

# An Evaluation of IFC-CityGML Unidirectional Conversion

Mohamed El-Mekawy a

Department of Computer and  
Systems Sciences,  
Stockholm University, Sweden

Anders Östman b

Faculty of Engineering and  
Sustainable Development,  
University of Gävle, Sweden

Ihab Hijazi c

An-Najah National University,  
Nablus,  
Palestinian

**Abstract**—Interoperability between building information models (BIM) and geographic information models has a strong potential to bring benefit to different demands in construction analysis, urban planning, homeland security and other applications. Therefore, different research and commercial efforts have been initiated to integrate the most prominent semantic models in BIM and geospatial applications. These semantic models are the Industry Foundation Classes (IFC) and City Geography Markup Language (CityGML) respectively. However, these efforts mainly: a) use a unidirectional approach (mostly from IFC to CityGML) for converting data, or b) Extending CityGML by conceptual requirements for converting CityGML to IFC models. The purpose of this paper is to investigate the potential of unidirectional conversion between IFC and CityGML. The different IFC concepts and its corresponding concepts in CityGML is studied and evaluated. The investigation goes beyond building objects, also including other concepts that are represented implicitly in building schemas such as building objects relations, hierarchies of building objects, appearance and other building characteristics. Due to the large semantic differences between IFC and CityGML standards, the schema mapping is based on a manual pragmatic approach without automatic procedures. The mappings are classified into three categories, namely ‘full matching’, ‘partial matching’ and ‘no matching’. The result of the study shows that only a few concepts are classified as ‘direct matching’, a few as well as ‘no matching’ while most of the building concepts are classified as ‘partial matching’. It is concluded that unidirectional approaches cannot translate all the needed concepts from both IFC and CityGML standards. Instead, we propose a meta-based unified building model, based on both standards, which shows a high potential for overcoming the shortages of the unidirectional conversion approaches.

**Keywords**-IFC; CityGML; UBM; Evaluation; Unidirectional.

## I. INTRODUCTION

Sharing and exchanging spatial information in various disciplines has been a major driving force behind the development of spatial technology and applications in the last decade (Isikdag and Zlatanova, 2009b). 3D city modeling applications is one of the most important products of this development. However, from a technical perspective, building 3D city applications requires common communication standards between design models (focusing on the applications of building industry and fulfilling the requirements of architecture, engineering and construction (AEC) industry) and real-world models (as geospatial information systems that

model the built environment around us with their spatial objects) (Pu and Zlatanova, 2006; van Oosterom et al., 2006).

Within the building industry, on the one hand, building information modeling (BIM) has been developed as a research area looking mainly at problems related to information integration and interoperability at the building industry domain (Barrett and Grobler, 2000). Industry Foundation Classes (IFC) has then been developed as a reference model standard for the building industry (IAI, 2011). The IFC standard does not just represent and model building components. It further supports different advanced processes and analyses based on spatial relations among these components. These processes can be scheduled in time for different activities. Different objects are represented by database entities that are characterized by properties such as name, geometry, materials, and so on (Khemlani, 2004). In the geospatial information field, on the other hand, CityGML has been developed as a geospatial model standard that represents geometric as well as entities and non-spatial relationships among entities. With its implementation as an application schema for Geography Markup Language 3 (GML3), CityGML is considered more appropriate for outdoor modeling in different levels of detail (Kolbe and Gröger, 2004). Considering their ability for modeling spatial objects with respect to entities and geometric and non-spatial characteristics, IFC and CityGML are seen today as the two most prominent semantic models for representation of design and real world city objects (Isikdag and Zlatanova, 2009b).

A great deal of research is focused on ways for exchanging of information and bringing IFC and CityGML together towards a unified 3D city model (Kolbe and Bacharach, 2006; Isikdag and Zlatanova, 2009b; Van Berlo and De Laat, 2011). However, these research efforts are mainly achieving integration by two different approaches, namely

1. Transforming IFC building models into CityGML (Nagel, 2007) or generating buildings in CityGML using IFC semantics and components (Isikdag & Zlatanova, 2009b).
2. Extending CityGML by conceptual requirements for converting CityGML to IFC models (Nagel et al., 2009) or using Application Domain Extensions (ADE) (Van Berlo, 2009; Van Berlo and De Laat, 2011) which provide a way to represent the information that is not possible to be presented using the current CityGML classes.

Additional to these efforts, commercial software products and conversion tools from IFC to CityGML such as IfcExplorer (IfcExplorer, 2010) and FME (Safe Software, 2010) have been developed mainly following the first approach where IFC concepts are transformed and forced to be represented by CityGML classes.

The first approach to achieve integration is based on unidirectional transformation of IFC building models into CityGML. It is argued to remain the only valid and more relevant to the goal of integrating and analyzing BIM information in geo-context (Isikdag & Zlatanova, 2009b). This method is also found to be dominant in the IFC-CityGML integration literature.

The second approach to achieve integration, as described above, focus on extending the CityGML model. There are two main methods for extending the CityGML model inside an application. The first method is through introducing generic city objects and attributes, while the second is through application domain extensions (ADEs). The first extension method has some clear limitations. For instance, since additional data types cannot be introduced, there are limited possibilities of introducing generic attributes. In addition, XML parsers have difficulties to validate the layout and occurrences of generic objects and attributes. Although, the second extension method (through ADEs) overcomes the limitation of the first method, other general disadvantages of the extension approach is highlighted, such as: i) moving the details of IFC by extending the CityGML schema results in huge CityGML file size that limit the usability of different 3D city applications, ii) IFC include private (as well as irrelevant) information for a city model. Integrating them within public city models is a problematic issue and iii) the two standards are developed by two different communities that have different goals and requirements. As a consequence, extending one standard to replace the other is not feasible, if both communities and target groups still are to be addressed.

As a third approach, El-Mekawy et al. (2011) proposed a unified building model (UBM) for the integration of IFC and CityGML. The UBM is based on reference ontology and meta-concepts and it aims to act as a mediator in the transformation process. Although the development of the UBM is not fully implemented yet, it shows promising results to this area where an increasing number of indoor and outdoor 3D city models needs a tighter integration.

In order to evaluate the potential of these approaches to data integration, the limitations of each method must be analyzed. Since the second method have general limitations in its feasibility and since the third method yet is not well developed, they are excluded from the analysis. As a consequence, this paper aims to investigate the potential of unidirectional conversion between IFC and CityGML. Different IFC concepts and their corresponding concepts in CityGML are studies and evaluated. The investigation goes further beyond building objects, also including other concepts that are represented implicitly in building schema such as relations between building objects, building hierarchies, appearance and other building characteristics. The purpose is to provide a quantitative analysis and validation of the currently most

common unidirectional conversion approaches that are utilized for this type of data integration.

The paper is structured as follows: Section 2 provides an overview of IFC and CityGML, its concepts and modeling paradigms. Section 3 presents the schema matching and mapping approach between the two domains. In section 4 we present the results of the mapping from IFC building to CityGML building model which is organized in two subsections: product extension and shared building elements. In section 5 the research is concluded and direction for further research is provided.

## II. BACKGROUND IFC AND CITYGML

### A. Industry Foundation Classes (IFC)

IFC is defined as an object oriented specification for exchanging, sharing and re-using information throughout the building industry's life-cycle. It has been initially developed since 1996 by the International Alliance for Interoperability (IAI) and it is still under ongoing process of development by different stockholders (IAI, 2008). The aim of IFC development is to facilitate interoperability in the building industry and sharing information among different participants and stakeholders. It is therefore used to assemble computer readable models that contain data elements and represent parts of buildings with their relevant information. Currently, a wide area of AEC/FM domains (such as architectural design, engineering, building analysis, HVAC, energy simulation, thermal analysis, maintenance and cost estimation) are covered by IFC compliance software vendors (IAI, 2011; Lapierre and Cote, 2008).

It is believed today that IFC is the data model that has widest scope for enabling interoperability within the AEC/FM industry (IAI, 2011). Additionally, with its extensible representation, it is being rapid growing for specific applications in construction, manufacturing and operation tasks within the AEC/FM industry. However, there is no universally accepted building model for IFC yet (Kolbe et al., 2008; Kiziltas et al., 2010). Therefore, primarily based on the work done by the IAI and ISO in form of IFC standard documentation (IAI, 2008), the ISO 16739 standard (ISO 16739) and Benner et al., (2005), an IFC building model is presented in figure 1. Important concepts from those efforts were highlighted and then relationships between them were built. Figure 1 show the proposed IFC building model based on the UML standard notations.

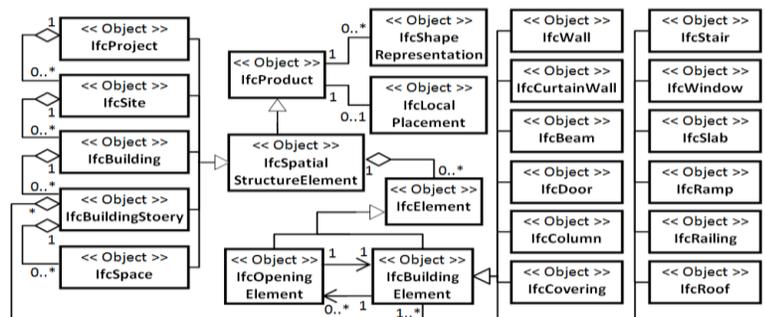


Figure 1. IFC Building Model

Several aspects regarding the IFC schema need to be highlighted. They are discussed as following:

- There are in total about 900 different classes. However most of them have an important role just for defining and linking spatial relationships between objects and their geometrical representation. Berlo (2009) found, in his comprehensive theoretical research, that only 60 to 70 classes out of the 900 can be transformed to GIS. Similarly, most of these classes are used for distributed elements (e.g. heating, ventilation, air conditioning, electrical and equipment elements) and structural calculations (e.g. *IfcStructuralCurveConnection*, *IfcStructuralLinearAction*, *IfcStructuralPointAction*). Consequently, only few are used for the representing the building and architectural elements.
- The structure of IFC is built to support dynamic models. This means more flexibility for users and developers to represent their building data. There are main core elements and containers of elements that store information about building elements in general. All other objects can then be created by using the core elements. This flexibility in IFC structure also explains that there is usually more than one way to connect two different building elements or objects in IFC. *IfcWindow* for example is placed in an opening element (*IfcOpening*) which is connected to a building element such *IfcWall*. The wall surrounds and connected to a space (*IfcSpace*) and the space is connected to the building (*IfcBuilding*). This route in the IFC structure that represents the relationship from an *IfcWindow* towards the *IfcSpace* in which it is located is a user-defined in a specific data file. However, it is not stored in the schema itself. Therefore, the route can be then defined as (*IfcWindow* – *IfcSpace* – *IfcBuildingStorey*) or (*IfcWindow* – *IfcBuildingStorey* – *IfcSpace*) if for example a space is expanded for more than one storey for the latter route.
- IFC define geometric shape definition of building objects explicitly by solid representation i.e. extrusion or faceted boundary representations. The semantics meanings of entities are directly mapped in IFC, for example the name of *IfcWall*, *IfcBeam* or *IfcColumn* implies its semantic meanings and its explicit geometric representation.

### B. City Geography Markup Language (CityGML)

CityGML is an OGC standard, which provides a specification for the representation of 3D urban objects (OGC, 2011). It is the only 3D information model for the exchange of 3D city models. One of the reasons for creating such a model was to enrich 3D city models with thematic and semantic information. The information model of CityGML is an XML-based format implemented as an application schema of Geography Markup language (GML3). The data model has the following features (Becker et al., 2010).

- CityGML seems to provide the best framework for semantic-geometric relations of 3D objects above earth surface (Emgard and Zlatanova, 2008; Groneman and Zlatanova, 2009). It maintains a good taxonomy and aggregations of Digital Terrain Models, sites (including buildings), vegetation, water bodies, transportation facilities, and city furniture. The underlying model differentiates five consecutive levels of detail (LoD), where

objects become more detailed with increasing LoD regarding both geometry and thematic differentiation. In LoDs 2-4 of CityGML the building facade is defined in the form of boundary surfaces, i.e. *WallSurface*, *Roof Surface*, *Ground Surface* or *Closing surface*. The LoD4 allows the representation of interior building elements, e.g. rooms, furniture, interior wall surfaces.

- In CityGML, the structure of linking building elements is fixed /static. For instance, the window (which is a subclass of the opening class) should be connected to the opening class, the opening is connected to a *WallSurface*, and the *WallSurface* is connected to *Room*.
- CityGML defined geometric representation of building and building objects implicitly. For example a wall is defined by different attributes that define each separate wall surface. They are then combined and their geometric definitions form a complete wall. Another example is beams and columns. They are defined by attributes that define their constructing surfaces and then the surfaces are combined geometrically in the class *IntBuildingInstallation* as interior building installations.

Conversely to the IFC, a CityGML Building Model has been produced in the CityGML standard (Kolbe, 2008). The building model (shown in Figure 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 describing the CityGML building model.

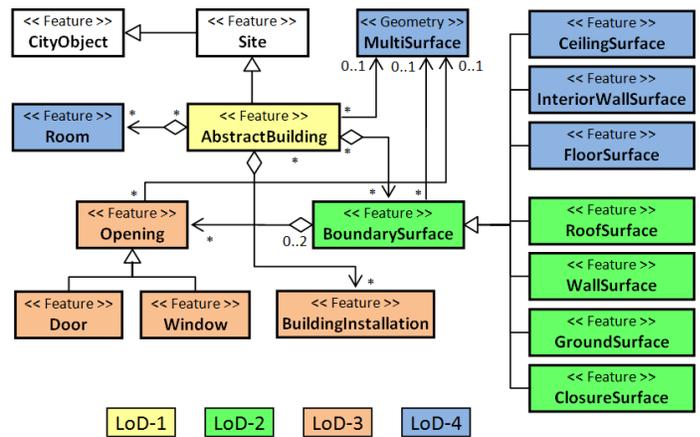


Figure 2. CityGML Building Model

### III. SCHEMA MATCHING AND MAPPING OPERATIONS

A schema transformation is characterized by the mapping functions that specify the transformations from one domain to another. If all information in the target schema can be derived from the information in the source schema, mapping functions may be specified that ensures a complete transformation fulfilling the needs of the target schema. In order to investigate the potential of an undirected transformation between data in IFC and CityGML, the properties of the schema matching and schema mapping operations are studied. The matching between IFC and CityGML, is here based on a pragmatic and manual approach. No automatic mapping procedure or matching between ontologies has been made. This is mainly due to the

large semantic differences between both standards. The aim of the schema matching is to specify which entities and attributes correspond to each other. The focus of this paper is on building objects and their relationships as specified by LoD4 in CityGML. In this context, the entities of the IFC Interoperability Layer – specifically shared building element – and core layer – specifically product extension – constitute the starting point for this study of the direct conversion process.

The shared building elements are the major elements, which constitutes the architectural design and the structure of a building. The building elements (e.g. wall, beam, column, slab, roof, stair, ramp, window, door and covering) are the main components of the raw building (or carcass) which is central for the exchange of project data. The Product Extension further specialize the concepts of a (physical) product, i.e. a component likely to have a placement within the project context. The spatial project structure is introduced within the *IfcProductExtension*. This structure defines the site (the land area where the building is constructed), the building or building parts, the building storey and the space and its decomposition structure. Relationships between the spatial structure and the building elements are defined, such as spatial containment and space boundaries. Other relationships like building elements, opening elements, furnishing elements, distribution elements (including heating, ventilation, air conditioning, electrical and equipment elements) and transportation elements are also defined.

The schema matching operation investigates which entities and attributes correspond to each other. Additionally, it investigates the semantic meaning of different building objects and the associated loss of information based on the direct transformation. The schema mapping operations investigate the relationships and hierarchies between different building objects and how they can be mapped from the IFC schema. The mapping is a specification describing how data is to be transformed from one model to another. Specifications of the relationships that cannot be represented or created from the source model will be used as indicators of the quality of the transformation.

- A full (direct) matching is defined as a matching where a single element in the target model (CityGML) can be directly reconstructed from one or several elements in the source model. Unfortunately, such matching is rare.
- A partial matching is defined as matching where a single element in the target model (CityGML) can be partially reconstructed from one or several elements in the source model. Such matching is found in most of the cases.
- A no matching is defined as a matching where a single element in the target model (CityGML) cannot be reconstructed at all from the information provided in any of the elements in the source model.

#### IV. MAPPING

Following the focus of this paper (discussed in Section 3), we keep the evaluation in this section divided in two parts: the product extension and the shared building elements. This gives more insights on both groups and provides modularity of the discussion as a scientific contribution.

#### A. Evaluation of Product Extension

The product extension discipline in IFC defines the spatial structure of buildings and the relationships between the spatial structure of a building and the building elements. The objects in the product extension are the site where the building/s/(parts of buildings) are placed, building as a structure, building storey and space. These objects are then also containing the architectural objects within a spatial structure. The product extension objects specialize the concepts of a physical product which are the components that represent shape and placement of building elements within a project context. There are 50 different classes/entities in the product extension data model (Table 1). These entities define 5 different concepts, namely:

- the spatial project structure (that defines site, the building, the building storey, and the space),
- the element (that has the types of building element, opening element, furnishing element, distribution element and transportation element),
- the grid (defines a constraint placement for elements),
- the port (defines connectivity between elements), and
- the annotation (captures additional annotations to a building model or plan such as explanatory text, dimensioning, etc.).

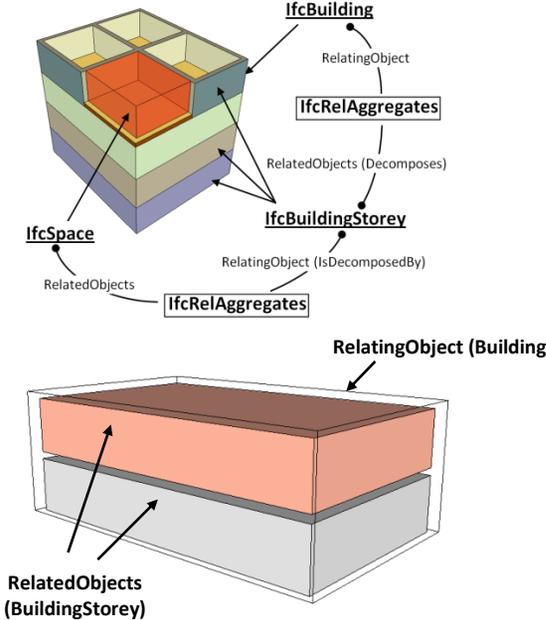
TABLE I. DIFFERENT 50 CLASSES/ENTITIES IN THE PRODUCT EXTENSION DATA MODEL

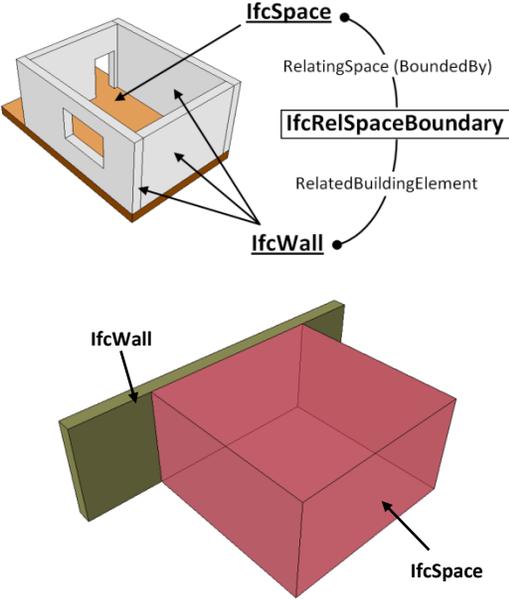
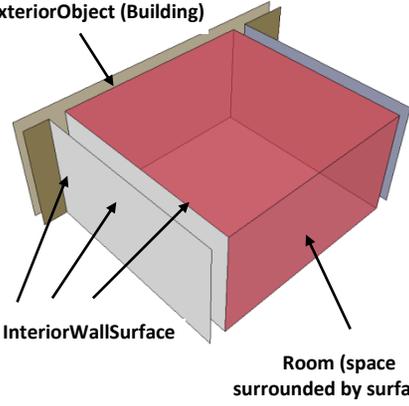
IfcAnnotation	IfcProjectionElement
IfcBuilding	IfcRelAssociatesMaterial
IfcBuildingElement	IfcRelConnectsElements
IfcBuildingElementProxy	IfcRelConnectsPortToElement
IfcBuildingElementProxyType	IfcRelConnectsPorts
IfcBuildingElementType	IfcRelConnectsWithRealizingElements
IfcBuildingStorey	IfcRelContainedInSpatialStructure
IfcCovering	IfcRelCoversBldgElements
IfcCoveringType	IfcRelCoversSpaces
IfcDistributionElement	IfcRelFillsElement
IfcDistributionElementType	IfcRelProjectsElement
IfcElectricalElement	IfcRelReferencedInSpatialStructure
IfcElement	IfcRelServicesBuildings
IfcElementAssembly	IfcRelSpaceBoundary
IfcElementQuantity	IfcRelVoidsElement
IfcElementType	IfcSite
IfcEquipmentElement	IfcSpace
IfcFeatureElement	IfcSpaceType
IfcFeatureElementAddition	IfcSpatialStructureElement
IfcFeatureElementSubtraction	IfcSpatialStructureElementType
IfcFurnishingElement	IfcSystem
IfcFurnishingElementType	IfcTransportElement
IfcGrid	IfcTransportElementType
IfcOpeningElement	IfcVirtualElement
IfcPort	IfcZone

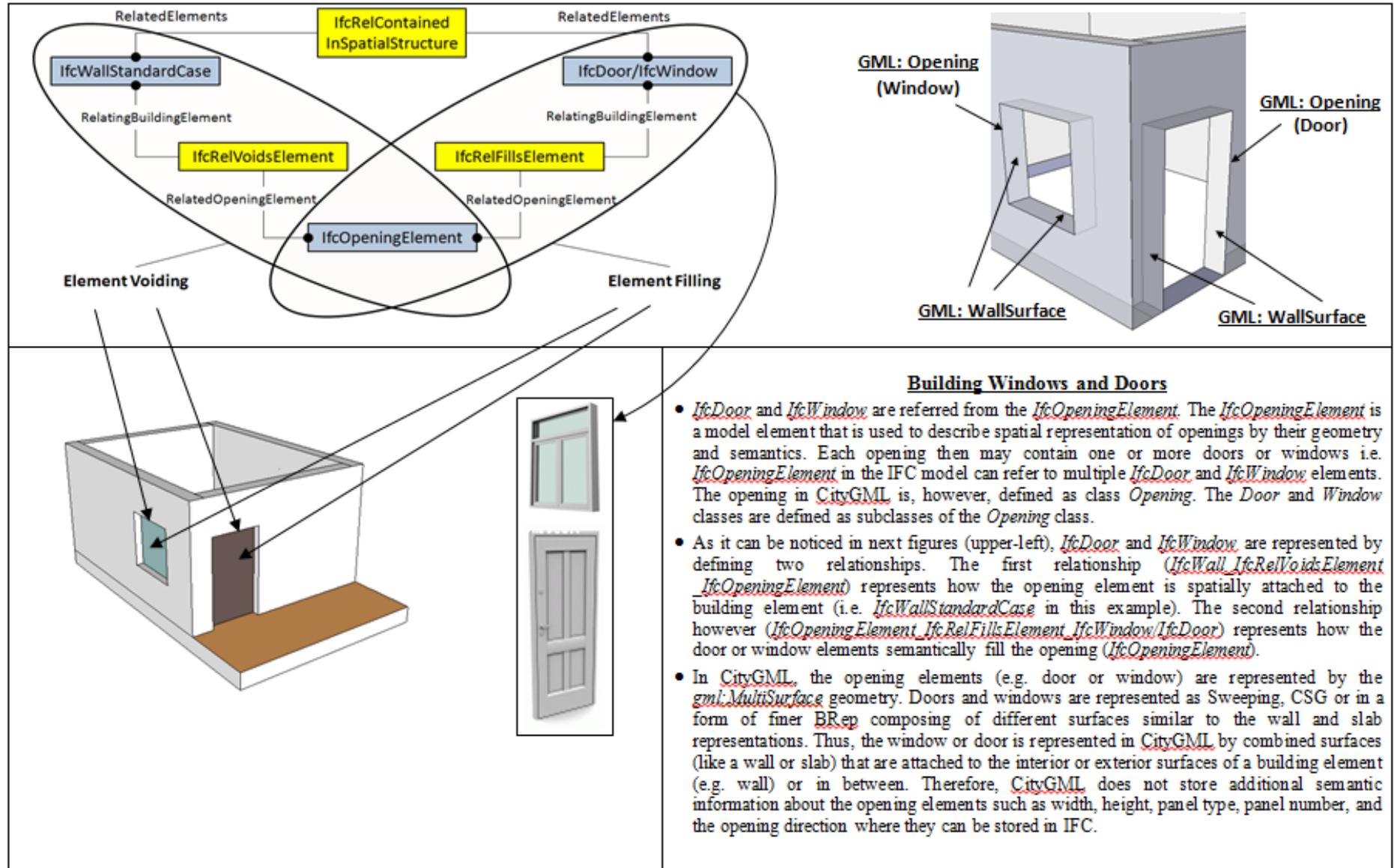
- Concepts
- Structural Objects
- Basic Types of Elements
- Classes defining spatial relationships between building objects.

Most of the 50 entities are used for defining and linking, spatial relationships between building objects and structure calculations. Therefore, not all of them can be used as spatial object in the GIS context. However, some of their properties can be transformed to CityGML attributes. The applicable classes are discussed in the matching below (Table 2).

TABLE II. MATCHING IFC AND CITYGML CLASSES BASED ON THE PRODUCT EXTENSION CONCEPTS

IFC	CityGML	Matching
 <p data-bbox="100 998 504 1031">/* Spatial structure element definitions */</p> <pre data-bbox="100 1055 730 1344"> #4=IFCBUILDING (); #8=IFCBUILDINGSTOREY (); #9=IFCBUILDINGSTOREY (); ..... ..... /*Assignment of two storeys to the building */ #10=IFCRELAGGREGATES ('...', \$, \$, \$, #4, (#8, #9)); ..... ..... </pre>	<pre data-bbox="737 365 1188 1063"> &lt; ?xml version="1.0" ... .. &gt; &lt; bldg:Building xmlns:gml = "... " [...] &gt;   &lt; bldg:Room &gt;     &lt; bldg:BoundedBy &gt;       &lt; bldg:WallSurface &gt;         &lt; bldg:Opening &gt;           &lt; bldg:Window/ &gt;         &lt; /bldg:Opening &gt;       &lt; /bldg:WallSurface &gt;     &lt; /bldg:BoundedBy &gt;   &lt; /bldg:Room &gt;   .....   .....   .....   ..... &lt; /bldg:Building &gt; </pre>	<p data-bbox="1365 365 1617 397"><b><u>Building Structure</u></b></p> <ul data-bbox="1207 406 1770 885" style="list-style-type: none"> <li>• In IFC, a building is smoothly structured by breaking down the building into storeys and into spaces that form a specific storey. However, in CityGML, there is no explicit definition of spaces or stories.</li> <li>• <i>IfcBuildingStorey</i> is defined geometrically as an elevation and typical representation of (nearly or in most cases) horizontally aggregated spaces (<i>IfcRelAggregates</i>) that are separated vertically. Storeys in CityGML can be represented as an explicit aggregation of all building features on a certain height level.</li> <li>• A building in CityGML consists of rooms, where 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 according to arbitrary user defined criteria on a certain height level.</li> </ul>
<p>The matching between these two different structures is seen as ‘no matching’. However, the needed information for defining the target structure can be extracted from entities that represent the hierarchy in the source model. Additionally, in CityGML, the proper way of linking objects in such a structure is strictly defined as (<i>Window, WallSurface, Room, Building</i>) the same all the time. Conversely, the spatial structure in IFC is user-defined and to be stored in a specific data file. It can be described as (<i>IfcWindow, IfcOpening, IfcWall, IfcSpace, IfcBuilding</i>) to highlight different openings and spaces if spaces are the focus of an application or (<i>IfcWindow, IfcStorey, IfcBuilding</i>) if storey’s are in the focus. A ‘no matching’ here is referred to the difficulties of following a single data model of IFC or CityGML for hierarchical structure.</p>		

 <pre data-bbox="115 787 682 1120"> /* Spatial structure element definitions */ #285=IFCSPACE (); #1000=IFCWALL (); ..... ..... /*Assignment of wall to the space */ #1036=IFCRELSPACEBOUNDARY ('...', #6, \$, \$, #285, #1000, #1037, .PHYSICAL., .INTERNAL.); ..... ..... </pre>		<h3 style="text-align: center;">Building Spaces</h3> <ul style="list-style-type: none"> <li>• The <i>IfcSpace</i> class defines all volumes and areas that are bounded in some way. This includes rooms that are bounded by different building elements. A building in CityGML is represented by rooms representing all type of spaces as semantic objects for modeling the free spaces inside a building.</li> <li>• Space in IFC is geometrically associated to a building storey and may be divided into partial spaces. The space is bounded by related building elements (<i>IfcRelSpaceBoundary</i>) that surround the space. However, in CityGML, rooms are not objects in itself. Instead they are represented by boundary surfaces that topologically close the rooms.</li> <li>• The <i>IfcSpace</i> is geometrically represented with multiple representations as usually a combination of Sweeping, CSG and rarely in a BRep. Rooms in CityGML are, however, usually geometrically represented by <i>GML:Solid</i> or <i>GML:MultiSurface</i>.</li> </ul>
<p>Although the differences between space concepts in both IFC and CityGML are apparent, there is a ‘partial matching’ between them. In IFC, there is no concept of specific <i>Room</i> object as in CityGML. <i>IfcSpace</i> class defines all volumes and areas that are bounded. This definition includes rooms that are bounded by different building elements but also other bounded volumes such as corridors etc. As all IFC objects, Sweeping/CSG geometry is used for spaces and their elements which requires conversions to BRep geometric models that are used in CityGML. To create spaces in CityGML, information from <i>IfcWall</i>, <i>IfcRoof</i> and <i>IfcSlab</i> that form the boundaries of rooms are used. Information about both <i>Ceiling</i> and <i>Floor</i> in CityGML can be acquired from <i>IfcSlab</i> 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 2<sup>nd</sup> storey is a floor for the 3<sup>rd</sup> storey).</p>		



This is the only case that shows direct matching where a CityGML feature (Window or Door) can be reconstructed from a single IFC element (*IfcWindow* or *IfcDoor*). The CityGML opening (window or door) can be reconstructed from the element voiding relationship (*IfcRelVoidsElement*) and the surfaces of a window or door can be then reconstructed from the element filling relationship (*IfcRelFillsElement*).

The two standards, IFC and CityGML, represent both 3D semantic models of buildings. They are both object oriented and represent the building objects (e.g. slabs, walls, spaces) based on their semantic rules. The data models for the IFC and CityGML standards are associated with detailed 3D geometry. The data models provide a clear hierarchy for organizing the representation of building structure.

In IFC, the entire spatial structure is subsumed from *IfcProject* class which is the uppermost container of all building information. Under the *IfcProject*, there are two mandatory levels should be defined, *IfcBuilding* and *IfcBuildingStorey*.

There are other optional classes as levels in the hierarchy that can be subsumed from the project. They are *IfcSite*, *IfcBuildingSection* and *IfcSpaces*. The site may contain zero or several (0 .. \*) buildings.

A building has at least one storey (1 .. \*). Each building storey may have zero or more spaces (0 .. \*). All building elements are assigned to the building storey in which they are located. If building elements (or spaces) span through several storeys, then they are assigned to the lowest storey among them in which they are based. CityGML in LoD4, however, provides a different hierarchy for the building structure than the IFC. A building in CityGML is composed of rooms which are enclosed by surrounding surfaces. Storeys are not explicitly defined in the structure. However, they can be modeled using the CityGML generic grouping mechanism.

In IFC, the users can define their own structure for representing different spaces of a building. Users can generate their route of connections with attached clarification regarding the reference service systems in buildings in the structure starting from a wall or slab and linking them to a space then storey the and ending by a building. There is, however, no unique way to connect a specific IFC object to another.

The user-defined route of connections is something we do not find in the IFC schema. Instead, they are stored in a specific data file (remember the optional levels such as *IfcSpace*). For example, *IfcFlowSegment* can be used for defining the containment relationship between *IfcWall* and *IfcSpace* based on the information that spaces are enclosed by walls. Similarly, *IfcSpace* can be spatially related to *IfcStorey* and then to *IfcBuilding*.

These kinds of spatial connections are defined in CityGML as more static and explicit. The relationship between *Window*, *WallSurface*, *Room* and *Building* is always the same. However, the wall class as a building element does not exist in CityGML. It is therefore, difficult for software implementations to transform data from IFC to CityGML. Supported by the claims of Van Berlo & De Laat (2011), this can also be a clear reason for why not all the 50 entities with their properties in the IFC Product Extension discipline can be transformed to CityGML model.

### B. Evaluation of Shared Building Elements

In the shared building elements schema of IFC, most of the architectural design elements of a building are defined. These elements include wall, beam, column, slab, floor, roof, stair,

ramp, window, door and covering. It is worthwhile here to say that the reason of naming these elements as the shared building elements is their use and involvement in other disciplines of the IFC schema and buildings in general.

There are 33 different classes/entities in the shared building elements schema (Table 3). These entities define the subtypes of *IfcBuildingElement* which is defined in the *IfcProductExtension*. The architectural design of a building structure is constituted by those subtypes.

TABLE III. DIFFERENT 33 CLASSES/ENTITIES IN THE SHARED BUILDING ELEMENTS DATA MODEL

IfcBeam	IfcRampFlight
IfcBeamType	IfcRampFlightType
IfcColumn	IfcRelConnectsPathElements
IfcColumnType	IfcRoof
IfcCurtainWall	IfcSlab
IfcCurtainWallType	IfcSlabType
IfcDoor	IfcStair
IfcDoorLiningProperties	IfcStairFlight
IfcDoorPanelProperties	IfcStairFlightType
IfcDoorStyle	IfcWall
IfcMember	IfcWallStandardCase
IfcMemberType	IfcWallType
IfcPlate	IfcWindow
IfcPlateType	IfcWindowLiningProperties
IfcRailing	IfcWindowPanelProperties
IfcRailingType	IfcWindowStyle
IfcRamp	

■ Building Elements

□ Classes that define spatial characteristics and representations of building objects

Similar to the *IfcProductExtension*, many of these 33 entities are used for defining spatial characteristics and representations of building objects.

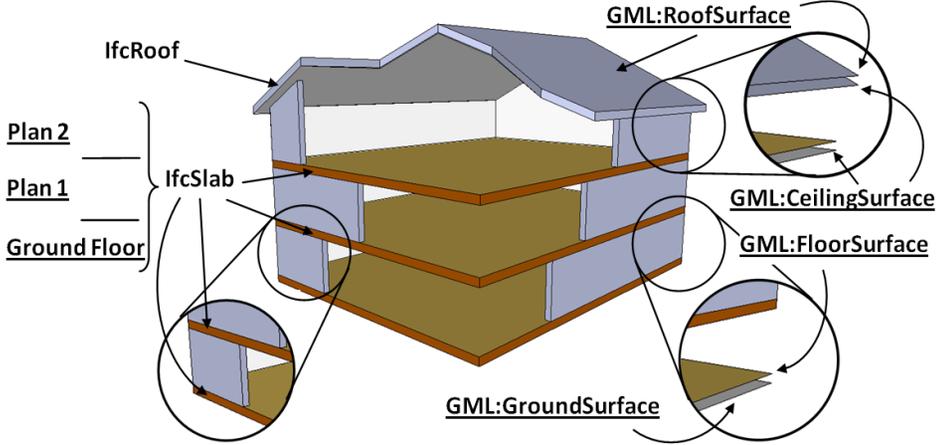
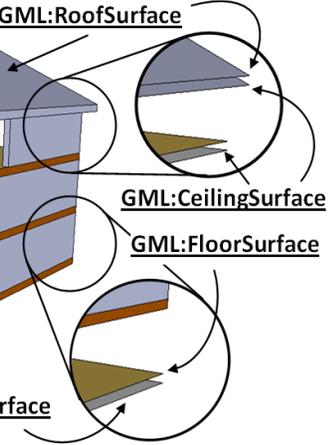
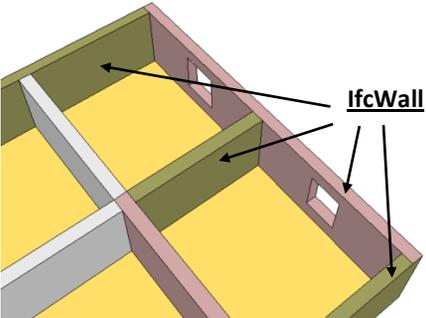
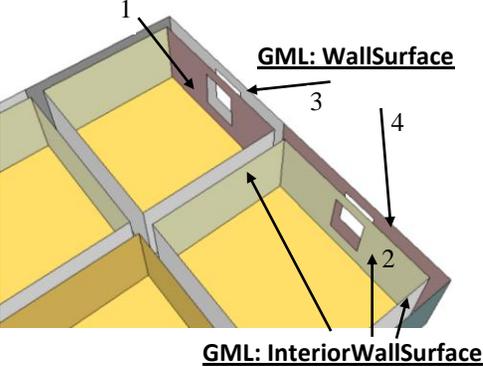
Therefore, not all of them can be used in the GIS context, and hence, the applicable classes are discussed in the mapping below (Table 4).

## V. DISCUSSION AND CONCLUSIONS

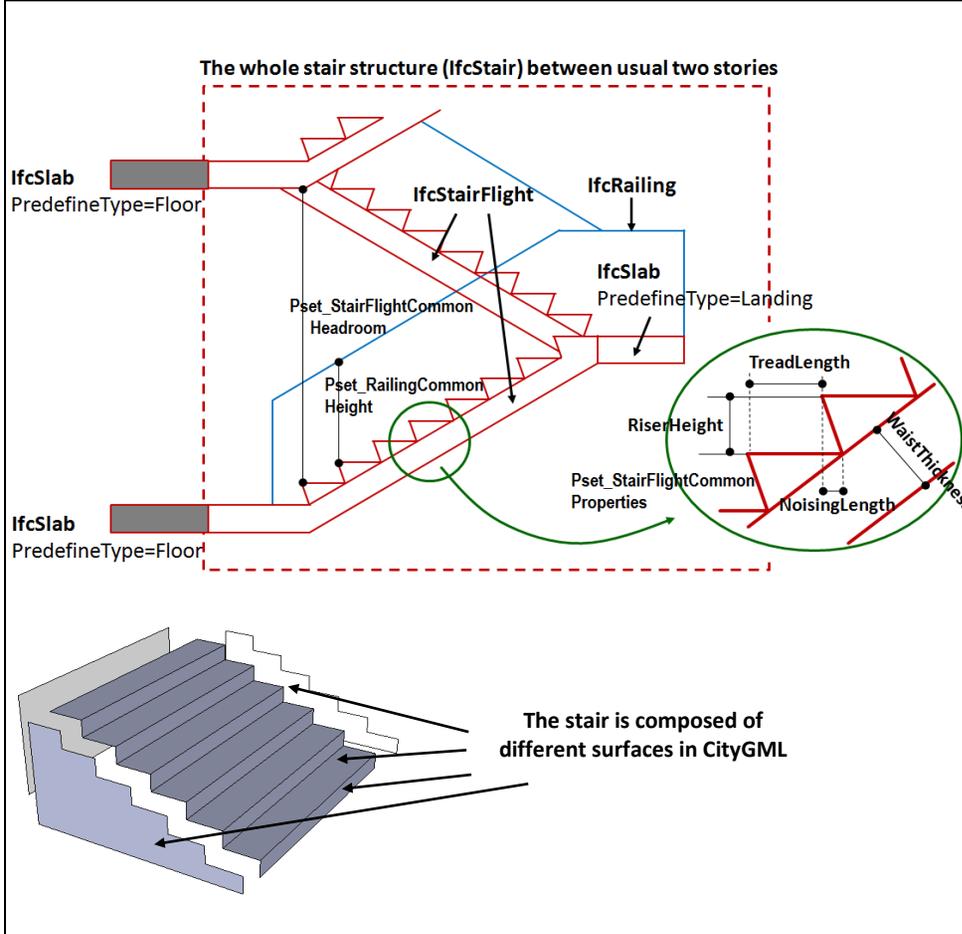
The potential of the unidirectional conversion between IFC and CityGML is mainly focused on the investigation of how much information in the BIM (represented by IFC) can be transferred to the GIS context (represented by CityGML). The reason for that is the very rich semantic model of IFC which still has more detailed schemas than the CityGML (Van Berlo & De Laat, 2011; Isikdag & Zlatanova, 2009b) for modeling specific buildings. Additionally, this has been shown in most of the unidirectional efforts transforming IFC building models to CityGML (e.g. Nagel, 2007; IFG, 2007; Isikdag & Zlatanova, 2009b).

The results of our paper shows that the information provided in IFC that can be transferred to GIS is mainly included in the *IfcProductExtension* and *IfcSharedBuildingElement* domains. Combined, they have 83 classes/entities in their schemas, 50 for the *IfcProductExtension* and 33 *IfcSharedBuildingElement*.

TABLE IV. MATCHING IFC AND CITYGML CLASSES BASED ON SHARED BUILDING ELEMENTS CONCEPTS

IFC	CityGML	Matching
		<p><b><u>Building Slabs and Coverings</u></b></p> <ul style="list-style-type: none"> <li>• <i>IfcRoof</i> and <i>IfcSlab</i> represent complete building elements as roofs and slabs respectively. <i>IfcRoof</i> represents the total roof and acts as a container for the all parts of the roof aggregated by the <i>IfcRelAggregates</i> relationship. <i>IfcSlab</i> represents a constructed component that encloses a space vertically in most of the cases. For any space in the building, it may provide the lower construction of a space (floor) or the upper one (roof slab).</li> <li>• The flooring and roofing slabs in CityGML are classified in different types; <i>RoofSurface</i>, <i>CeilingSurface</i>, <i>FloorSurface</i> and <i>GroundSurface</i> (Figure XXX). However, all of them are represented by their split, but geometrically connected surfaces in which different building spaces are surrounded by these surfaces (e.g. one slab has its lower surface as a <i>CeilingSlab</i> for a building space in Plan N, and its upper surface as a <i>FloorSurface</i> for a building space in plan N+1).</li> <li>• <i>IfcRoof</i> and <i>IfcSlab</i> are geometrically solid representations of a roof and slab respectively where such building element exists. However, in CityGML, the surfaces (<i>Roof</i>, <i>Ceiling</i>, <i>Floor</i> and <i>Ground</i>) represent the visible parts of the building elements from inside the space.</li> </ul>
<p>The matching case in the above mentioned elements is seen as a ‘partial matching’. We deal in these examples with 1:m (one-to-many) matching relationships. For instance, <i>IfcSlab</i> is the only source information for reconstructing the CityGML Ceiling and Floor surfaces i.e. 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 2<sup>nd</sup> storey is a floor for the 3<sup>rd</sup> storey). The ceiling and floor surfaces in CityGML are then reconstructed from different slabs in IFC. Looking from the other side, an <i>IfcSlab</i> represented the ground floor of a building can be reconstructed by combining information from both CityGML Floor and Ground surfaces representing n:1 (many-to-one) matching relationships.</p>		
		<p><b><u>Building Walls</u></b></p> <ul style="list-style-type: none"> <li>• <i>IfcWall</i> represents a complete wall as building element. In the CityGML, the wall is split into geometrically connected surfaces in which different building spaces are surrounded by these surfaces.</li> <li>• <i>IfcWall</i> is geometrically a solid representation of a wall volume where the building element exists. In CityGML, the surface <i>WallSurface</i> represents the visible part of the wall from inside the space.</li> </ul>

Similar to the concepts of slabs and ceilings, the “partial matching” is seen visible for mapping the building walls between IFC and CityGML. For example, two or more CityGML surfaces have to be combined to form a single IFC element. For example, in the next figure (CityGML part), the interior wall surfaces 1, 2 and the exterior surfaces 3, 4 represent only one wall object and to be mapped to the corresponding IfcWall elements. This represents n:1 (many-to-one) example from CityGML to IFC. The 1:n can also be seen from the other way around. Furthermore, the relationship n:m is important to be considered for the case of curtain walls. A curtain wall in CityGML is represented as surfaces spanning for the whole wall. However, in IFC, the building is structured into storeys that require walls to vertically respect storey’s boundaries, and hence, each CityGML wall surface is then divided into parts.



### Building Installations

- All spatial objects that can be components of a building architecture or structure are defined in IFC as building elements. However, in CityGML, only the main building structural objects are considered as building elements. Other spatial elements including some structural elements (e.g. columns, beams) are considered in CityGML as internal installations and represented by *IntBuildingInstallation* class.
- *IntBuildingInstallation* class in CityGML represents all the objects inside a building with a specialized function or semantic meaning. However, *IntBuildingInstallations*, which is different from *BuildingFurniture*, are permanently attached to the building structure and cannot be moved. Typical examples of these installations are interior stairs, ramps, railings, beams, columns, pipes and radiators.
- The geometry of *IntBuildingInstallation* objects are represented, as can be noticed from the next figure (lower), in different surfaces that are aggregated and defined by *gml:Geometry* type to build the installation object similar to the building elements (e.g. wall, slab). However, internal installations in IFC are represented as building elements having their own representations with geometrical entities. For the *IfcStair*, the geometric representation is defined by *IfcProductDefinitionShape* that allows multiple geometric representations of the stair components. The way of representing the whole stair is user-defined. It can either be an aggregation of all stair components (then the stair geometry is the sum of the representation of all components within the aggregate) or individually (then independent geometric representations for all components are used separately).
- In CityGML, the *IntBuildingInstallation* objects can either be associated to a specific room or to the complete building/building part. For each *BuildingInstallation* object, there are three optional attributes, class, function and usage. The class attribute defines a general classification for the object. The function and usage attributes define prospectively the nominal and real functions of a building installation. The geometry of the *IntBuildingInstallation* objects are represented by *gml:Geometry* type.

The “partial matching” is clearly observed in all building installations. A clear example for that is the n:1 relationship for building components that spans over other components and penetrate different building spaces. The (stairs, columns) and beams, are examples of these components that spans over different storeys and rooms respectively. These building installations are represented in CityGML as different surfaces. Then the same beam can be seen in different rooms as represented in two thematic building installation objects with separated surface geometries. Columns and stairs are similarly observed from different vertically structured spaces (rooms) in two or more different storeys. Therefore they are also represented in different thematic building installation objects with own geometries. This clearly conforms the results of (Nagel et al., 2009) who suggest that in order to aggregates different CityGML surfaces into one IFC building element, we do not only need the semantic information. Instead, geometric-topological (such as parallelism, perpendicularity, distance, and adjacency) relationships are also important to be analyze

In the *IfcProductExtension*, the 50 classes include mostly entities that are used for defining the structure of how building objects are spatially connected or related. They, however, have important information for defining how the building objects are spatially represented. In addition to that, entities like *IfcBuildingStorey* and *IfcSite* have information that is not used in CityGML which causes clear information loss when converting from IFC model to CityGML model. Table 1 shows all the 50 entities. In our study, we have focused on the transferred IFC entities to CityGML. Out of these 50 entities, it is found that only four IFC classes can be mapped directly to CityGML in either partial or full matching. These four classes are: *IfcBuilding*, *IfcFurnishingElement*, *IfcSpace*, *IfcAnnotation*. *IfcBuilding* can be mapped to *\_AbstractBuilding*. *IfcFurnishingElement* can be mapped to the GML *BuildingFurniture*. The geometry of furniture is usually represented in IFC model as a BRep. The information in *IfcFurnishingElement* can be used to generate the CityGML *BuildingFurniture* and to be represented by the *gml:Geometry*. Although there is a need for geometry conversion from Sweeping/CSG to BRep, the information in *IfcSpace* can be transferred for generating the *Room* object in CityGML. *IfcAnnotation* is usually represented in IFC model as a graphical representation within the geometric context of a building or a project. Annotation adds a meaning to objects in forms of (e.g. text, line drawings, hatching) and they are modeled by the *IfcRelContainedInSpatialStructure* relationship. The information in *IfcAnnotation* can be transferred to the geometry annotation in CityGML.

All the concepts that are related to the product extension of IFC are found to have matching possibilities with CityGML, except the building structure as discussed in Section 4.1.

The relation between the opening elements and the building element e.g. wall, slab is found to be full matching as windows and doors with their opening can be reconstructed in the target model from the information provided in the source model in both sides. The space (from IFC) and room (from CityGML) concepts and their relationship to their bounding objects cannot be transformed fully without a need for additional information. Therefore, it is considered as partial match. The concepts related to the building structure and its hierarchy, no matching is observed as the structure of both IFC and CityGML building models are built on different concepts.

Therefore, the unidirectional approach seems not to be able or having capabilities for supporting the transformation of concepts in the product extension discipline, and substantial amounts of information will get lost through such transformation approach. A mapping summary of the product extension concepts is provided in Table 5.

TABLE V. MAPPING SUMMARY OF THE PRODUCT EXTENSION CONCEPTS

Concept	Investigation Result
Building Structure	No match
Building Spaces	Partial match
Building Windows and Doors	Full match

In the *IfcSharedBuildingElement*, the 33 classes define the subtypes of *Ifcbuildingelement* that are defined in the *IfcProductExtension*. These entities, which are the central for

the exchange of project data, are represented in the Table 6. For each element, a geometric use definition is introduced in order to define the correct application of available shape representation types for the element. However, not all of them can be used in the GIS context as they are used for defining the spatial characteristics, representations of building objects and structure calculations. Nine entities out of the 33 in *IfcSharedBuildingElement* are found to be directly and partially mapped to CityGML. The nine entities are: *IfcBeam*, *IfcColumn*, *IfcDoor*, *IfcRailing*, *IfcRoof*, *IfcSlab*, *IfcStair*, *IfcWall* and *IfcWindow*. Similar to the entities of *IfcProductExtension*, the non-transformed-to-GIS entities have important information for defining the spatial representation of building objects. In addition to that, there are other entities (e.g. *IfcCurtainWall*, *IfcRamp*) that have attributes and information to be used in CityGML building.

Based on the investigation done on the *IfcSharedBuildingElement* data model, all the transformed entities are found to be partial matching. They can be transformed using additional information from the relationships entities/classes or processing of the spatial data (e.g. splitting, aggregating). As a result, a unidirectional conversion between IFC and CityGML implies data loss and database processing. Table 6 provides a summary for this investigation.

TABLE VI. MAPPING SUMMARY OF THE SHARED BUILDING ELEMENTS CONCEPTS

Concept	Investigation result
Building Slabs and Coverings	Partial match
Building Walls	Partial match
Building Installations	Partial match

## VI. CONCLUSIONS

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

The need for integrating indoor IFC information of buildings and outdoor CityGML built-environment information is apparent. However, different findings, shown in this paper, need to be considered while combining data from both standards. These findings form the following conclusions:

- Most of the CityGML objects can be mapped from the IFC model by partial matching, which implies that additional information or complex processing are required in the mapping. One example is slabs in IFC which should be mapped to several different surfaces in CityGML. How to extract parts of a certain slab and map it to the correct class in CityGML is not a straight forward task. Probably manual processing is required to map the correct surfaces and the whole process is not easily automated.
- The IFC standard is flexible in the sense that structures are user-defined. As a consequence, it can accept a variety of 3D building models from different communities. The 3D building models of today are however scattered over different data holders, in the public as well as the private

sector, in different systems, different conceptual models, different data formats, different data schemas, different levels of detail and different quality. Having different user-defined structures in all these different organizations for serving different purposes makes the mapping to CityGML a complex process especially if automation is needed because the main reason for this is that the structures in CityGML are static.

- In order to overcome the limitations of the existing unidirectional approaches, a new approach is required in order to achieve interoperability. It is recognized that both IFC and CityGML have been developed for different purposes. In addition, CityGML was not originally designed to fully comply to the semantics of the IFC standard. Therefore, CityGML needs to be overloaded with additional information to be smoothly matched and integrated with IFC. One solution may be the proposed unified building model (UBM), which is based on reference ontology and meta-concepts, see figure 3.

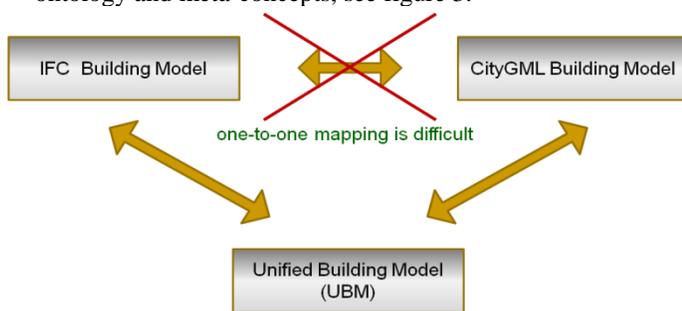


Figure 3. The UBM as a Meta Model

We believe that such an approach might overcome the limitations identified in this paper. By bringing both IFC and CityGML to the UBM in a common model, both standards can be smoothly integrated without a need for conversion. In addition, the number of unified applications, which requires an integration of indoor and outdoor data, may increase without affecting existing users of either model. Additionally, the UBM might have the potential to be extended to support applications that neither IFC nor CityGML can support.

#### REFERENCES

- [1] Barrett, D. and Grobler, F., 2000. Understanding the different purposes of IFCs and aecXML in achieving Interoperability. <http://www.iaina.com> [last accessed Feb-2012].
- [2] Becker, T., Nagel, C. and Kolbe, T., 2010. UtilityNetworkADE,-Core Model-Draft version. <http://www.citygmlwiki.org/index.php/CityGMLUtility-NetworkADE> [last accessed Feb-2012]
- [3] Benner, J., Geiger, A. and Leinemann, K. 2005. 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. [www.eurosd.net/km\\_pub/no49/workshops\\_docs/citymodels\\_june\\_05/paper03\\_benner.pdf](http://www.eurosd.net/km_pub/no49/workshops_docs/citymodels_june_05/paper03_benner.pdf) [last accessed Feb-2012]
- [4] El-Mekawy, M. Östman, A. and Shahzad, M.K., 2011. Towards Interoperating CityGML And IFC Building Models: A Unified Model Based Approach, 3D Geoinfo Conference, Berlin, Germany, Lecture Notes in Geoinformation and Cartography series (Springer-Verlag, Heidelberg).
- [5] Engard, KL. and Zlatanova, S., 2008. Design of an integrated 3D information model. In: Coors, Rumor, Fendel & Zlatanova S.(eds.),

- Urban and regional data management: UDMS annual 2007 (pp. 143-156), Taylor & Francis Group, London, UK.
- [6] Groneman, A. and Zlatanova, S., 2009. TOPOSCOPY: a modelling tool for CITYGML. In: Onsrud H, R van de V (Eds.), GSDI Association, pp. 1-13.
- [7] IAI, 2011. BuildingSMART. <http://www.iai-tech.org/> [last accessed Feb-2012].
- [8] IAI, 2008. IFC Model documentation webpage. International Alliance for Operability. [http://www.iai-international.org/Model/R2x3\\_final/index.htm](http://www.iai-international.org/Model/R2x3_final/index.htm) [last accessed Feb-2012].
- [9] IfcExplorer 2008. IfcExplorer CityGML Export. [http://www.ifcwiki.org/index.php/IfcExplorer\\_CityGML\\_Export](http://www.ifcwiki.org/index.php/IfcExplorer_CityGML_Export) [last accessed Feb-2012].
- [10] Isikdag, S. and Zlatanova, S., 2009b. Towards defining a framework for automatic generation of buildings in CityGML using BIM, in Lee and Zlatanova (eds.), 3D geo-information sciences, LNG&C, Springer Verlag, pp. 79-96.
- [11] ISO 16739, 2011. Industry Foundation Classes, Release 2x, Platform Specification. [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=38056](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38056) [last accessed Feb- 2012].
- [12] ISO 19107. 2011. Geographic information -- Spatial schema. [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_detail.htm?csnumber=26012](http://www.iso.org/iso/iso_catalogue/catalogue_detail.htm?csnumber=26012) [last accessed Feb- 2012].
- [13] ISO 19109. 2011. Geographic information -- Rules for application schema. [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=39891](http://www.iso.org/iso/catalogue_detail.htm?csnumber=39891) [last accessed Feb- 2012].
- [14] Khemlani, L., 2004. Analysis, Research and Review of AEC Technology. <http://www.aecbytes.com/feature/2004/IFCmodel.html> [last accessed Feb-2012].
- [15] Kiziltas, S., Leite, F., Akinci, B. and Lipman, R. 2009. "Interoperable Methodologies and Techniques in CAD", CAD and GIS Integration, Auerbach Publications. ISBN 978-1-4200-6805-4
- [16] Kolbe, T., 2008. Representing and Exchanging 3D City Models with CityGML, In: Jiyeong Lee and Sisi Zlatanova, Lecture Notes in Geoinformation and Cartography, 3D Geo-Information Sciences, Springer Berlin Heidelberg.
- [17] Kolbe, T. and Bacharach, S., 2006. CityGML: An Open Standard for 3D City Models, <http://www.directionsmag.com/articles/citygml-an-open-standard-for-3d-city-models/123103> [last accessed Feb- 2012].
- [18] Kolbe T, Gröger G., 2004. Unified Representation of 3D City Models. Geoinformation Science Journal, Vol.4, No.1.
- [19] Kolbe, T., Nagel, C. and Stadler, A., 2008. CityGML – A Framework for the Representation of 3D City Models from Geometry Acquisition to full Semantic Qualification. In: Proc. of ISPRS Congress 2008 in Beijing, China.
- [20] Lapiere, A. and Cote, P., 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; Taylor and Francis, London, pp. 381-393. Proceedings of the 26th UDMS, October 10-12, Stuttgart.
- [21] Nagel, C., Stadler, A. and Kolbe, T., 2009. Conceptual Requirements for the Automatic Reconstruction of Building Information Models from Uninterpreted 3D Models, Academic Track of Geoweb 2009 Conference, Vancouver.
- [22] Nagel, C., 2007. Conversion of IFC to CityGML; Meeting of the OGC 3DIM Working Group at OGC TC/PC Meeting, Paris (Frankreich), July 2007.
- [23] OGC, 2011. <http://www.openeospatial.org/> [last accessed Feb- 2012]
- [24] Pu, S. and Zlatanova, S., 2006. Integration of GIS and CAD at DBMS level, In: Fendel & Rumor (Eds); Proceedings of UDMS'06 Aalborg, Denmark May 15- 17, TU Delft, pp. 961-971.
- [25] Safe Software, 2008. FME Desktop Translator/Converter Software <http://www.safe.com/products/desktop/formats.php> [last accessed Feb-2012].
- [26] Van Oosterom, P.J.M., Lemmen, C.H.J., Ingvarsson, T., van der Molen, P., Ploeger, H.D., Quak, W., Stoter, J.E. and Zevenbergen, J. 2006. The

- core cadastral domain model. In: *Computers, Environment and Urban Systems: an International Journal*, 30 (2006)5, pp. 627-660.
- [27] Van Berlo, L., 2009. CityGML extension for Building Information Modelling (BIM) and IFC. *Free and Open Source Software for Geospatial (FOSS4G)*, Sydney.
- [28] Van Berlo, L. and De Laat, R., 2011. Integration of BIM and GIS: The Development of the CityGML GeoBIM Extension. *3D GeoInfo Conference, Berlin, Germany, Lecture Notes in Geoinformation and Cartography Series (Springer-Verlag, Heidelberg)*.