into storeys and then into spaces that form a specific storey. In the UBM, we consider concepts from IFC as well as CityGML. A building in the UBM consists of explicit definition of storeys, at least one storey in a building. Each building storey may have zero or more spaces related to it. A space can be either opened or closed as shown in Fig. 4. The closed space represents a room which corresponds to the room definition in CityGML.

In CityGML, boundary surfaces are used by the class (*_BoundarySurface*) for structuring the exterior shell of a building and visible surfaces of a room. However, in IFC, a room or space is built by building elements that structure and surround it. We can then highlight that some of the IFC classes (e.g., *IfcWall*, *IfcDoor* and *IfcWindow*) are (or handled as) boundary surfaces in CityGML. In the UBM we have used concepts from both CityGML and IFC. We propose that a boundary surface is only used when it is needed to extract the box model of a complete building, part of a building or a specific storey. Building elements, however, are used to represent the elements which exist in the building structure. As every spatial place in a building (storey or space) may have a door or a window, opening is connected to both boundary surface and building element objects.

For building elements, we have defined some new concepts in such a way that all concepts from both IFC and CityGML can be covered. The main difference between building elements in CityGML and IFC is the representation of different surfaces, interior and exterior parts of a building (wall, roof and ground). Because of the need for different LoD representations and definitions of elements like (roof, ceiling, and slab) and (ground, floor and slab), we have defined the building elements as follows:

- *Covering* is a closing level that covers a space from the top side. It has two types; *Roof* for the top covering of a building or the top storey which gives the external shape of a building from above, and *Ceiling* for the covering of any space in a building. In both subtypes only constructed parts for a space are considered in the covering representation.
- *Level* is a walkable (not only horizontal) level that represents bottom level of a space. It has two types; *Ground* for the bottom level of ground floor which has a connection to the outer ground to give the external shape of a building from bottom level, and *Floor* for the bottom level of a space in any space of a building except the bottom (lowest) storey.
- *Wall* is a vertical/semi-vertical element that surrounds or subdivides spaces. It has three subtypes; (a) *CurtainWall* for the outer wall that covers a complete facade of a building or a part of it. It is usually not limited or attached to only one storey, (b) *Interior* for an internal wall between rooms or spaces (none of its faces has connection with the outer environment), and (c) *Exterior* for an external wall that has connection with the outer environment and represents a part of external facades of a building.

5 Conversion Between IFC and CityGML

In this section, we present our method for converting IFC to CityGML and for converting CityGML to IFC. IFC to CityGML conversion is done in two steps, i.e., IFC model is converted to the UBM, and the UBM is then converted to CityGML. Similarly, the CityGML to IFC conversion is done in two steps. We demonstrate the use of our approach by a case study of a hospital building. In the two following subsections, we present the proposed steps with their conversion procedures.

Locum Company (Locum AB), an owner of hospitals' buildings in Sweden, decided to adopt IFC standard in all of their upcoming new public buildings (<http://www.locum.se>). Modelling of existing buildings is still under discussion as it requires much more effort and updates of different data sources. In a research initiative, the first hospital building to model on IFC was the Norrtälje City Hospital in the north of Stockholm. A full IFC model for the hospital building was built (Fig. 6).

5.1 Conversion from IFC to CityGML

Step1. As a first step of IFC to CityGML conversion, we define a set of rules which describe how the objects in the UBM can be produced from IFC model.

At LoD1 of the UBM, a building is represented by the *UBMBuilding* object as a box/block model with a solid geometry. The needed information for constructing the box model can be acquired from attributes of different objects in IFC. For example, from *IFCWall*, *IFCSlab* and *IFCRoof* objects, it is possible to extract the construction information and coordinates of external walls and building roof. At this LoD, each building element (walls and roofs) should be simplified in its IFC

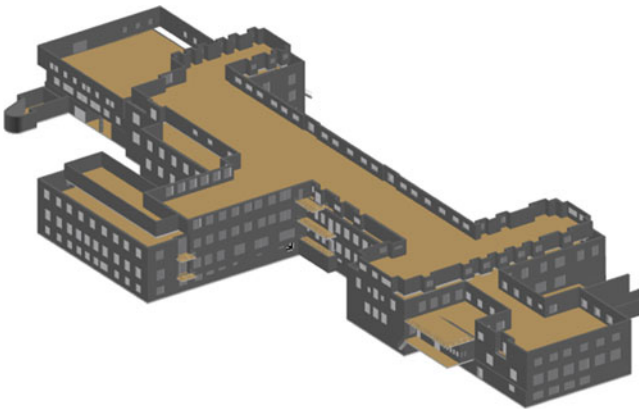


Fig. 6 IFC model of Norrtälje hospital

model into a single flat surface. These surfaces are used as surrounding surfaces of the box model. As what is needed at this LoD is only the outer envelope of a building, extension objects such as balconies and chimneys and geometrical curvature should be eliminated from the model. Therefore, the bounding boxes of the *IfcWall* elements combined with the top and ground floors from *IFCSlab* entity can formulate the outer envelope which is used for the box model. Alternatively, *IfcRoof* can also be used to form the boundary representation (BRep) box of the whole building by simplifying the roof shape, extracting its height and projecting it on the ground of the building. This may be written as:

$$f : \Omega_{[IfcWalls, IfcRoof]} \rightarrow \Omega_{UBMBuilding}$$

where Ω x represents the domain of x

At LoD2 of the UBM, external objects and their details are represented, i.e., curves in roofs, surfaces of grounds and external details of a wall. The needed information for *UBMBoundarySurface* can be acquired from a combination of the building elements like *IFCRoof*, *IFcWall* and *IFCSlab*. The needed information for *UBMBuildingElements* can be acquired from *building elements* of IFC. However, at this level only exterior elements (see Fig. 4) are presented. For example, *UBMExteriorWall* and *UBMCurtainWall* should be represented whereas *UBMInteriorWall* should not.

Information about different parts of the same wall that form a facade can be obtained from combined/Union-of *IfcWall* objects. These parts can be aggregated and stored in *UBMBoundarySurface* because they can be used as boundary surfaces to different rooms and spaces in the building. Additional to that, information about every part, which is obtained from *IfcBuildingElement*, will be referenced using *UBMBuildingElement* for the purpose of using them as separated building elements. *IfcCurtainWall* can be stored in *UBMCurtainWall* because they might be used as bounding surfaces with specific dimensions and attributes. Similarly, *IfcRoof* and *IfcSlab* can be aggregated and simplified to form *UBMCovering.Roof* and *UBMLevel.Ground* respectively.

Exterior building installations at this LoD represent an important part of the building image. Information about UBM building installations can be obtained from *IfcStair*, *IfcRamp*, *IfcRailing*, *IfcBeam* and *IfcColumn*. To differentiate between exterior and interior building installation, we however propose a generation of two projection footprints, vertical and horizontal. The vertical footprint is a result of projecting vertical structure elements, e.g., *IfcWalls* and *IfcColumn*, whereas, the horizontal footprint is a result of projecting horizontal structure elements, e.g., *IfcSlab* and *IfcBeam*. Using geometrical information, building elements are then checked versus the two projected footprints. Only connected objects will be represented as exterior building installations and others are omitted. Information about external objects and building installations with their surfaces will be then stored in *UBMBuildingElements* and *UBMBoundarySurface*. Rules may be formally written as:

$$\begin{aligned}
 f_{\text{Aggregation}} &: \Omega_{\text{IfcWalls}} \rightarrow \Omega_{\text{UBMBoundarySurface}} \\
 f &: \Omega_{\text{IfcBuildingElement}} \rightarrow \Omega_{\text{UBMBuildingElement}} \\
 f_{\text{Combination}} &: \Omega_{[\text{IfcWalls}, \text{IfcRoof}, \text{IfcSlab}]} \rightarrow \Omega_{\text{UBMBoundarySurface}} \\
 f &: \Omega_{\text{IfcWall}} \rightarrow \Omega_{\text{UBMExterior}} \\
 f &: \Omega_{\text{IfcCurtainWall}} \rightarrow \Omega_{\text{UBMCurtainWall}} \\
 f &: \Omega_{\text{IfcSlab}} \rightarrow \Omega_{\text{UBMLevel.Ground}} \\
 f &: \Omega_{\text{IfcRoof}} \rightarrow \Omega_{\text{UBMLCovering.Roof}} \\
 f &: \Omega_{[\text{IfcStair}, \text{IfcRamp}, \text{IfcRailing}, \text{IfcBeam}, \text{IfcColumn}]} \rightarrow \Omega \\
 &\text{UBMBuildingInstallation}[\text{UBMStair}, \text{UBMRamp}, \text{UBMRailing}, \text{UBMBeam}, \text{UBMColumn}]
 \end{aligned}$$

At LoD3 of the UBM, parts of external objects (that remains unrepresented at LoD2) are represented. Internal elements of building structure and opened spaces are represented as well. The information needed for *UBMStorey* can directly be acquired from *IfcBuildingStorey*. The information needed for *UBMSpace.Opened* can be acquired from a combination of *IfcSpace* and *IfcOpeningElement*.

In IFC, opening elements (*IfcOpeningElement*) are attached to building elements. Inside each opening element, there is usually one filling element (*IfcWindow* or *IfcDoor*). *IfcOpeningElement* is used itself as an element to describe the geometry and semantics of the opening. In this case, *IfcOpeningElement* may contain multiple *IfcDoor* and *IfcWindow* elements with referencing their geometry. Two classes in IFC define the representation of doors and windows. The first, *IfcRelVoidsElement*, is a one-to-one linking relationship between an element that contain an opening element (e.g., a wall) and one opening element that creates a void in the element. The second, *IfcRelFillsElement*, is another one-to-one relationship between an opening element and an element that fills that opening (Fig. 7). Geometric information about doors and windows are defined as Sweeping and CSG models as other building elements (e.g., walls and slabs). In CityGML, Window and Door objects are defined as subclasses of the abstract class Opening. They are, however, represented by different surfaces and modelled by *gml::MultiSurface*

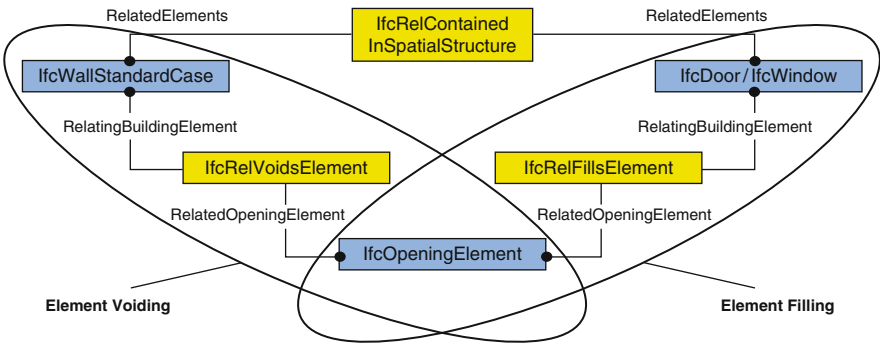


Fig. 7 Opening concepts in IFC

geometry. In our proposed UBM, we apply the same concept for opening elements as in CityGML. Therefore, the Door and Window objects in the UBM can be generated by acquiring information from classes of *IfcDoor* and *IfcWindow*. However, conversions from Sweeping/CSG geometric models that are used in IFC to BRep models that are used in CityGML have to be performed.

Formally, the conversion rules can be written as follows:

$$\begin{aligned}
 f &: \Omega_{\text{IfcBuildingStorey}} \rightarrow \Omega_{\text{UBMStorey}} \\
 f_{\text{Combination}} &: \Omega_{[\text{IfcSpace}, \text{IfcOpeningElement}]} \rightarrow \Omega_{\text{UBMSpace.Opened}} \\
 f &: \Omega_{\text{IfcBuildingElement.IfDoor}} \rightarrow \Omega_{\text{UBMOpening.UBMDoor}} \\
 f &: \Omega_{\text{IfcBuildingElement.IfWindow}} \rightarrow \Omega_{\text{UBMOpening.UBMWindow}}
 \end{aligned}$$

At LoD4 of the UBM, all objects of a building structure, i.e., the interior walls, floors, ceilings, etc. are represented. In IFC, there is no concept of specific *Room* object as in CityGML. *IfcSpace* class defines all volumes and areas that are bounded actually or theoretically. This definition includes rooms that are bounded by different building elements. As all IFC objects, Sweeping/CSG geometry is used for spaces and their elements which requires conversions to BRep geometric models that we use in the UBM. For differentiating between exterior and interior elements, the same concept of generating two projected footprints is used as at LoD2. The vertical footprint results projecting vertical structure elements, whereas, the horizontal footprint results from projecting horizontal structure elements. The geometries of these two footprints are checked to extract all building elements that intersect with them. Only connected objects will be stored as exterior building elements (i.e., walls and roof). To create spaces in the UBM, information from *IfcWall*, *IfcRoof* and *IfcSlab* that form the boundaries of rooms are used. Information about both *UBMCeiling* and *UBMFloor* can be acquired from *IfcSlab* where a slab may represent both a ceiling for a storey and a floor for another storey on the top of it (for example, a ceiling for the second storey is a floor for the third storey). Geometries of elements that form each *IfcSpace* are checked. If they close a shape (coordinates of starting point is the same as of ending point), the space is stored in *UBMSpace.Closed* class. Otherwise, it is stored in *UBMSpace.Opened* class. It is important here to mention that information of all building elements that form a space is stored in *UBMBuildingElement* class and all boundaries of the room are represented within the classes aggregated under the *UBMBoundarySurface* class. They are connected through the *UBMSpace* class.

It is worthwhile to mention here that for simplification, *BuildingFurniture* and *IntBuildingInstallation* are not included in our model.

Formally, the transformation rules may be written as:

$$\begin{aligned}
 f_{\text{Combination}} &: \Omega_{[\text{IfcSpace}, \text{IfcWall}, \text{IfcRoof}, \text{IfcSlab}]} \rightarrow \Omega_{\text{UBMSpace.Closed}} \\
 f &: \Omega_{\text{IfcWall}} \rightarrow \Omega_{\text{UBMWall.Interior}} \\
 f &: \Omega_{\text{IfcSlab}} \rightarrow \Omega_{\text{UBMCovering.Ceiling}} \\
 f &: \Omega_{\text{IfcSlab}} \rightarrow \Omega_{\text{UBMLevel.Floor}}
 \end{aligned}$$

The first step in this process shows conversion rules for converting IFC building model to different LoDs in the UBM. To demonstrate the use of step1, we have applied the transformation rules on the hospital building case previously discussed. Figure 8 shows a set of building elements conform to the proposed UBM for the Norrtälje Hospital Building at LoD4.

Step2. As a second step of IFC to CityGML conversion we define a set of rules which describe how the objects in the CityGML model can be produced from UBM. Due to space limitations we provide only the set of rules and limited discussion about important rules in the form of a table.

Table 1 shows the transformation rules for step2 of converting IFC to CityGML. On applying these transformation rules, Fig. 9a–d shows the hospital building in CityGML at LoD1 to LoD4 respectively. Figure 9e shows an interior view of the building at LoD4 which represents the interior faces of building elements.

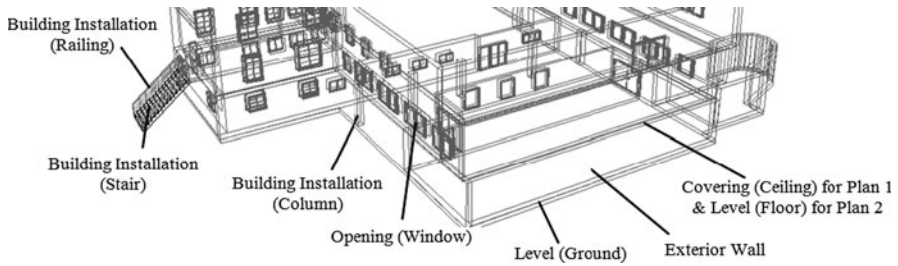


Fig. 8 A part from Norrtälje hospital showing UBM objects

Table 1 Transformation rules from UBM to CityGML

LoD	Rule
LoD1	$f: \Omega_{UBM} \text{Building} \rightarrow \Omega_{CityGML} \text{AbstractBuilding}$
LoD2	$f: \Omega_{UBM} \text{BoundarySurface} \rightarrow \Omega_{CityGML} \text{BoundarySurface}$ $f: \Omega_{UBM} \text{Covering.Roof} \rightarrow \Omega_{CityGML} \text{RoofSurface}$ $f: \Omega_{UBM} \text{Wall.Exterior} \rightarrow \Omega_{CityGML} \text{WallSurface}$ $f: \Omega_{UBM} \text{Level.Ground} \rightarrow \Omega_{CityGML} \text{GroundSurface}$
LoD3	$f: \Omega_{UBM} \text{BuildingInstallation} \rightarrow \Omega_{CityGML} \text{BuildingInstallation}$ $f: \Omega_{UBM} \text{Opening} \rightarrow \Omega_{CityGML} \text{Opening}$ $f: \Omega_{UBM} \text{Door} \rightarrow \Omega_{CityGML} \text{Door}$ $f: \Omega_{UBM} \text{Window} \rightarrow \Omega_{CityGML} \text{Window}$
LoD4	$F: \Omega_{UBM} \text{Space.Closed} \rightarrow \Omega_{CityGML} \text{Room}$ $f: \Omega_{[UBM} \text{BoundarySurface, UBM.BuildingElement}] \rightarrow \Omega_{CityGML} \text{Multisurface}$ $f: \Omega_{UBM} \text{Covering.Ceiling} \rightarrow \Omega_{CityGML} \text{CeilingSurface}$ $f: \Omega_{UBM} \text{Wall.Interior} \rightarrow \Omega_{CityGML} \text{InteriorWall}$ $f: \Omega_{UBM} \text{Wall.Floor} \rightarrow \Omega_{CityGML} \text{FloorSurface}$

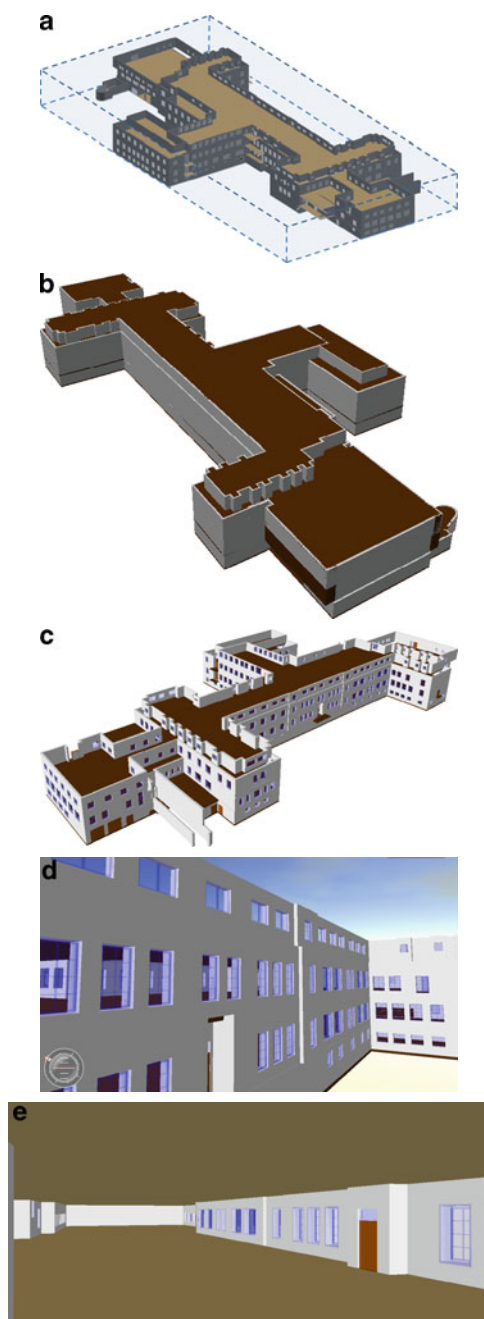


Fig. 9 (a) LoD1-box for the Hospital building. (b) LoD2 for the hospital building. (c) LoD3 for the hospital building. (d) LoD4 for the hospital building – external view. (e) LoD4 for the hospital building – internal view

5.2 Conversion from CityGML to IFC

Similar to the conversion from IFC to CityGML, a two-steps method is also carried out to convert CityGML to IFC. In the first step, UBM is produced from CityGML and then, IFC model is produced from UBM in the second step. Due to space limitations we provide only a set of rules with limited discussion for the important ones in the form of a table.

One can ask here why we need to go from the a very generic level (i.e., LoD1 and LoD2 of CityGML) to a more detailed model (i.e., IFC). However, we argue that these conversions result in models which are represented in their exterior structures. In Architecture, Engineering and Construction (AEC) applications, such generic models can be useful for analysis of testing the prototyping of building structures, templates for architectural interior designs, 3D interior logistics and utility planning, and hence, supporting the unified applications that integrate IFC and CityGML in city modelling applications. We, therefore, start by describing the conversion from CityGML LoD1 and LoD2 to the proposed UBM.

Step1. As a first step of CityGML to IFC conversion, we define a set of rules that describe how the objects in the UBM can partially be produced from CityGML model. The rules are given in the following description.

At LoD1, as mentioned in (Sect. 5.1), a building is presented by a box/blocks model. Information needed for constructing the box model should be stored in the *UBMBuilding* object. This object conforms to the CityGML *AbstractBuilding* object. Different structural entities of exterior surfaces of a building are aggregated to simple boxes without any details. As an alternative, information of the exterior shell of a building can be stored in *CityGMLMultiSurface*. In this case, the information can be transformed to *UBMBuilding* object attributes. Conversion rules may be written as:

$$f : \Omega_{\text{CityGMLAbstractBuilding}} \rightarrow \Omega_{\text{UBMBuilding}}$$

At LoD2, the exterior shell of a building starts to be decomposed into details. *CityGMLBoundarySurface* represents the base class for all other objects that form a building shell. At this LoD, the only needed details are about; the covering of a building and shapes of its roof (from *CityGMLRoofSurface*), exterior walls with their details (from *CityGMLWallSurface*) and the ground floor shape of the building (from *CityGMLGroundSurface*). By applying conversion of BRep to Sweeping/CSG geometric models, information of these building elements are transformed to the *UBMBuildingElement* class and their surfaces are stored in the *UBMBoundarySurface*.

For building installations, the concept in the UBM is very close to that of CityGML. Therefore, we can use one-to-one mapping of the *CityGMLBuilding-Installations* geometries into *UBMBuildingInstallation* geometries. In some cases, building installations are not needed at this LoD, so they can then be omitted from the model. When only external installations are needed, similar two-footprints

process that is discussed in (Sect. 5.1 LoD2 and LoD4) can be used. Only connected objects with these two footprints will be stored as exterior building installations. However, we suggest that in case of small models, all building installations can be added at LoD2. Formally, conversion rules may be written as:

$$\begin{aligned} f &: \Omega_{\text{CityGMLRoofSurface}} \rightarrow \Omega_{\text{UBMCovering.Roof}} \\ f &: \Omega_{\text{CityGMLGroundSurface}} \rightarrow \Omega_{\text{UBMLevel.Ceiling}} \end{aligned}$$

At LoD3, *CityGMLBoundarySurface* holds information about different bounding objects of different spaces. Information about objects that share the same horizontal level in one building storey (or may be based on user definitions) can be aggregated to form *UBMStorey*. In IFC a storey is defined in three types; complex (when it spans overall several connected storeys), element, (when all building storeys are similar in their structure and height) and partial (when it is decomposed in horizontal parts). Considering these types and their definitions, the *UBMBuildingStorey* is geometrically described by all spaces that share the same floor surface. These spaces will be referenced from their boundary surfaces of *CityGMLRoofSurface*, *CityGMLWallSurface* and *CityGMLGroundSurface* which define the external shell of each storey. In LoD4, however, these objects will be supported by the internal objects for more detailed floors.

Opened spaces can be also formed at this level in a condition of a room in CityGML that does not have six bounding objects (four walls or sides of walls, ceiling and a floor) or does not have a closing geometry of its bounding elements. Alternatively, classification of opened and closed spaces can be done manually or to be left for LoD4 as division of open spaces (e.g., dividing a long corridor in different open spaces). At this LoD, opening elements should also appear in the model. Our concept of opening elements is similar to that of CityGML as door and window are subclasses of the class opening. Therefore, one-to-one mapping can be done between *CityGMLOpening*, *CityGMLWindow* and *CityGMLDoor* into corresponding classes in the UBM. Formally, conversion rules may be written as:

$$\begin{aligned} f_{\text{AggregationWithConditions}} &: \Omega_{\text{CityGMLBoundarySurface}} \rightarrow \Omega_{\text{UBMStorey}} \\ f_{\text{Aggregation}} &: \Omega_{[\text{CityGMLBoundarySurface}, \text{CityGMLBuildingElements}]} \rightarrow \Omega_{\text{UBMSpace.Opened}} \\ f &: \Omega_{\text{CityGMLOpening.Door}} \rightarrow \Omega_{\text{UBMOpening.Door}} \\ f &: \Omega_{\text{CityGMLOpening.Window}} \rightarrow \Omega_{\text{UBMOpening.Window}} \end{aligned}$$

LoD4 represents the highest detailed LoD in which all the building elements, exteriors and interiors, should be represented. We have solved the problem of splitting an IFC element (for example a wall) in its exterior and interior faces by adding, in the UBM, exterior and interior classes as subtypes of wall. An opposite process should be carried out to aggregate different faces of a building element (for example wall) to form an IFC element. We, however, propose the generation of two projected footprints as it is introduced above in (Sect. 5.1 LoD2 and LoD4) to

differentiate between exterior and interior elements. Information about building elements themselves is stored in different subclasses of the *UBMBuildingElements* class. In this case conversion from BRep to Sweeping/CSG geometric models are required. However, their surfaces information will be also referenced in the *UBM-BoundarySurface*. For the current step, to convert from CityGML to the UBM, we do not show transformation of all building elements, as objects in CityGML can be matched easily to their corresponding UBM objects, for example *CityGMLCeiling* is matched to *UBMCovering.Ceiling*. Rooms at this level should be converted to closed spaces in the UBM. We have discussed differentiation between opened and closed (room) spaces at LoD3. Formally, rules may be written as:

$$\begin{aligned}
 f &: \Omega_{\text{CityGMLCeilingSurface}} \rightarrow \Omega_{\text{UBMCovering.Ceiling}} \\
 f &: \Omega_{\text{CityGMLInteriorWallSurface}} \rightarrow \Omega_{\text{UBMWall.Interior}} \\
 f &: \Omega_{\text{CityGMLFloorSurface}} \rightarrow \Omega_{\text{UBMLevel.Floor}} \\
 f &: \Omega_{\text{CityGMLRoom}} \rightarrow \Omega_{\text{UBMSpace.Closed}}
 \end{aligned}$$

Step2. As a second step of CityGML to IFC conversion we define a set of rules which describe how the objects in the IFC model can be produced from the proposed model. Due to space limitations we provide only the set of rules and limited discussion about important rules in the form of a table (Table 2).

6 Discussion and Conclusion

IFC and CityGML represent indoor and outdoor spatial objects of a building. In order to fulfil the demands in urban planning applications and construction analysis, it is important to integrate IFC and CityGML. However, existing approaches do not provide complete integration because they mostly offer a unidirectional transformation, i.e., from IFC to CityGML.

There is a significant overlapping for information content for mapping IFC and CityGML to the UBM. However, there is no one-to-one mapping for all data. While there are concepts in the UBM are adopted as in IFC, there are others that are closer or adopted from CityGML definitions. The first step for this mapping is to extract

Table 2 Transformation rules from UBM to IFC

LoD	Rule
LoD3	$f: \Omega_{\text{UBMOpening}} \rightarrow \Omega_{\text{IfcOpening}}$
	$f: \Omega_{\text{UBMOpening.Door}} \rightarrow \Omega_{\text{IfcDoor}}$
	$f: \Omega_{\text{UBMOpening.Window}} \rightarrow \Omega_{\text{IfcWindow}}$
	$f: \Omega_{\text{UBMStorey}} \rightarrow \Omega_{\text{IfcBuildingStorey}}$
	$f: \Omega_{\text{UBMSpace}} \rightarrow \Omega_{\text{IfcSpace}}$
LoD4	$f: \Omega_{\text{UBMCovering.Ceiling}} \rightarrow \Omega_{\text{IfcSpace}}$
	$f: \Omega_{\text{UBMCovering.Roof}} \rightarrow \Omega_{\text{IfcCovering}}$
	$f: \Omega_{\text{UBMFloor}} \rightarrow \Omega_{\text{IfcSlab}}$
	$f: \Omega_{\text{UBMWall.CurtainWall}} \rightarrow \Omega_{\text{IfcCurtainWall}}$
	$f: \Omega_{\text{UBMWall.Interior}} \rightarrow \Omega_{\text{IfcWall}}$

the relevant semantic objects from IFC and CityGML that contain needed information and their geometries. That is followed by identifying the level of detail (LoD) which we target in the UBM. Table 3 presents the corresponding elements for the mapping from IFC and CityGML into the UBM.

In this paper we have presented an approach based on a Unified Building Model (UBM) for reaching interoperability between IFC and CityGML. It is a two-steps method that can be used for converting IFC to CityGML building models and vice versa. By using this approach a building model is first converted to the UBM, followed by conversion to the target building model.

Limitations of our approach are, (a) CityGML to IFC is not fully demonstrated, (b) UBM should also be tested on several buildings, (c) the conversion process is not verified. The identified limitations are also our future research directions.

From the study, we can conclude the following: (a) the approach is a candidate for bidirectional conversion of CityGML and IFC, and (b) the approach provides a starting point towards complete integration of CityGML and IFC. The latter

Table 3 IFC – UBM – CityGML mapping

IFC	UBM	CityGML
IfcBuilding	UBMBuilding	_AbstractBuilding
IfcBuildingStorey	UBMStorey	BoundarySurface RoofSurface WallSurface GroundSurface Other building elements
IfcSpace	UBMSpace	
	UBMOpenedSpace	
	UBMClosedSpace	Room
IfcSlab	UBMLevel	
(Ground Slab)	UBMGround	GroundSurface
(Floor Slab)	UBMFloor	FloorSurface
(Ceiling Slab)	UBMCovering	
	UBMCeiling	CeilingSurface
IfcRoof	UBMCovering	RoofSurface
	UBMRoof	
IfcWall	UBMWall	
(Exterior Wall)	UBMExteriorWall	WallSurface
(Interior Wall)	UBMInteriorWall	InteriorWallSurface
IfcCurtainWall	UBMWall	
	UBMCurtainWall	WallSurface
IfcOpeningElement	UBMOpening	Opening
IfcDoor	IfcDoor	Door
IfcWindow	IfcWindow	Window
IfcBeam	UBMBuildingInstallation	BuildingInstallation
IfcColumn	UBMBuildingInstallation	BuildingInstallation
IfcCovering	UBMBuildingInstallation	BuildingInstallation
IfcStair	UBMBuildingInstallation	BuildingInstallation
IfcRailing	UBMBuildingInstallation	BuildingInstallation
IfcRamp	UBMBuildingInstallation	BuildingInstallation

conclusion represents how different classes, attributes and relations have been considered from IFC and CityGML when building the UBM. Additional constraints are also added in such a way that may lead, with future development, to full semantic integration of the building models.

To illustrate the applicability of our proposed approach, we have used a hospital case study. The purpose of the case study is to present how different building elements at different levels of detail can be rebuilt. Considering future research possibilities, we look to our integration approach as a starting point for developing a common database that formulates a UBM's platform. With such platform, data from IFC and CityGML can be automatically integrated and processed in different analyses. Other formats can also be included in further steps. We finally believe that our approach needs future research efforts beyond only the building models and on implementation process for testing and verification.

Acknowledgement This paper has been prepared as a part of research work in the Future Position X project NYSTA 39686, financially supported by the European Union structural funds, objective 2.

References

- Benner, J., Geiger, A., Leinemann, K.: Flexible generation of semantic 3D building models. In: Gröger, G. et al. (Eds.): Proceedings of the 1st International Workshop on Next Generation 3D City Models, Bonn (2005).
- CityGML homepage, <http://www.citygml.org> [last accessed 01-2010].
- Cox, S., Daisy, P., Lake, R., Portele, C., Whiteside, A.: OpenGIS Geography Markup Language (GML3), Implementation Specification Ver 3.1.0, OGC Doc.No. 03-105r1. 2004.
- Gröger, G., Kolbe, T.H., Czerwinski, A., Nagel, C.: OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Version 1.0.0, OGC Doc. No. 08-007r1, Open Geospatial Consortium (2008).
- Hallberg, D., Tarandi V.: On the use of 4D BIM in LMS for Construction Works, Journal of Information Technology in Construction (Itcon), 2009
- IAI, <http://www.iai-tech.org/> [last accessed 01-2010].
- IFC Model documentation webpage. International alliance for operability, 2008, http://www.iai-international.org/Model/R2x3_final/index.htm [last accessed 01-2010].
- IFCExplorer CityGML Export, http://www.ifcwiki.org/index.php/IfcExplorer_CityGML_Export [last accessed 01-2010].
- IFG, http://www.iai.no/ifg/index_history.html [last accessed 01-2010].
- Isikdag, U., Zlatanova, S.: Towards defining a framework for automatic generation of buildings in CityGML using BIM. In: Lee, J., Zlatanova, S. (Eds.): 3D Geo-information Sciences, LNG&C, Springer-Verlag, pp. 79–96 (2009a).
- Isikdag, U., Zlatanova, S.: A SWOT analysis on the implementation of BIM within geospatial environment. In: Krek, A., Rumor, M., Zlatanova, S., Fendel, E. (Eds.): Urban and Regional Data Management, UDMS Annuals 2009, CRC Press, Boca Raton, FL, pp. 15–30 (2009b).
- ISO 16739 Industry Foundation Classes, Release 2x, Platform specification, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38056 [last accessed 11-2009].
- ISO 19107 Geographic information – spatial schema, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26012 [last accessed 11-2009].
- ISO 19109:2005 Geographic information – rules for application schema, http://www.iso.org/iso/catalogue_detail.htm?csnumber=39891 [last accessed 11-2009].

- ISO TC211, Geographic information/geomatics, <http://www.isotc211.org/> [last accessed 11-2009].
- Jech, T.: Lectures in set theory, Springer-Verlag Lecture Notes in Mathematics 217 (ISBN 978-3540055648) (1971).
- Karola, A., Lahtela, H., Hänninen, R., Hitchcock, R., Chenc, Q., Dajkadand, S., Hagströme, K.: BPro COM-server-interoperability between software tools using industrial foundation classes. *Energy and Buildings* 34(9), 901–907 (2002).
- Kolbe, T. H.: Representing and Exchanging 3D City Models with CityGML, In: Lee J., Zlatanova, S., *Lecture Notes in Geoinformation and Cartography, 3D Geo-Information Sciences*, Springer, Berlin-Heidelberg (2008)
- Kolbe, T.H., Nagel, C., Stadler, A.: CityGML – A framework for the representation of 3D city models from geometry acquisition to full semantic qualification. In: *Proceedings of ISPRS Congress 2008, Beijing, China* (2008).
- Lapierre, A., Cote, P.: 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, Stuttgart* (2007).
- Lapierre, A., Cote, P.: Using open web services for urban data management: a testbed resulting from an OGC initiative offering standard CAD/GIS/BIM services. In: Coors, V., Rumor, M., Fendel, E., Zlatanova, S. (Eds.): *Urban and Regional Data Management, Taylor & Francis, London*, pp. 381–393 (2008).
- Locum, A.B., Sweden’s larger property managers, <http://www.locum.se> [last accessed 01-2010].
- Miguel, M., Jourdan, J., Salicki, S.: Practical Experiences in the Application of MDA, Fifth International Conference on the Unified Modeling Language – the Language and Its Applications, Dresden, Germany (2002).
- Nagel C.: Conversion of IFC to CityGML; Meeting of the OGC 3DIM Working Group at OGC TC/PC Meeting, Paris (Frankreich), Juli (2007).
- Nagel, C., Stadler, A., Kolbe, T.: Conceptual Requirements for the Automatic Reconstruction of Building Information Models from Uninterpreted 3D Models, Academic Track of Geoweb 2009 Conference, Vancouver (2009).
- OGC, <http://www.opengeospatial.org/> [last accessed 01-2010].
- Safe Software, FME Desktop Translator/Converter Software, <http://www.safe.com/products/desktop/formats.php> [last accessed 01-2010].
- Van Berlo, L.: CityGML extension for Building Information Modelling (BIM) and IFC. Free and Open Source Software for Geospatial (FOSS4G), Sydney (2009).