ISPRS Int. J. Geo-Inf. **2012**, *1*, 120-145; doi:10.3390/ijgi1020120

OPEN ACCESS
ISPRS International
Journal of
Geo-Information
ISSN 2220-9964
www.mdpi.com/journal/ijgi/

Article

A Unified Building Model for 3D Urban GIS

Mohamed El-Mekawy [1], Anders Östman [2,*] and Ihab Hijazi [3]

[1] Department of Computer and Systems Sciences (DSV), Stockholm University,
SE-106 91 Stockholm, Sweden; E-Mail: moel@dsv.su.se

[2] Faculty of Engineering and Sustainable Development, University of Gävle, SE-801 76 Gävle, Sweden

[3] Department of Urban and Regional Planning Engineering, Faculty of Engineering,
An-Najah National University, Nablus, Palestine; E-Mail: eehab@najah.edu

* Author to whom correspondence should be addressed; E-Mail: Anders.Ostman@swedesurvey.se;
Tel.: +46-26-633-337; Fax: +46-26-651-819.

*Received: 15 May 2012; in revised form: 15 June 2012 / Accepted: 2 July 2012 /
Published: 17 July 2012*

Abstract: Several tasks in urban and architectural design are today undertaken in a geospatial context. Building Information Models (BIM) and geospatial technologies offer 3D data models that provide information about buildings and the surrounding environment. The Industry Foundation Classes (IFC) and CityGML are today the two most prominent semantic models for representation of BIM and geospatial models respectively. CityGML has emerged as a standard for modeling city models while IFC has been developed as a reference model for building objects and sites. Current CAD and geospatial software provide tools that allow the conversion of information from one format to the other. These tools are however fairly limited in their capabilities, often resulting in data and information losses in the transformations. This paper describes a new approach for data integration based on a unified building model (UBM) which encapsulates both the CityGML and IFC models, thus avoiding translations between the models and loss of information. To build the UBM, all classes and related concepts were initially collected from both models, overlapping concepts were merged, new objects were created to ensure the capturing of both indoor and outdoor objects, and finally, spatial relationships between the objects were redefined. Unified Modeling Language (UML) notations were used for representing its objects and relationships between them. There are two use-case scenarios, both set in a hospital: “evacuation” and “allocating spaces for patient wards” were developed to validate and test the proposed UBM data model. Based on these two scenarios, four validation queries were defined in order to validate the appropriateness of the proposed unified building model. It has been validated, through the case scenarios and four queries, that the UBM being developed is able to integrate CityGML data as well as IFC data in an apparently seamless way. Constraints and enrichment functions are used for populating empty database tables and fields. The motivation scenarios also show the needs and benefits of having an integrated approach to the modeling of indoor and outdoor spatial features.

Keywords: IFC; CityGML; UBM; unified building model; 3D city modeling

1. Introduction

A 3D city model is generally defined as the digital representation of the Earth's surface and the built environment within a city. Using these models, a variety of applications may be created, covering the whole city or a focused specific building model. As the models become more detailed, the relationships between the spatial objects have to be modeled [1]. The 3D city models are of two types-design and real world models. While the design models are used by the architecture, engineering and construction (AEC) industry, the real world models are usually used by the geospatial information industry (mapping industry) [2]. As a result, design models are usually based on the maximum available levels of detail of the geometric representation, while the real world models are based on the requirements from the mapping task (cost, coverage, *etc*.).

The current focus of today's 3D city models is on representing geometric objects and element types. This supports the use of 3D city models mainly for visualization purposes. Many GIS applications, however, use thematic queries, analysis operations, simulation modeling and spatial data mining, which requires the representation of entities and non-spatial characteristics among them [3]. Although some practitioners argue that visualization fulfills most of today's application needs, important application areas would benefit from richer data. Examples of these applications are urban planning, building construction analysis, homeland security, real estate management, and emergency preparedness *etc.* [4,5,6].

From a technical perspective, reuse of information is required for building applications based on 3D city models. As the number of spatial applications increases, the need for common communication standards is becoming apparent [3]. In the building industry area, for instance, several years of research and development have resulted in the development of building information modeling (BIM). BIM is well recognized today as an important research area for dealing with problems related to information integration and interoperability [7]. Following these efforts, the Industry Foundation Classes (IFC) has been developed as a reference model standard for the building industry [8]. The IFC standard does not only 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 [9]. In the geospatial information field, 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 [10]. 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 [11].

This paper focuses on the integration of IFC and CityGML models. The motivation for this work is the development plans of the hospital in Norrtälje, Sweden, where the city is looking forward to extending the services offered by the hospital. The city of Norrtälje is located in the northern part of the county of Stockholm and the Norrtälje hospital is the biggest hospital in the municipality, serving over 120,000 inhabitants. All the hospital buildings in the municipality are owned by Locum AB. In accord with the service level agreement between the municipality and Locum AB, a request for needed extensions has been sent to the Locum AB. Various designers and architectural organizations propose designs in an open competition.

It is worth mentioning that in 2009 Locum AB decided to adopt the IFC standard in all of their upcoming new public buildings in order to have more and better quality indoor information about their hospital buildings. The Norrtälje City Hospital was the first hospital to be tested and modeled on IFC. Therefore, Locum AB has provided all the competitors with the current IFC model of the hospital buildings and required the new extensions to also be in the same format. These different designs need to be tested against a number of criteria imposed by the authorities for hospital buildings. Some of these criteria clearly require information from the building design as well as information from the surrounding environment and buildings. Therefore, the committee that is responsible for the evaluation of the design has requested information about the surrounding environment from the city of Norrtälje. The city has a complete geo-dataset of the City of Norrtälje, based on data captured by photogrammetric and laser scanning technologies. This city data is currently stored in CityGML format Level 2 for the entire city. However, more detailed levels (Levels 3 and 4) are available for specific focused areas. After gathering the data about the surrounding environment, the evaluation team needs to use the IFC design and the CityGML information for the surrounding built environment to evaluate the best design against a set of criteria as follows:

- *Evacuation plan*: The criterion states that all persons including the handicapped must be evacuated within a specified period of time—two minutes in the case of an emergency. Concerning a single building, the design team therefore has to test all space locations *versus* the closest possible exits. A matrix of evacuation time between all spaces and possible exits in the building needs to be created. The process starts by defining all spaces inside the IFC building (including all floors) and connecting them to the possible exits. Possible exits include all suitable doors, windows that can be opened as suitable exits, and also exits on the roofs of adjacent buildings that can be used to evacuate people. In addition, possible exits should be restricted to suitable surrounding spaces (like open areas, gardens, safe, flat adjacent roofs, far from main traffic roads, *etc*.). The information required for the above-mentioned criteria requires information from both the interior design of the building (for allocating spaces and possible exits) and the external environment (including information about surrounding buildings and their use, adjacent roofs, traffic, possible outdoor spaces and green areas).
- *Patient ward placement*: To fulfill the future plan, the hospital needs to add patient wards for 250 beds. The criterion strongly recommends that the distance between wards and heavy traffic

> or noisy places is at least 300 m, *i.e.*, a patient ward which is close to noisy places is not recommended. The design needs to be tested against the surrounding environment for noise pollution. This process starts by creating 300 m buffers around all the noise spots in the surrounding environment (e.g., heavy traffic) as well as noisy activities (e.g., workshops, schools, restaurants, *etc*.). The newly designed patient wards' positions and spaces are then tested *versus* these buffers to decide the suitability of their design. In some cases, the 300 m distance is difficult to fulfill for all patient rooms. Therefore, other back-up alternatives are considered. These alternatives include constructing fountains or planting trees for reducing external noise. This process also requires indoor design data (for the patient wards) and the outdoor city geodatabase model (for surrounding activities, land use and buildings).

In the cases described above, information about the interior of the building as well as the exterior environment is required. Moving the data from one world to the other is, however, not an option as it will create loss of data. Two examples: Information that may be lost when converting from CityGML, includes information about surrounding streets, since it cannot be represented in IFC; and secondly, information about building elements such as walls, and details of doors and windows coming from IFC towards CityGML, may be lost in the CityGML model. Therefore, a new approach for integration of both IFC and CityGML data and for providing the information to the above cases is required.

The objective of this paper is to describe a unified building model (UBM) which encapsulates both the CityGML and IFC models, thus avoiding translations between the models and loss of information. Four validation queries have been defined, allowing us to validate the appropriateness of the proposed unified building model, namely:

Q.1. Define the possible windows on a specific floor that can be made into doors in emergency situations or have access to the roof of an adjacent building.

Q.2. Find the doors that have access to the outer environment of a building and define the use of the accessed outer space.

Q.3. Find the surrounding buildings that have industrial use and are within a 300 meter distance from the development site of the hospital.

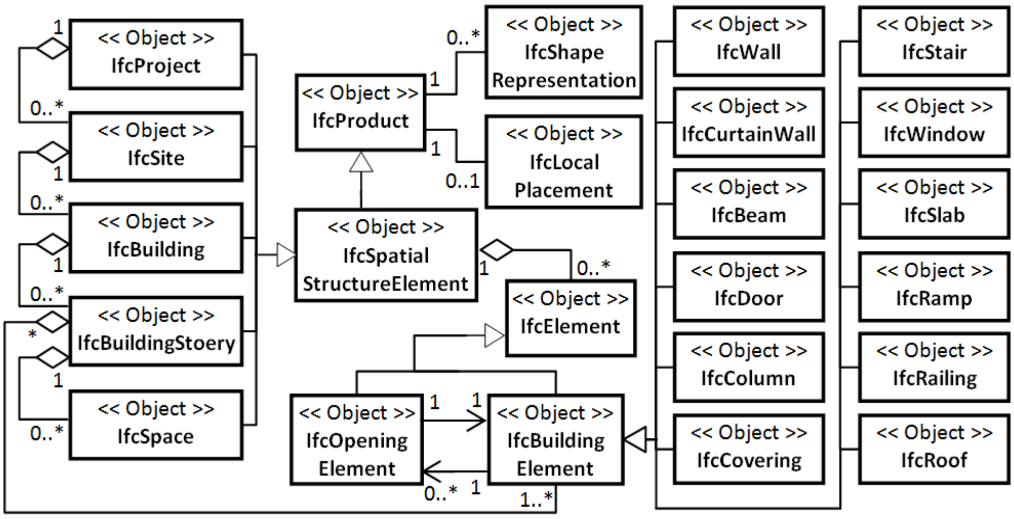
Q.4. Select outdoor areas that have trees or are able to be planted and furnished with a water fountain or noise inhibitors.

2. Research Background

2.1. IFC Building Model

IFC is an object-oriented format developed by the International Alliance for Interoperability (IAI) [12]. It is used for facilitating 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 [13]. However, there is no universally accepted building model for IFC [14]. In this section we present an IFC building model that is primarily based on the work done by the IAI and ISO in the form of IFC standard documentation [12], the ISO 16739 standard [15] and Benner *et al*. [16]. From these efforts, important concepts (Figure 1) are identified. UML standard notations are used for developing the IFC building model.

Figure 1. IFC building model.



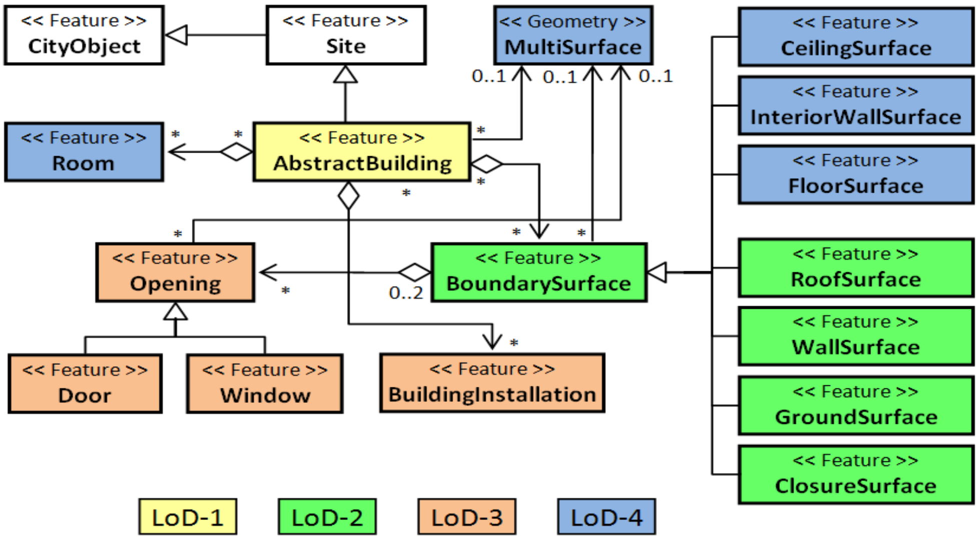
Based on the IFC model here presented, a building should have at least one storey and may have multiple storeys. Each storey may have zero or more spaces related to it, *i.e.*, a building structure which has only one wall is a building with zero spaces. Building elements and opening elements are subtypes of structural elements. 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 a door, a window) is attached to only one building element. Figure 1 shows twelve types of building elements that can represent a building structure in the IFC standard. The scope of this study is limited to building models that only represent constructed parts of buildings.

2.2. CityGML Building Model

CityGML is an open standard that has been implemented as an application schema for GML3 [17,18]. GML3 is the extendible international standard for spatial data exchange that has been developed and issued by the Open Geospatial Consortium (OGC) [19] and the ISO TC211 [20]. 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 [17]. Based on ISO 19107 [21] and ISO 19109 [22], a CityGML Building Model has been produced in the CityGML standard [23]. The building model (Figure 2) 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 the CityGML building model.

CityGML is developed in five levels of detail (LoDs) which are used to represent city model objects according to the required level of detail in different applications. The 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 a CityGML building model from LoD1 to LoD4. Figure 2 represents the LoDs by different colors.

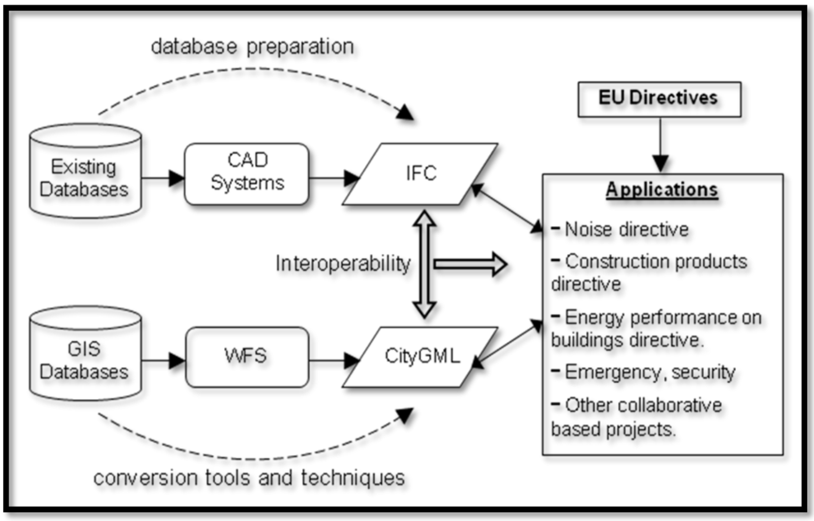
Figure 2. CityGML building model.


<< Feature >> CityObject
<< Feature >> Site
<< Geometry >> MultiSurface
<< Feature >> CeilingSurface
<< Feature >> InteriorWallSurface
<< Feature >> FloorSurface
<< Feature >> Room
<< Feature >> AbstractBuilding
<< Feature >> Opening
<< Feature >> BoundarySurface
<< Feature >> RoofSurface
<< Feature >> WallSurface
<< Feature >> GroundSurface
<< Feature >> ClosureSurface
<< Feature >> Door
<< Feature >> Window
<< Feature >> BuildingInstallation
0..1
0..1
0..1
0..2
*
LoD-1
LoD-2
LoD-3
LoD-4

2.3. The Need for Integration

Unified applications are seen as necessary for increased globalization and common problems. At the EU level, different initiatives, such as the Infrastructure for Spatial Information in the European Community (INSPIRE) directive [24], have suggested building common spatial applications for EU countries. Related applications and directives (e.g., noise directive, construction products directive, energy performance, emergency, security and others) are seen here as a motivation for the integration of IFC and CityGML (Figure 3).

Figure 3. Proposed conceptual model for CityGML and IFC integration.



The IFC standard is seen as an entry point for different applications as shown in Figure 3, which considers indoor areas in the building industry. The indoor objects are to be translated to the IFC standard through the preparation of enriched databases. A large number of drawings, documents and CAD files are already available, however, on paper or stored in proprietary file formats, and converting this information into IFC might be a very time-consuming process.

On the CityGML side (Figure 3), there are a number of databases available that store information about outdoor spatial objects. These are usually modeled on GIS standards and presented in simple and separated applications covering small areas. At the initiation of the Open Geospatial Consortium (OGC) [17], the building of an advanced collection of web services is in progress. These web services will cover the whole of Europe according to the INSPIRE directive [24], and have different levels of detail for different applications and analyses. This means that many European spatial datasets will probably be transferred by means of GML and CityGML formats. These datasets can be directly accessed from Web Feature Services (WFS) or converted from other GIS formats/databases. To what extent CityGML will be used in future INSPIRE data specifications remains to be seen.

2.4. Related Work

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 [25]. IFC is then used to assemble computer readable models that contain information related to different parts of a building [8,26]. CityGML is used, however, to define information related to topological and semantic properties of a geographical area, including buildings [17,18]. On the one hand, IFC has been developed as an ISO standard and it has been largely accepted for the building industry [27]; on the other, CityGML has been recently adopted as an international standard for modeling cities in the Open Geospatial Consortium (OGC) and the EU [17,18]. This has led to major increases 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 necessary step for getting a more complete picture of 3D modeling at different levels of detail, *i.e.*, sharing and exchanging information between building industry objects (represented in IFC) and geospatial objects (represented in CityGML). Several efforts have been made to integrate IFC and CityGML. All these efforts are mainly in the form of developing frameworks, extending the 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 in creating a mapping specification from the XML version of IFG geometry to GML and *vice versa* [28]. Looking specifically at technical aspects, another framework was proposed by Nagel [29], aiming for algorithms that automatically transform IFC building models into CityGML models. Isikdag and Zlatanova [11] complemented Nagel’s research by proposing a guide framework for the automatic generation of buildings in CityGML using BIM and based on a definition of building semantics and components. Following the holistic view of 3D city modeling 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 [30]. 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. A few additional conversion environments can be seen in this area. Van Berlo [31] demonstrates the latest contribution of their team as the Application Domain Extensions (ADE) which integrates Building Information Model (BIM) data based on the open standard IFC into CityGML. Not only research efforts, but commercial software products for conversion from IFC to CityGML (e.g., IfcExplorer, 2008 [32] and Safe Software, 2008 [33]) also contribute to the development of 3D city modeling integration.

These efforts, however, 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 downgrading IFC to lower LoDs in CityGML or (d) a discussion on the interest in the rich semantics of IFC.

Several studies, such as [11,16,30], emphasize that a formal framework for strict semantic and geometry conversion is required for a complete integration of CityGML and IFC. Geometry is highlighted as one of the major problematic concerns for integrating IFC and CityGML [27,30]. However, most of the recent efforts have focused on conceptual integration or conversion processes. Considering the vast amount of attempts at defining geometric differences and developing conversion algorithms [13], we do not address this type of problem in our research. In the present study, we focus on the conceptual integration and mapping of different objects in both IFC and CityGML standards.

3. The Research Approach

From the motivation scenario, the need for combining indoor IFC information on buildings and outdoor CityGML built-environment information is apparent. We nevertheless contend that conversion from IFC to CityGML can lead to the following limitations:

- Losing data in conversion from IFC to CityGML, because CityGML cannot sustain all the IFC concepts. Similarly, IFC cannot sustain the concepts of CityGML. This has also been affirmed by the findings of [27].
- 3D city models are currently scattered over public and private sectors in different systems, different conceptual models, different data formats, different data schemas, different levels of detail and different quality [34]. In addition, the potential of 3D models goes beyond visualization of 3D objects of virtual scenes to real 3D city models. In such environments, conversions greatly limit the number of users who are able to fully exploit the potential of IFC or CityGML separately.
- It is not possible to integrate IFC semantics into CityGML by default [31]. An alternative is the use of an extension mechanism for CityGML and IFC; however, only few extension efforts have been reported (refer to Section 2.4).
- The CityGML and IFC show an overlapping scale of data represented, in which CityGML has a structure that has the capability to manage data to all scales (e.g., from city-scale to 1:100 or more detailed) and IFC focuses on the scale of rooms, walls and even structural details. However, examples of attempts at extending either the CityGML or IFC models to cover the detailed information covered by the other have been shown to be complex processes, generating extremely large file sizes as well as data inconsistency. This can also be complemented by the result of Van Berlo and De Laat [35], which additionally emphasizes the need for new extensions and different considerations for each intended application according to the included scale of represented data.

- Continuously moving the data from one format to another can be inefficient in that it increases the conversion overhead.

Following these limitations, this study proposes a unified model-oriented approach for developing a Unified Building Model (UBM) that can contribute to increasing the integration of CityGML and IFC for extending 3D city model applications. We believe that the UBM will allow both IFC and CityGML worlds to interact with it without loss of data and without frequent conversion overheads between IFC and CityGML models. The essence of the approach is developing a unified model that can capture information about both IFC and CityGML. Due to its detailed information model, it enables access to IFC and CityGML as well as integrated access to building information. These benefits motivate the need for the UBM to facilitate the conversion from IFC to CityGML or from CityGML to IFC. This also means that the original Revit/CAD standards are not considered. The IFC is becoming a unique standard that is supported by read/write capabilities of different tools regardless of their origin.

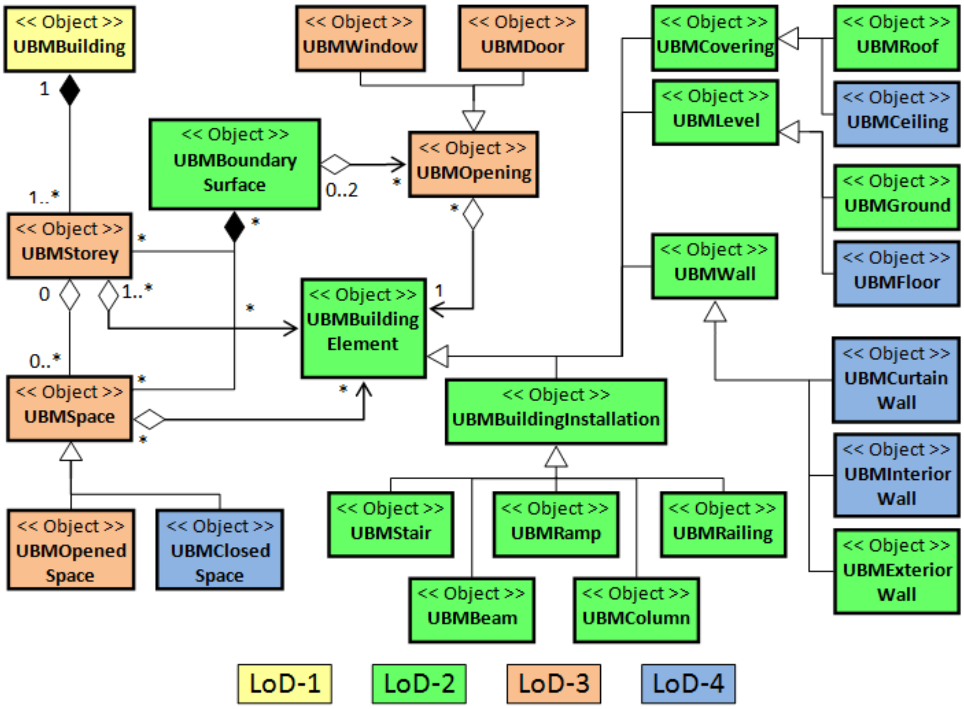
A unified model in the current study is defined as a superset model concept that contains the features and objects from both the IFC and CityGML building models. The unified model is originally derived from the superset concepts of mathematics in the 1970s led by Thomas J. Jech [36] and other researchers. It has also been used in software engineering from the mid 1990s [37].

To build the UBM, all classes with their concepts were initially collected from both models, but their relationships were omitted. Overlapping concepts were merged and new objects were created in such a way that both indoor and outdoor objects were captured. Finally, relationships between the objects were rebuilt to produce our UBM. Similar to IFC and CityGML, UML notations are used to represent the UML objects and the relationships between them. The model is also developed in different LODs that are represented in different colors in Figure 4. The details in LODs are considered to match those of CityGML. The following is a brief explanation of LOD1-LOD4 without LOD0, since buildings only start to appear at LOD1:

- ***LOD1*** is the first representation of single buildings. They appear only as blocks with flat roofs for all types of buildings as a basic level.
- ***LOD2*** represents single buildings with their roofs and exterior surfaces differentiated. External details attached to building facades also appear at LOD2.
- ***LOD3*** represents buildings in an architectural model with details of roofs, walls and exterior as well as interior structural elements. Openings also start appearing at this LOD in different building elements as doors or windows**.**
- ***LOD4*** gives a complete model of a building by adding interior structures of building elements. All desired details can be presented in LOD4, such as exterior installations, interior installations and even movable furniture elements. In this study, the focus is only on building structural elements. Therefore, furniture objects are not considered in the proposed model or reference ontology.

Owing to its wide acceptance, unified modeling language (UML) is selected as the modeling language for the proposed unified building model (UBM) approach. The requirements of the proposed UBM entailed that conversion was done with (a) minimum information loss, and (b) an efficient schema matching and mapping process. Therefore, the UBM aims to offer free representation for the data model building elements, merging all information from both CityGML and IFC as well populating the missing ones. Thus they can be all brought into a single unified model that can be used for performing all the needed spatial analyses and queries.

Figure 4. The unified building model.



4. The Unified Building Model

As mentioned in the research approach, to build the UBM, all classes with their concepts were initially collected from both models while omitting their relationships. The omission is motivated by a significant reason regarding apparent differences between the structure and relationships of both IFC and CityGML building models. 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 the structure of buildings. This can, for example, start with a wall or slab which can then be linked to a space, then a storey and end up with a building. There is, however, no unique way to connect one 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. 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. 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.

The relationships are rebuilt in the ArcGIS between the UBM classes and tables according to the spatial as well as tabular relationships between them. These relationships can be built differently, according to the available tools. Overlapping concepts were then 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 colors are used to represent different LODs. The UBM elements are 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 certain height level. 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 an 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 Figure 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 some of the IFC classes (e.g., *IfcWall*, *IfcDoor* and *IfcWindow*) as (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.

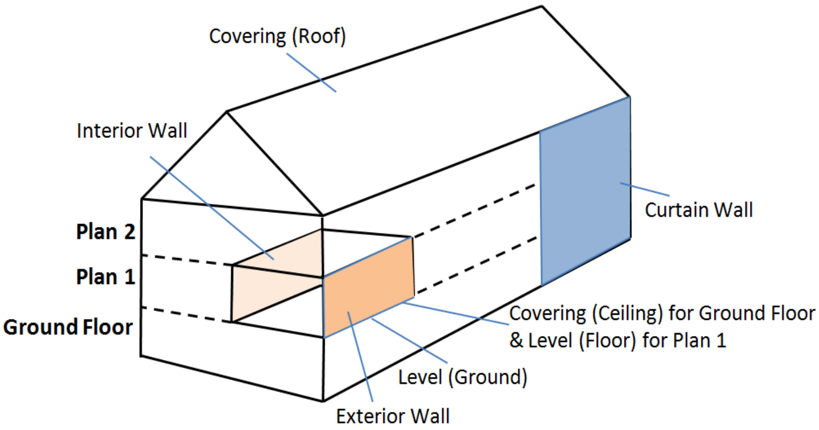
For building elements, we have defined some new concepts in such a way that all concepts from both IFC and CityGML can be covered. The main difference between building elements in CityGML and IFC is the representation of different surfaces, interior and exterior parts of a building (wall, roof and ground). Because of the need for different LoD representations and definitions of elements, such as (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 the 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 the 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: (i) *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; (ii) *Interior* for an internal wall between rooms or spaces (none of its faces has connection with the outer environment); and (iii) *Exterior* for an external wall that has connection with the outer environment and represents a part of external facades of a building.

These concepts are shown in Figure 5.

Figure 5. Covering, level and wall concepts.



Building installations (ramps, chimneys, balconies, beams, columns, *etc*.) are defined differently in both IFC and CityGML. In IFC, they are defined as normal building elements (like walls, slabs, *etc*.) with the same geometric concepts. However, in CityGML building installations are specified in a separated object named ‘building installation’. In the latter case, the geometries of these installations (ramps, chimneys, balconies, beams, columns, *etc*.) are defined by multi surfaces that construct the objects and stored in *CityGMLMultiSurface*. In the UBM, we have defined these installations as subclasses of the *UBMBuildingInstallation* class in order to represent the external building installations that are important for the external shape of a building at LoD2. This also allows the turning-off of building installations if a smaller model is desired for faster processing or less detailed analysis.

When merging classes, constraints among attributes may be defined. Attributes of the class *UBMBuilding* is, for instance, based on the merging of *IFCBuilding* and *CityGMLAbstractBuilding*. The following constraints may then be defined:

$$UBMBuilding.Envelope = IfcBuilding.Envelope = CityGMLAbstractBuilding.Envelope$$
$$UBMBuilding.Id = [IfcBuilding.Id, CityGMLAbstractBuilding.Id]$$

5. The UBM Implementation

Existing tools for data conversion like IfcExplorer and Safe Software vary in their converting capabilities, complexity of use, and visualization of results [11]. It is important here to emphasize that different tools can be used for the implementation of our proposed UBM with various levels of complexity for the implementation process. In order to keep the focus on the conceptual UBM and to represent its capabilities, we have chosen ArcGIS as an implementation environment and a tool primarily due to flexible implementation of data interoperability that it offers, its maturity in GIS applications, its inclusion of a big number of 3D functionalities and its familiarity among our research members. In addition, the following benefits of ArcGIS over other tools have led to its being chosen for implementation:

- It facilitates the navigation of different conversion processes and their visualization before the final product.
- It provides an environment where our data model can be built between both IFC and CityGML domains. This can facilitate automated processes between both domains.
- Within an intermediate environment, it is possible to monitor our filled-in tables from each IFC and CityGML and show how the proposed data model can be enriched.
- It enables the utilization of the ArcGIS geo-processing environment, including the ModelBuilder framework, which *enables* modularity and customization processes.

The implementation and demonstration of the applicability of the UBM is presented in the following three subsections. The first two subsections (5.1–5.2) explain how the UBM can be implemented in ArcGIS using Geodatabase technology. The third subsection (5.3), however, demonstrates the usefulness of the UBM over IFC and CityGML building models separately.

5.1. Mapping UBM to ArcGIS Geodatabase

In order to implement the UBM model in ArcGIS, *file geodatabase* is firstly created in the *ArcCatalog*. Our opting for *file geodatabase* over *personal geodatabase* is for the following reasons: (i) IFC and CityGML file can be large in size, depending on the level of details they represent. *File geodatabase* is then chosen as it does not have a storage size limit, whereas *personal geodatabase* is limited to only 2 GB, (ii) Performance is improved in *file geodatabase* compared to *personal geodatabase,* especially after the database reaches about 500 MB, and (iii) *File geodatabase* provides capabilities for customizing storage through compression of vector data and configuration keywords that also support processing of big files.

After the construction of *file geodatabase,* feature classes are created that correspond to the main classes in the UBM. On these feature classes, different subtypes are also created in order to define the subset of features in a corresponding feature class that shares the same attributes. The subtypes are used to categorize the data in our data model. Choosing to create subtypes instead of separated feature classes or relationships to the supertype feature class is motivated by the following reasons:

(i) Increasing the geodatabase performance by representing different objects as a subset of feature in the main feature class. This shows the advantage of creating *UBMRoof* as a subtype feature of the *UBMCovering* supertype where *UBMRoof* represents the final roof of a building which is represented once for each building. There is, however, no need for creating a separated feature class for it.

(ii) Capability of setting default values for pre-defined fields which automatically apply when creating new features. The *UBMCeiling* subtype, for example, is created and defined so that whenever this type of covering is added to the feature class, its thickness is automatically set to 15 or 20 centimeters.

(iii) Possibility of applying coded or ranged domains to features. This further allows constraining input information to only a valid set of values that are pre-defined. Building materials, for example, can be restricted by a coded domain on the *UBMCeiling* and *UBMFloor*, different from the *UBMLevel* and *UBMCovering* for facing the external weather conditions.

(iv) Enabling connectivity rules with other subtypes and feature classes. This represents how the UBMCurtainWall should only be connected to external walls (*UBMExteriorWall*) but not to an internal wall (*UBMInteriorWall*).

(v) Enabling the creation of topology rules with other subtypes and feature classes. In the UBM, topology rules represent how to classify the *UBMSpace* class into the *UBMOpenedSpace* and *UBMClosedSpace* subtype classes. The *UBMClosedSpace* is defined by topology rules to ensure that all the surrounding walls, ceiling and floor are connected to have a closed geometry; otherwise it is classified as *UBMOpenedSpace*.

(vi) Enabling the development of relationship rules with other subtypes and feature classes. In the UBM, different relationship rules can be defined, explaining that wooden walls cannot be exterior walls or spiral stairs cannot be used as external emergency stairs, *etc*.

Table 1. Feature classes with corresponding subtypes.

• **UBMBuilding**
• **UBMStorey**
• **UBMSpace** (with the following two subtypes) o UBMOpenedSpace o UBMClosedSpace
• **UBMBoundary Surface**
• **UBMBuilding Element** (with the following four subtypes) o UBMCovering (with the following two subtypes) ▪ UBMRoof ▪ UBMCeiling o UBMLevel (with the following two subtypes) ▪ UBMGround ▪ UBMFloor o UBMWall (with the following three subtypes) ▪ UBMCurtainWall ▪ UBMInteriorWall ▪ UBMExteriorWall o UBMBuilding Installation (with the following five subtypes) ▪ UBMStair ▪ UBMRamp ▪ UBMRailing ▪ UBMBeam ▪ UBMColumn
• **UBMOpening** (with the following two subtypes) o UBMWindow o UBMDoor

The feature classes and their subtype classes that represent the UBM are shown in Table 1. It is important to note that we have listed all the feature classes and subtypes that only represent spatial objects in the UBM. Other non-spatial information (like tables only representing attributable data about spatial objects) and metadata tables are also created in the file geodatabase. For instance: organization table (for information about the construction organization), associated material table (for information about building materials and their description), and owner history (for information about how the building was owned and used), *etc*.

For all the feature classes (representing spatial objects) in the UBM, multipatch geometry type is used. This is because this type of geometry enables us to represent spatial objects in 3D geometry as multiple surfaces. Additionally, it gives the possibility to perform all the GIS operations and analysis tools in 2D as well as 3D.

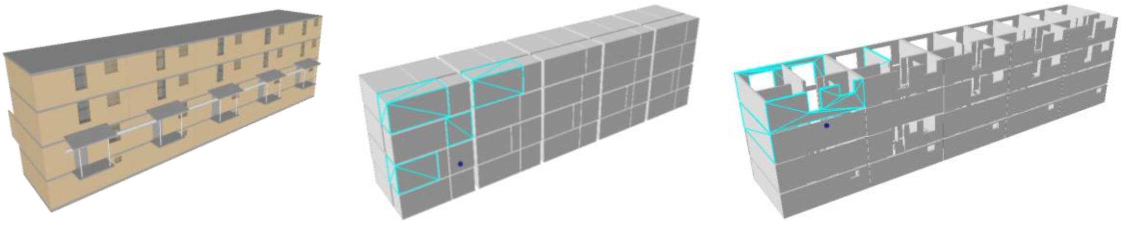
5.2. UBM Internal Enrichment Functions

As discussed in Sections 3 and 4, the UBM contains all the features and spatial objects from both IFC and CityGML building models. However, the building information is typically available in either IFC or CityGML standard, giving an incomplete UBM as a result. For instance, in the Norrtälje Hospital scenario, Locum AB captures indoor information (in IFC standard), but it also needs outdoor information to evaluate extended building design. In this situation, the information coming from only one side, IFC or CityGML, is not sufficient to represent the entire UBM data model. In the hospital building example, only the segments of the UBM data model that are related to IFC can be populated. Additionally, there are conceptual differences in representing building parts and elements between IFC and CityGML according to the geometrical representation. Our solution to populate empty tables of the UBM data model is based on defining internal enrichment functions and constraints into our UBM implementation. The aim of these enrichment functions is to use existing information for populating empty tables or fields. Constructing the UBM tables is, in part, performed automatically. This means that based on the implementation on the ArcGIS, the tabular data (descriptive tables) and geometric tables are automatically created either from IFC or CityGML. However, other tables need different level of manual processing, based on the required spatial analyses. It is worthwhile emphasizing here that ArcGIS can automatically bring IFC and CityGML data with their geometry. In addition to that, customized details can be further manually added. Some of the enrichment functions are presented below:

(1) Extracting Surfaces from *IFCSpace* and *IFCWall*

Figure 6(a) shows an example of an IFC building. When importing this IFC building model into a newly created *File Geodatabase* in *ArcGIS*, a number of feature classes representing all the building elements in IFC are created.

Figure 6. (**a**) IFC building. (**b**) Spaces in IFCSpace feature class. (**c**) Walls in IFCWallStandardCase feature class.

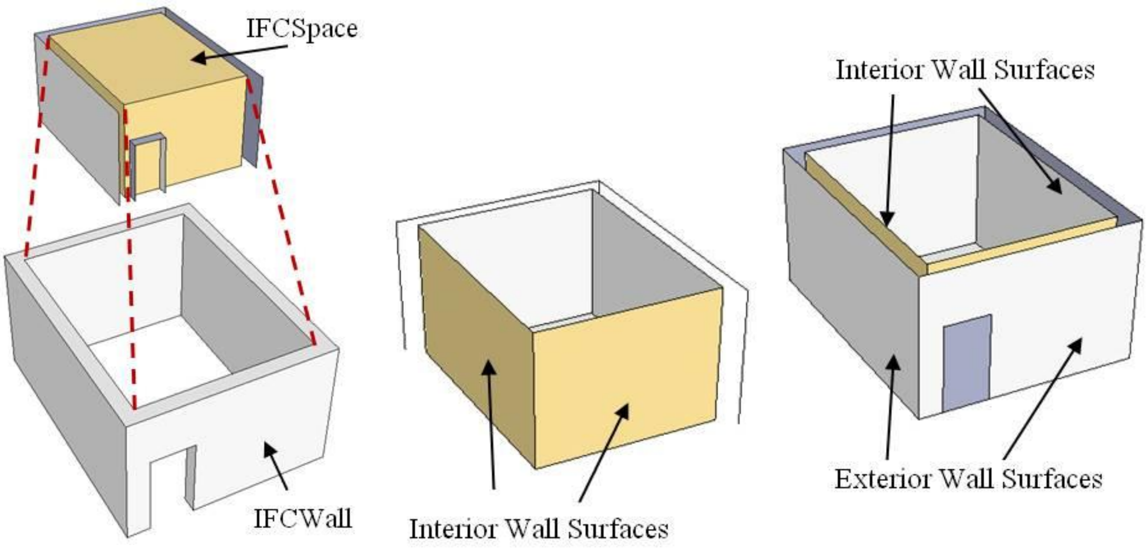

(**a**) (**b**) (**c**)

Two of these feature classes are *IFCSpace* (Figure 6(b)) and *IFCWallStandardCase* (Figure 6(c)). Representing building parts (e.g., *IFCSpace* and for *IFCWallStandardCase*) without representation of their surfaces does not fulfill the representation of these elements in CityGML. Therefore, there is a need for an internal function on the UBM to extract the different surfaces of walls (interior and exterior) in order to populate their representation in CityGML.

Figure 7(a) shows how a space, a room in this case, is represented by four surrounding walls. The ceiling and floor are intentionally omitted to show the inside space in the figure. The following two steps explain how this task is performed in ArcGIS functionality:

- Using the 3D analyst tools, a 3D intersection is performed between spaces in *IFCSpace* and walls in *IFCWallStandardCase*. The result of this process is only the interior boundaries of spaces (Figure 7(b)) where the volumes of the spaces intersect the walls that surround each space. Similarly, the same process is repeated for intersecting spaces with *IFCSlab* to extract the ceiling and floor surfaces.
- For extracting the exterior wall surfaces, a simple operation for selecting the remaining parts out of the intersection is performed. This results in the exterior boundaries as shown in Figure 7(c).

Figure 7. (**a**) Representing space in IFCSpace. (**b**) The intersection result between IFCSpace and IFCWallStandardCase. (**c**) Interior *versus* exterior wall surfaces of the space.

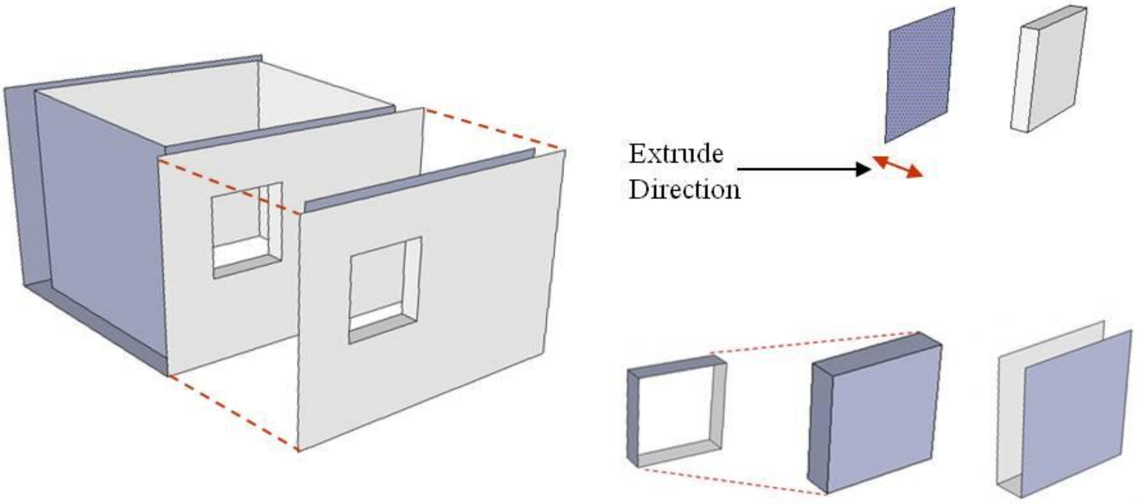


(**a**) (**b**) (**c**)

(2) Extracting Windows from CityGML Surfaces

The surfaces in CityGML are defined as abstract classes that give the structure of an exterior shell of a building and visible surfaces of interior spaces in the building. In these surfaces, opening elements (doors and windows) are represented as holes within the surfaces. A hole, representing an opening element, is represented by an interior and exterior ring within the corresponding surface geometry object. The two rings are typically the same, *i.e.*, the outer boundary may consist of the same points as the inner ring (denoting the hole) (Figure 8). Representing an opening by surfaces only does not fulfill the representation of windows and doors in IFC as building elements. Therefore an internal function on the UBM should be performed.

Figure 8. (**a**) A surface in CityGML with a window opening in its wall. (**b**) Extruding the window from the interior to the exterior surface of the wall.



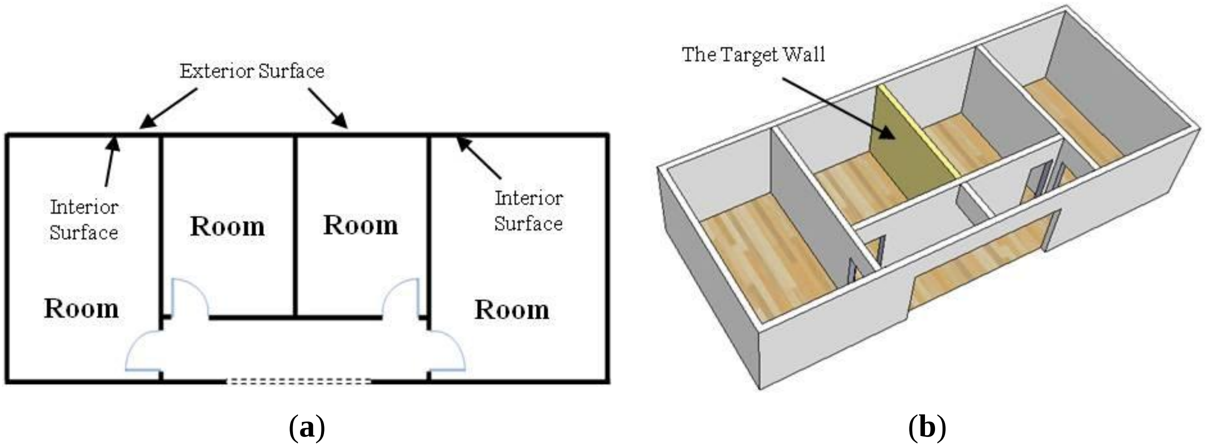
(**a**) (**b**)

Figure 8(a) shows a room with a window represented in one of the walls. An extrude function is performed from the interior surface towards the exterior surface (Figure 8(b)) in order to construct the window as a building element. The thickness of windows and doors is used to define the extrude measurement.

(3) Extracting Building Elements (Walls and Slabs) from CityGML Surfaces

In this internal function, we present how to populate walls, slabs and floors as building elements in the UBM from a given CityGML model. This of course applies only for levels of detail 3 (LoD3) and 4 (LoD4), as these building elements are not represented in lower level of details of CityGML. Figure 9(a) shows a part of an architecture plan that represents only four rooms. Coming from CityGML, each room has four surrounding walls, one ceiling and one floor to be considered as a room space. Figure 9(b) shows a perspective view of this architectural plan with walls and floors in which the ceiling is removed for visualization purposes. Each room in the CityGML has interior as well as exterior surfaces, as shown in Figure 9(a). The type of each surface, interior or exterior, is defined by an attribute in each element in CityGML. However, representing only surfaces does not fulfill the representation of walls, ceilings and floors in IFC as building elements. Therefore an internal function on the UBM should be performed.

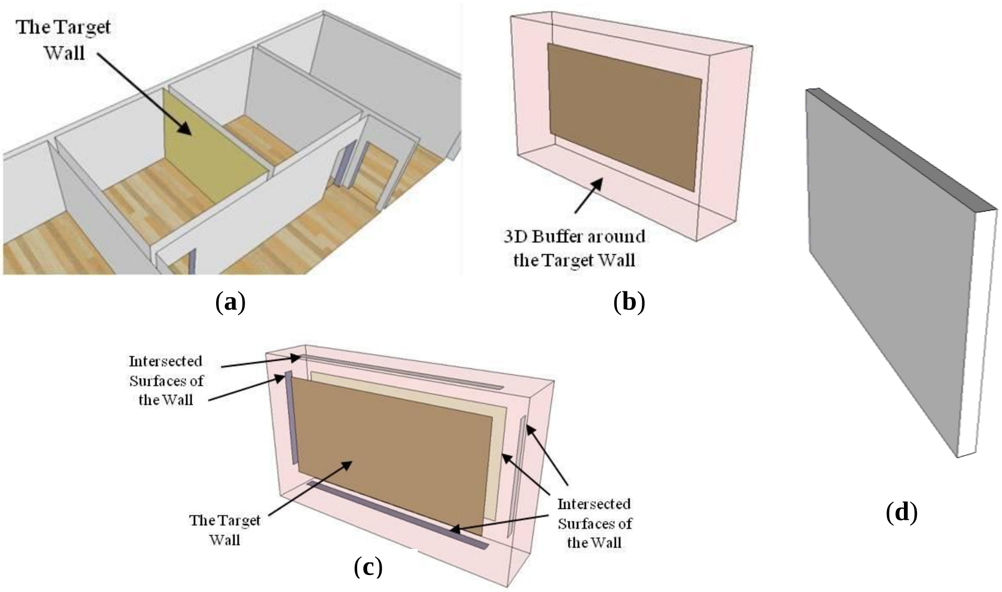
Figure 9. (**a**) Part of an architecture plan of a building. (**b**) Perspective view of the architecture design.



The following steps demonstrate how this internal function should be performed to extract building elements:

- Selecting a room and its surrounding surfaces. This step is done by a simple select query on the Table that stores information about rooms and their boundary surfaces.
- Starting with the interior surfaces, for example, all interior surfaces of the space can be selected by the definition type of surface from the Table. As an example, Figure 10 shows one interior surface for visualization purposes only.
- Creating a 3D buffer around the selected surfaces. The buffer distance is defined by the user and must be bigger than the thickness of the thickest wall, ceiling or floor. Figure 10(b) shows an example of a 3D buffer created around the target wall in this example.
- Intersecting the 3D buffer with all surfaces. This step (Figure 10(c)) is important in order to check which surfaces are completely contained by the 3D buffer. This ensures that only corresponding surfaces of the same wall are selected as no other wall surfaces are completely placed in this buffer. In Figure 10(c), the six surfaces are moved from each other only for visualization purposes.
- Constructing the wall. This step is done to combine the selected surfaces with each other for constructing one wall out from the 6 different surfaces. Figure 10(d) shows an example of a target wall constructed as a building element.
- A final step should be a classification of the resulting wall. This classification is based on a condition on the target surface (wall in this example). If at least one of the contained surfaces is an exterior surface, the wall is stored as an exterior wall in the UBM. Otherwise, it is stored as an interior wall if all surfaces are interior. Similarly, for the case of a ceiling, an exterior surface leads to classification of the ceiling as *UBMCovering-UBMRoof* and for the case of a floor, an exterior surface leads to classification of the floor as *UBMLevel-UBMGround*.

Figure 10. (**a**) Perspective view of the target wall. (**b**) The target wall surrounded by a 3D Buffer. (**c**) The target wall composed of its 6 surfaces. (**d**) The target wall constructed as a building element.


The Target Wall
3D Buffer around the Target Wall
(a)
(b)
Intersected Surfaces of the Wall
The Target Wall
Intersected Surfaces of the Wall
(c)
(d)

5.3. Demonstrating the UBM for the Use Cases

To demonstrate the applicability of the UBM for the previously presented use case scenarios (Section 3), we present in this section how the selected queries in the scenario are implemented. These queries are posed during the execution operations undertaken for the presented use cases. The first two queries are discussed first:

- *Q.1. Define the possible windows in a specific floor that can be extended as doors in emergency situations or have an access to the roof of an adjacent building.*
- *Q.2. Find the doors that have access to the outer environment of a building and define the use of the accessed outer space.*

These two queries involve operations that are needed for the building evacuation plan criterion, *i.e.*, within 2 min. Figure 11 shows part of a perspective model representing an IFC building to the right of the figure, and a combination of a CityGML building on the left, and its CityGML surrounding environment. They are brought all together into the implemented UBM in ArcGIS.

For the space selected in Figure 12(a), the design team needs to check the nearest evacuation option. Therefore, firstly, the team uses the UBM to select all the possible exit doors or the windows that can be opened as doors in the building. These exits typically come from the IFC part where different attributes provide the dimension of the spaces. These dimensions are used for calculating distances and times for passing through spaces and corridors as presented in Figure 12(b). The result shown in Figure 12(b) only comes with the IFC model. However, access to the built-environment surrounding the building may result in different solutions, which may be illustrated by considering the space in Figure 13(a). The calculated distance from this space to each possible door in the building is more than 2 min, as there is vertical connection only via stairs, which are required because elevators are prohibited in the case of an emergency. However, by adding the information from the adjacent CityGML building, the access to the safe flat roof of the adjacent building is considered as having a less-than-2-minute escape time. Based on the combined indoor and outdoor data stored in the UBM, it then becomes possible to define the doors and windows that can be utilized as exits in emergency case (Figure 13(b)).

Figure 11. UBM representation of IFC and CityGML data.

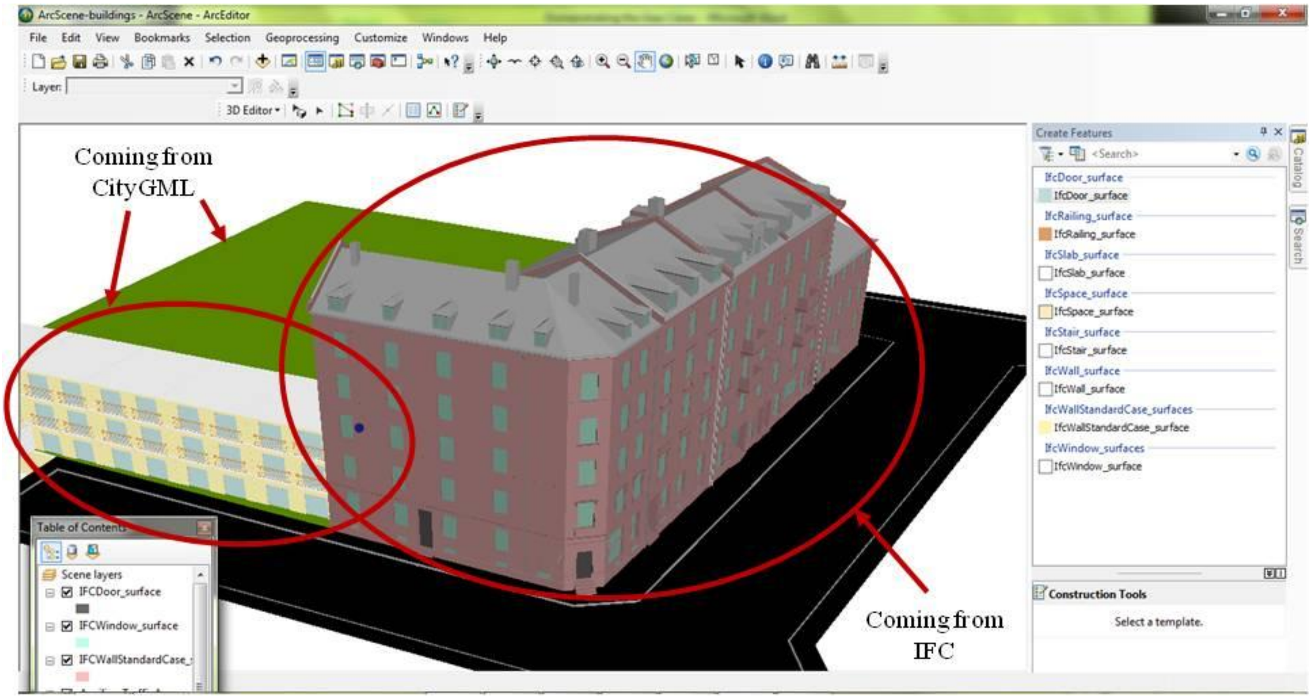
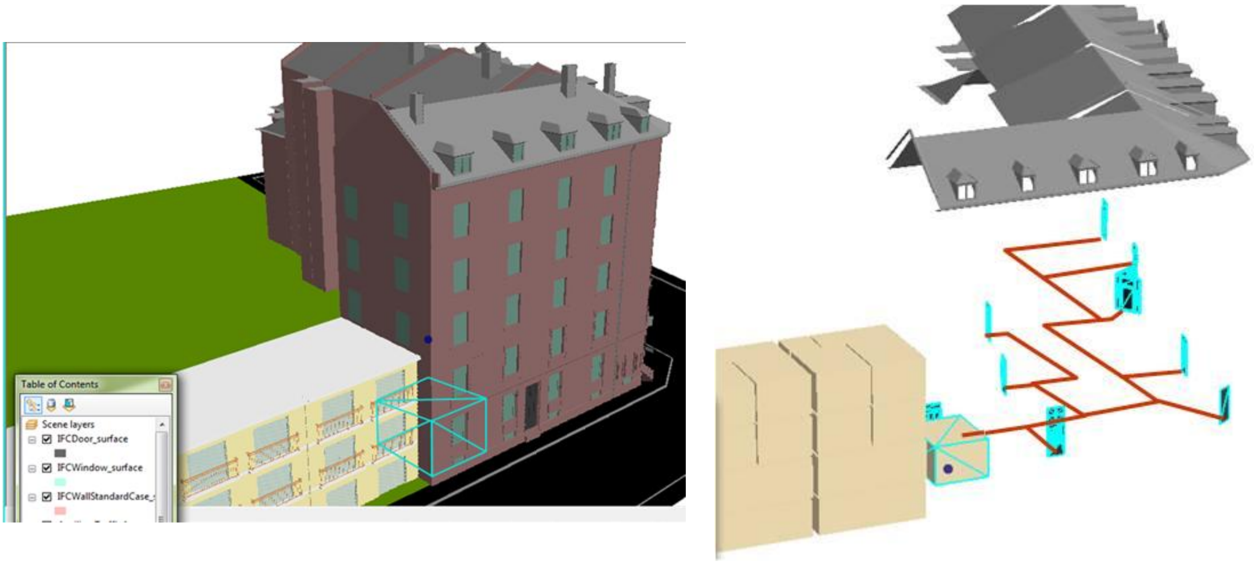


Figure 12. (**a**) A space to be tested *versus* the evacuation plan. (**b**) Connections from the space to all exits to be used for calculating distances and times.



(**a**) (**b**)

Figure 13. (**a**) A space to be tested *versus* the evacuation plan. (**b**) Considering the roof of adjacent CityGML building for the evacuation plan.

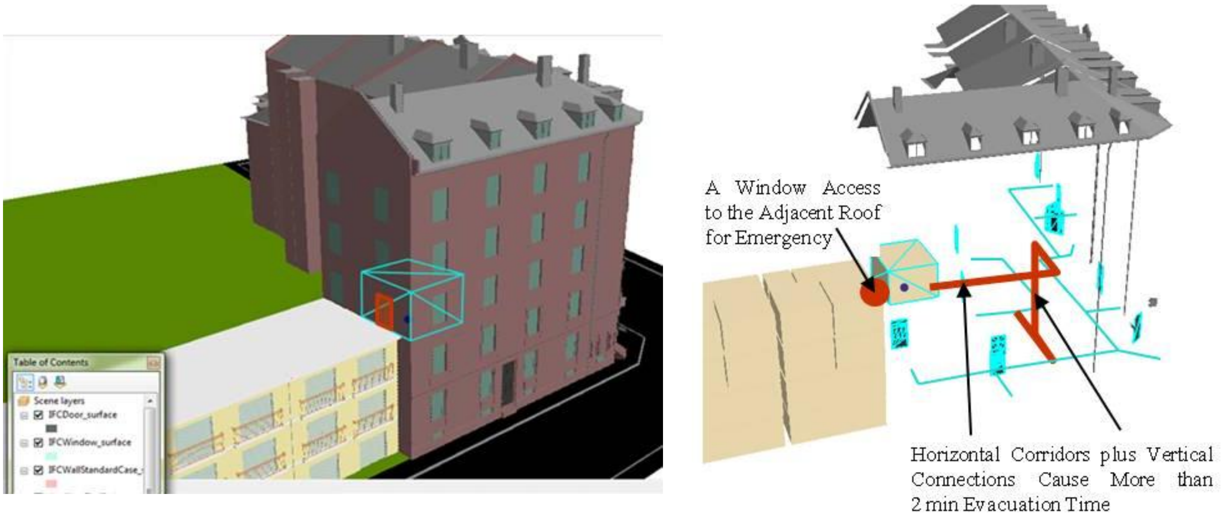
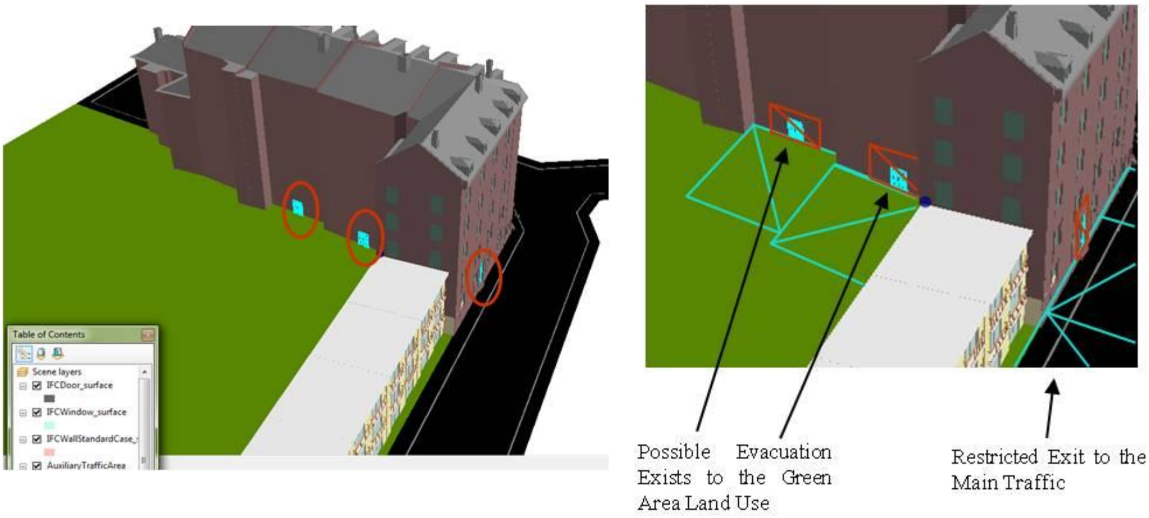


Table of Contents
Scene layers
IFCDoor_surface
IFCWindow_surface
IFCWallStandardCase_
A Window Access to the Adjacent Roof for Emergency
Horizontal Corridors plus Vertical Connections Cause More than 2 min Evacuation Time

(**a**) (**b**)

In the second query (Q.2.), the evacuation is to be restricted to outer spaces such as open areas or gardens and not to a main traffic road or unsafe areas. The implementation of this query requires the selection of the exit doors. Let us assume that the possible exit doors are the ones highlighted in Figure 14(a). The design team needs to look at each of these doors and check the outdoor space reached by each door, its land use and activities. This information is only stored in the outdoor CityGML data and is not available for them by only using the IFC data. Through spatial queries, it becomes possible to intersect between the facades where the highlighted exits are located and the land use feature class, and define the undesired unsafe exits to be omitted from the evacuation plan as shown in Figure 14(b).

Figure 14. (**a**) Three possible evacuation exits based only on the IFC building information. (**b**) Combining surrounding information will restrict some exit options.



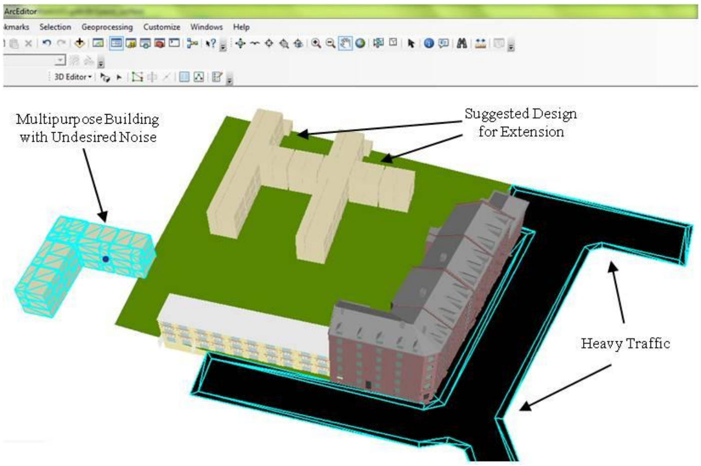
(**a**) (**b**)

- *Q.3. Select outdoor areas that have trees or the applicability/ability to be planted and furnished with a water fountain or noise inhibitors.*

- *Q.4. Find the surrounding buildings that have industrial use and are within a 300 meter distance from the development site of the hospital.*

The third query (Q.3.) concerns the newly added patient ward of 250 beds. The design of the new extension places it to the rear of the main building. However, a suitability analysis is put forward to test the number of beds out of 250 that are further than 300 m away from heavy traffic or noisy places. In order to perform this test, the design team needs first of all to know the uses of all the surrounding buildings, in order to identify those uses with undesirable air or noise pollution. Secondly, transportation data is also needed to identify the levels of traffic around the patient wards. Figure 15 shows the suggested design that needs to be tested *versus* the above-mentioned criteria, and a multipurpose building that has noisy workshops and a group of restaurants. In addition, a heavy traffic road is highlighted as well with a near distance that should be included in the design test.

Figure 15. Suggested extension for the patient ward.


ArcEditor
kmarks Selection Geoprocessing Customize Windows Help
3D Editor
Multipurpose Building with Undesired Noise
Suggested Design for Extension
Heavy Traffic

Having obtained the information about the surrounding built environment, we have been able to create a 3D buffer with 300 m around the heavy traffic and the undesired building. The result is shown in Figure 16. In the figure, the two 3D buffers clearly intersect the tested design, showing that almost 30% (around 77 beds) of the patient wards do not fulfill the testing criteria. We can also create a separate feature class containing only the intersected patient rooms for reporting or further investigation.

Figure 16. 3D Buffer Intersect with the Tested Design

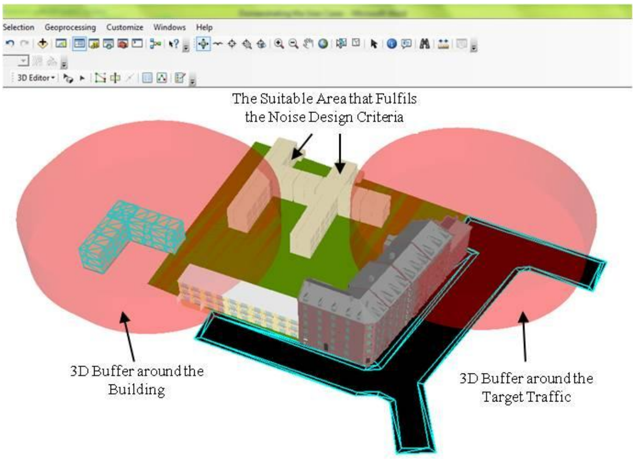

Selection Geoprocessing Customize Windows Help
3D Editor
The Suitable Area that Fulfils the Noise Design Criteria
3D Buffer around the Building
3D Buffer around the Target Traffic

Figure 16 also shows that the area specified for the extension is somewhat limited; fulfilling the criteria for all the 250 beds might be difficult, considering that extending vertically is also limited to five storeys by building regulations in the whole area. Therefore, other alternatives, such as planting trees, constructing noise inhibitors or using landscape features can also be considered for a limited number of beds, which would be defined by the decision-makers.

These alternatives are more design-specific; therefore, they should be based on a specific design combining indoor building information and outdoor built environment data, complemented by additional studies of wind movements and effects, echo systems, natural ventilation, *etc*. In our case, we neither received all of this data nor the final design as it is in the decision phase. Instead of presenting something unrealistic, we prefer to present all the considerations and leave these alternatives for future studies.

6. Conclusions

By both CityGML and IFC standards in Section 3, the objective of this paper is to describe a unified building model (UBM) which encapsulates both the CityGML and IFC models and thus avoids translations between the models and loss of information. Four validation queries have been used to validate the appropriateness of the proposed unified building model. The conclusions may be summarized as follows:

- The fact that the UBM being developed is able to handle CityGML data and IFC data in an apparently seamless way is validated through the case study. This integration is achieved through a unified building model which encapsulates both the CityGML and IFC models. In addition, constraints and enrichment functions are used for populating empty database tables and fields.
- The motivation scenario clearly shows the needs and benefits of having an integrated approach to the modeling of indoor and outdoor spatial features.
- Some limitations in the current setup have been observed. The model has been validated in only a few cases. More case studies and validation tests are required in order to evaluate the full potential of the UBM, for instance, the set of constraints and enrichment functions is not yet fully complete.

The paper succeeded in showing how the research approach provides a starting point for a complete integration of CityGML and IFC. Following the discussion on differences in scale of the data-managed approach shows (for the cases presented) a high consistency of data, clear processes of conversion and tracing, in addition to taking care of all spatial information needed from both standards, without any loss. There is then no need for extending any of the standards by newly added Application Domain Extensions (ADE) in each application.

The UBM is presented both implemented and tested as an integration environment, rather than merely a conversion tool between CityGML and IFC standards. In addition, the UBM implementation shows that different spatial computations and analyses cannot be performed by only IFC or CityGML individually. In the event of a crisis, a unified approach for 3D city modeling is required. For unified modeling, two approaches can be highlighted. The first stores IFC and CityGML data in the UBM.

This is needed, for example, in queries (Q.1 and Q.2) where different processes are required that combine data from both IFC and CityGML. However, in other cases (e.g., Q.3 and Q.4) on-the-fly functions can be performed as inputs to other functions without storing the data on the UBM.

Following our motivation in Section 5, it is worthwhile emphasizing again that our goal was to test the applicability of the UBM's functionality rather than a quantification evaluation of implementation tools, which can of course be an interesting future subject for research. Not only ArcGIS, but similar approaches for the UBM implementation may be based on other commercial or open source platforms.

The next step for this research is envisaged to include the undertaking of elaborated tests in order to evaluate and enhance performance issues. This, however, will focus on extending the model to include other concepts that are available in each standard separately. Enrichment functions also need to be developed for this purpose to allow for the extraction of missed classes.

References

1. Stadler, A.; Kolbe, T. Spatio-Semantic Coherence in the Integration of 3D City Models. In *Proceedings of 5th International ISPRS Symposium on Spatial Data Quality ISSDQ*, Enschede, The Netherlands, 13–15 June 2007.
2. Pu, S.; Zlatanova, S. Integration of GIS and CAD at DBMS Level. In *Proceedings of UDMS'06 Aalborg*, Aalborg, Denmark, 15–17 May 2006; pp. 961–971.
3. Kolbe, T.; Bacharach, S. *CityGML: An Open Standard for 3D City Models*; Available online: http://www.directionsmag.com/articles/citygml-an-open-standard-for-3d-city-models/123103 (accessed on 20 November 2011).
4. Open Geospatial Consortium, Inc. *Candidate OpenGIS CityGML Implementation Specification* (*City Geography Markup Language*); OpenGIS® Engineering Report; Gröger, G., Kolbe, T., Czerwinski, A., Eds.; 2007. Available online: http://portal.opengeospatial.org/modules/admin/license_agreement.php?suppressHeaders=0&access_license_id=3&target=http://portal.opengeospatial.org/files/?artifact_id=22120 (accessed on 20 November 2011).
5. Pontiggia, M.; Derudi, M.; Alba, M.I.; Scaioni, M.; Rota, R. Consequences assessment of hazardous gas releases in urban areas through CFD modeling. *J. Hazard. Mater.* **2010**, *176*, 589–596.
6. Scaioni, M.; Alba, M.; Rota, R.; Caragliano, S. A GIS-Based SW Prototype for Emergency Preparedness Management Applied to a Real Case Study. In *Proceedings of the International Conference on Computational Science and Its Applications*, Seoul, Korea, 29 June–2 July 2009; Gervasi, O., Taniar, D., Murgante, B., Mun, Y.S., Lagana, A., Gavrilova, M., Eds.; Springer-Verlag: Berlin, Germany, 2009; Volume 5592, Part 1, pp. 79–93.
7. Peachavanish, R.; Karimi, H.; Akinci, B.; Boukamp, F. An ontological engineering approach for integrating cad and gis in support of infrastructure management. *Adv. Eng. Inform.* **2006**, *20*, 71–88.
8. IAI. *IFC Model Documentation Webpage. International Alliance for Interoperability*; Available online: http://www.buildingsmart-tech.org/specifications/ifc-releases/summary (accessed on 10 July 2012).
9. Liebich, T. *IFC 2x Edition 4 Model Implementation Guide*. Available online: http://www.buildingsmart-tech.org/downloads/ifc (accessed on 10 July 2012).

10. Kolbe, T; Gröger, G; Plümer, L. CityGML–Interoperable Access to 3D CityModels. In *Proceedings of the International Symposium on Geo-Information for Disaster Management*, Delft, The Netherlands, 21–23 March 2005.
11. Isikdag, U.; Zlatanova, S. Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using BIM. In *3D Geo-Information Sciences*; Lee, J., Zlatanova, S., Eds.; Springer-Verlag: Berlin, Germany, 2009; pp. 79–96.
12. IAI. *BuildingSMART*; Available online: http://www.iai-tech.org/ (accessed on 25 November 2011).
13. 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 *Urban and Regional Data Management*; Rumor, M., Coors, V., Fendel, E.M., Zlatanova, S., Eds.; Taylor and Francis Group: London, UK, 2008; pp. 381–393.
14. Kiziltas, S; Leite, F.; Akinci, B.; Lipman, R.R. CAD and GIS Integration. In *Interoperable Methodologies and Techniques in CAD*; Auerbach Publications: Boca Raton, FL, USA, 2009; pp. 73–110.
15. ISO 16739. *Industry Foundation Classes, Release 2x, Platform Specification*; Available online: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38056 (accessed on 25 November 2011).
16. Benner, J.; Geiger, A.; Leinemann, K. Flexible Generation of Semantic 3D Building Models. In *Proceedings of the 1st International Workshop on Next Generation 3D City Models*, Bonn, Germany, 21–22 June 2005; pp. 17–22.
17. OGC. *Open Geospatial Consortium*; Available online: http://www.opengeospatial.org/ (accessed on 25 November 2011).
18. Open Geospatial Consortium, Inc. *OpenGIS City Geography.Markup Language (CityGML) Encoding Standard*; Gröger, G., Kolbe, T.H., Czerwinski, A., Nagel, C., Eds.; Available online: http://portal.opengeospatial.org/modules/admin/license_agreement.php?suppressHeaders=0&access_license_id=3&target=http://portal.opengeospatial.org/files/?artifact_id=28802 (accessed on 25 November 2011).
19. Open Geospatial Consortium, Inc. *OpenGIS GeographyMarkup Language (GML3), Implementation Specification*; Cox, S., Daisy, P., Lake, R., Portele, C., Whiteside, A., Eds.; Available online: http://portal.opengeospatial.org/modules/admin/license_agreement.php?suppressHeaders=0&access_license_id=3&target=http://portal.opengeospatial.org/files/?artifact_id=4700 (accessed on 25 November 2011).
20. ISO TC211. *Geographic Information/Geomatics*; Available online: http://www.isotc211.org/ (accessed on 25 November 2011).
21. Geographic Information—Spatial Schema; ISO 19107:2003. Available online: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=26012 (accessed on 25 November 2011).
22. Geographic Information—Rules for Application Schema; ISO 19109:2005. Available online: http://www.iso.org/iso/catalogue_detail.htm?csnumber=39891 (accessed on 25 November 2011).
23. Kolbe, T. Representing and Exchanging 3D City Models with CityGML. In *3D Geo-Information Sciences*; Lee, J., Zlatanova, S., Eds.; Springer-Verlag: Berlin, Germany, 2009; pp. 15–31.
24. European Commission. *INSPIRE Directive*; Available online: http://www.ec-gis.org/inspire (accessed on 25 November 2011).

25. Hallberg, D.; Tarandi, V. On the use of 4D BIM in LMS for construction works. *J. Inform. Technol. Constr*. **2009**, *16*, 445–466.
26. Karola, A.; Lahtela, H.; Hänninena, R.; Hitchcock, R.; Chenc, Q.; Dajkadand, S.; Hagströmer, K. BSPro COM-Server-interoperability between software tools using industrial foundation classes. *Energ. Bldg.* **2002**, *34*, 901–907.
27. Isikdag, U.; Zlatanova, S. A SWOT analysis on the implementation of BIM within geospatial environment. In *Urban and Regional data Management*, *UDMS Annuals 2009*; Krek, A., Rumor, M., Zlatanova, S., Fendel, E.M., Eds.; CRC Press: Boca Raton, FL, USA, 2009; pp. 15–30.
28. International Alliance for Interoperability. *Industry Foundation Classes for GIS*; Available online: http://www.iai.no/ifg/index_history.html (accessed on 25 November 2011).
29. Nagel, C. Conversion of IFC to CityGML. In *Proceedings of Meeting of the OGC 3DIM Working Group at OGC TC/PC Meeting*, France, Paris, 9–12 July 2007.
30. Nagel, C.; Stadler, A.; Kolbe, T.H. Conceptual Requirements for the Automatic Reconstruction of Building Information Models from Uninterpreted 3D Models. In *Proceedings of Academic Track of Geoweb 2009 Conference*, Vancouver, Canada, 27–31 July 2009.
31. Van Berlo, L. CityGML Extension for Building Information Modelling (BIM) and IFC. In *Proceedings of Free and Open Source Software for Geospatial* (*FOSS4G*) *Conference 2009*, Sydney, Australia, 4 November 2009.
32. IfcExplorer. *IfcExplorer CityGML Export*; Available online: http://www.ifcwiki.org/index.php/IfcExplorer_CityGML_Export (accessed on 20 November 2011).
33. Safe Software. *FME Desktop Translator/Converter Software*; Available online: http://www.safe.com/products/desktop/formats.php (accessed on 20 November 2011).
34. Tolk, A.; Diallo, S.; Turnitsa, C.; Winters, L. Composable M&S web services for net-centric applications. *JDMS* **2006**, *3*, 27–44.
35. Van Berlo, L.; De Laat, R. Integration of BIM and GIS: The Development of the CityGML GeoBIM Extension. In *Advances in 3D Geo-Information Sciences*; Kolbe T.H., König, G., Nagel, C., Eds.; Springer-Verlag: Berlin, Germany, 2011; pp. 211–225.
36. Jech, T.J. *Lectures in Set Theory*; Springer-Verlag: Berlin, Germany, 1971.
37. Miguel, M.; Jourdan, J.; Salicki, S. Practical Experiences in the Application of MDA. In *Proceedings of Fifth International Conference on the Unified Modeling Language—The Language and Its Applications*, Dresden, Germany, 30 September–4 October 2002.