FLOOR PLANS IN CITYGML

Amol Konde [1], Helga Tauscher [1], Filip Biljecki [1]*, James Crawford [2]

[1] National University of Singapore, Singapore – (amolkonde,helga.tauscher,filip)@nus.edu.sg
[2] Ordnance Survey, Southampton, United Kingdom – james.crawford@os.uk

KEY WORDS: CityGML 3.0, IFC, LoD0, floor plans, storeys

ABSTRACT:

This paper (1) discusses the modelling of floor plans in CityGML; (2) proposes a delineation of multiple variants of indoor LoD0 in line with the current proposal for CityGML 3.0; (3) demonstrates a method to generate CityGML datasets with included floor plans; and (4) explores their usability. The use of an Application Domain Extension (ADE) is being proposed in order to preserve potentially useful information found within detailed building information models (BIM), specifically Industry Foundation Class (IFC), that cannot be stored in CityGML natively. Our work follows the current developments of CityGML 3.0, and based on the discussions in the CityGML Standards Working Group (SWG) it showcases one of the first datasets consistent with the ongoing development of CityGML 3.0 and one that follows the proposals for a new LOD concept and new interior features.

1. INTRODUCTION

In the past decade, there has been a growing interest in modelling the interior of buildings, for various applications such as indoor navigation and disaster management (Biljecki et al., 2015). The standard CityGML 2.0 of the OGC (Open Geospatial Consortium, 2012) allows modelling the interior of buildings and of other thematic classes such as bridges. However, both in research and practice deficiencies have been exposed, such as the lack of datasets containing indoor information and a way to store simple representations. In order to overcome such limitations, researchers and practitioners affiliated with the OGC CityGML Standards Working Group (SWG) have proposed the extension of the level of detail (LoD) concept for the forthcoming CityGML version 3.0 in such a way that, among other advancements, it gives more attention to the modelling of the interior (Löwner et al., 2016). Some of the key points of the proposal are the possibility to model the interior in different LoDs (incl. allowing floor plans in CityGML as LoD0) and combining different LoDs of the interior with different LoDs of the exterior.

In this paper we aim to complement the cited work by discussing multiple aspects related to floor plans in the context of 3D city models and CityGML, a subject that has not been much in focus of research so far. More specifically, we primarily seek to cover the following topics: (i) current developments of modelling floor plans in 3D GIS; (ii) the standardisation aspect of CityGML LoD0 datasets containing building indoor (i.e. discussing whether there are multiple ways to model floor plans); (iii) generation of CityGML datasets containing floor plans (in our work we choose architectural models as the data source); and (iv) potential uses of 3D city models with basic interior (Figure 1).

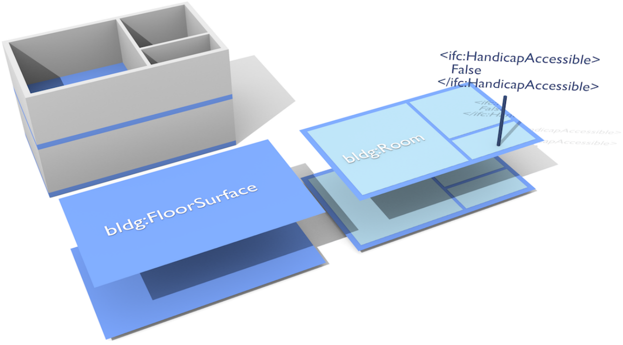

Figure 1. Overview of the project: we generate floor plans in CityGML in multiple variants (bottom and right) of the newly introduced LoD0 proposal by sourcing them from Building Information Model (BIM) data (top left) and preserving certain attributes not natively supported by CityGML but potentially useful for certain use cases. To support this process, we develop an IFC → CityGML workflow and draft an ADE supporting the conservation of some of the semantics from IFC. This image is based on actual data derived in the frame of this project (as it will be explained later in the paper).

2. BACKGROUND AND RELATED WORK

2.1 CityGML LoDs and recent developments

3D building models are modelled according to different LoDs. The CityGML 2.0 standard defines five LoDs (Open Geospatial Consortium, 2012). The two LoDs of relevance to this paper are LoD0 and LoD4. The former is a 2.5D representation containing the building footprint and optionally roof edges (Gröger and Plümer, 2012). Because of its non-volumetric representation, LoD0 is not given that much consideration in 3D GIS — LoD0 datasets usually serve as a stepping stone to generate LoD1 models by extruding them to the height of the building obtaining a volumetric model (Ledoux and Meijers, 2011).

On the other end of the CityGML LoD spectrum lies the LoD4, a detailed architectural model containing interior features such as rooms, openings, and furniture. While the exterior geometry in CityGML can be modelled according to multiple LoDs, the well-known disadvantage of CityGML 2.0 is that there is only one level for the interior (LoD4), i.e. no further differentiation is available allowing modelling geometrically and semantically simpler features, such as storeys or units. This shortcoming has

*Corresponding author

https://doi.org/10.5194/isprs-annals-IV-4-W6-25-2018

been raised by several researchers, who propose the refinement of the interior LoD or extensions to the data model allowing 3D city models with a basic interior geometry that is sufficient for different applications (Billen et al., 2012; Kemec et al., 2012; Tardy et al., 2012; Kang and Lee, 2014; Hagedorn et al., 2009; Boeters et al., 2015; Löwner et al., 2016).

The recent work of Löwner et al. (2016) within the frame of the OGC CityGML SWG and following the development of the upcoming CityGML 3.0, discusses multiple interior LoDs with varying spatio-semantic complexity. Most importantly, the proposal defines the modelling of LoD0 floor plans that includes footprints of rooms and openings, and storeys (e.g. a collection of stacked 2D floor plans). Among others, the focus of this paper is the consideration of storeys. So far (up to and including version 2.0) the only mention of storey was the numerical attribute of a building describing the number of storeys in a building (i.e. `storeysAboveGround`). For further reading on the topic of the delineation of the indoor LoDs the reader is also referred to initial ideas presented in (Benner et al., 2013; Löwner et al., 2013).

Our paper partially builds on top of the work presented in (Löwner et al., 2016) in two ways: (i) it implements the concept by generating an LoD0 dataset containing indoor features, and (ii) it discusses multiple variants of LoD0 datasets.

2.2 Floorplans in 3D GIS and their acquisition

Mapping of the interior of buildings has been a subject of intense research efforts, mostly focusing on the reconstruction of detailed architectural models. Examples of acquisition methods include laser scanning. However, in the GIS context (lower resolution than BIM, i.e. only storey geometry and basic details), there is an extended selection of approaches given the lower requirements for the amount of detail and accuracy. Some methods include: predicting the basic interior from the exterior geometry and other information (Boeters et al., 2015; Loch-Dehbi et al., 2017) and crowdsourced sensor data from smartphones (barometric sensors and WiFi positioning) (Pipelidis et al., 2018; Tuncer et al., 2017).

2.3 Floorplans and LoD in IFC

The Industry Foundation Classes (IFC) constitute a standardised (ISO, 2013) data model to maintain and manage digital information during the design and planning process of construction projects. As such it seems obvious to use IFC data as a source of detailed building information to be included in GIS datasets.

In BIM, the acronym LoD is used for *Level of Development*, denoting the progress in the design and planning process and the successive concretion up to the point of the realization of the building. Ideally, with increasing LoD, the information becomes more dense, but also more reliable and consistent. This prescriptive modeling process is accompanied by an increasing LoD for the ‘specification’ of a building in BIM. Once built, a descriptive modeling process based on surveying reduces the LoD again for the ‘documentation’ of a building in GIS. Thus, particularly during the earlier design stages, geometry of lower LoD may be present and can be preserved in building models. IFC accommodates the notion of different geometrical representation of potentially varying LoD with the concept of representation contexts.

The traditional drawing-based information exchange in construction projects is based on intricate conventions for 2D floorplan representations. CAD applications obtain these representations from their native 3D geometry representations by cutting the building elements of a storey in a specific height above the raw floor or floor finish. The same principle can be applied to IFC data, such as done in the IFCOpenShell engine by applying Open Cascade’s `BRepAlgoAPI_Section` to obtain SVG floorplans [1].

Current efforts to extract information from IFC for GIS application mainly focus on 3D geometry, e.g. Diakité and Zlatanova (2016) extract volumes for indoor navigation, Atazadeh et al. (2017) for cadastre registration. Oldfield et al. (2017) provide an exhaustive overview of relevant geometric IFC representations for cadastre use cases, also focussing on 3D but including 2.5D and 2D representations.

3. EXTENDING THE NEW CITYGML LOD0

While the newly proposed LoD0 (Löwner et al., 2016) is a considerable advancement since the current version 2.0 of CityGML, it still leaves room for different interpretations. In this paper we work on elaborating on the concept and we showcase multiple instances of the interior LoD0 that we have identified.

Biljecki et al. (2014) have shown that the LoD concept comprises a variety of complementing aspects that are impossible to capture with a single ordinal scale. Because these aspects are orthogonal, their metrics can vary independently. Out of the 6 aspects and the respective metrics as proposed by Biljecki et al. we take one into consideration to be the essential one for LoD0: the *dimensionality*. This adheres to the interpretation of Löwner et al. (2013) for CityGML 3.0 where lower dimensionality is the defining aspect of LoD0. With a fixed dimensionality we are now exploring variation in the other aspects, in particular presence of city objects and elements, spatio-semantic coherence, and attribute data.

In Section 3.1 we explore variation in the first two of these aspects and identify multiple ways to model floorplan features and their geometric representations for GIS. In Section 3.2 we show how the concepts are aligned with and how the variants can be implemented in the currently drafted CityGML 3.0. In Section 3.3 we cover the third aspect and demonstrate how CityGML can be extended to include additional semantic attributes that may be related in this context.

3.1 Included features, spatial concepts and geometrical representations

We are first looking at the two selected LoD aspects in isolation and subsequently combine both aspects.

For the *presence of city objects and elements*, we considered different features to be included in the floorplan model and identified three variants that are successively richer in content as shown in Figure 2. When discussing the variants of this aspect, we will refer to the uniquely added concepts.

A This comprises only building storeys and external walls.

B In addition to A, internal walls and rooms are included.

C In addition to B, we consider also doors and windows.

For the *spatio-semantic coherence*, we identified three variants of how geometry is modelled and how features are connected to the geometry as shown in Figure 3 and described below:

[1] http://blog.ifcopenshell.org/2015/07/creating-2d-svg-floor-plans-from-ifc.html

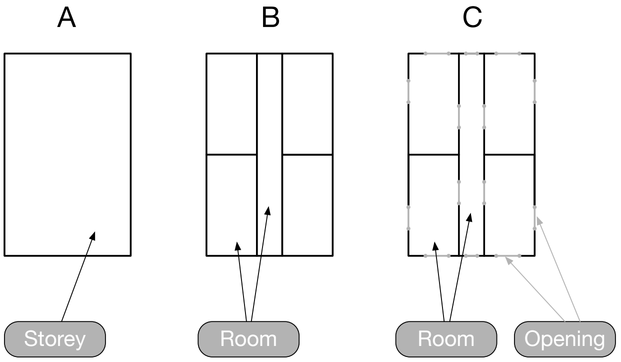

Figure 2. Multiple variants of the realisation of the LoD0 of floor plans. This conceptual illustration does not imply the different ways of the geometric realisation of the features. That is topic is subject of Fig. 3.

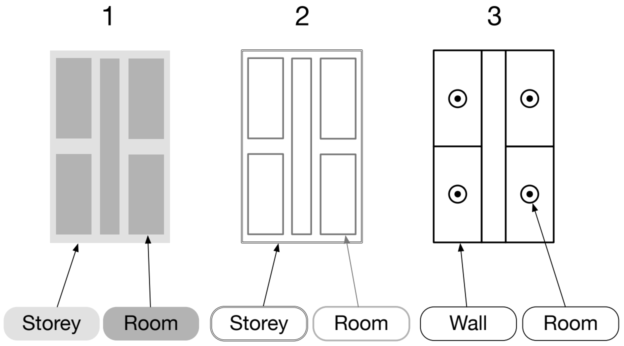

Figure 3. Variants for the geometric representation of floorplan features. This illustration does not imply presence or absence of specific features. For each variant, rooms could be removed or openings could be added (Fig. 2).

1. Included features are represented as footprint surface: the volume of the feature is projected to the floor plane.

2. Included features are represented by their boundaries: vertical boundary surfaces are projected to the floor plane.

3. Included features are represented with an abstraction of lower dimensionality in the floor plane, e.g. axis lines for walls and a marker symbol for rooms.

Owing to the orthogonality of the LoD aspects, we can combine the above variants (A–C and 1–3) freely, such that we obtain a total of 9 variants which we will refer to as A1 to C3.

3.2 An implementation based on the latest CityGML 3.0 proposal

The CityGML 3.0 proposal (Kutzner and Kolbe, 2018) considers LoD0 as suggested by Löwner et al. (2013). Following this implementation, in LoD0 features may be represented as the vertical projections of their 3D geometries into the 2D floor plane.

The proposal introduces a new model of spatial representation that distinguishes between two essential feature types *Space* and *SpaceBoundary*. While the *Space* type comprises features with 3D volume representation, the *SpaceBoundary* type adheres to features with 2D surface representation. These are connected such that *Space* type features are bounded by *SpaceBoundary* type features. Consequently, in LoD0, *Space* type features are represented as surfaces (projected volumes) and their space boundaries as curves (projected vertical surfaces). Storeys, rooms, walls, and openings are specific subtypes of the abstract *Space* type while the semantic surfaces known from CityGML 2.0 are subtypes of the abstract *SpaceBoundary* type.

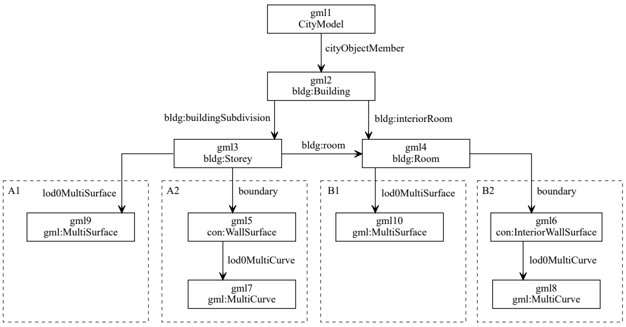

Figure 4. Example object graph of one storey and one room each with a 2D surface and a 2D curve representation

This model of LoD0 representation aligns well with our *spatio-semantic coherence* variants 1 and 2, while variant 3 is not covered by CityGML 3.0. The specification of rooms, storeys and semantic surfaces appears to be coherent and stable and provides a base for our *presence of city objects and elements* variants A and B, but for variant C we need a model of openings. The CityGML 3.0 proposal treats doors and windows either as a constructive element or as a special voiding *Space* subtype. The preliminary version of the proposal that we could study seems weakly defined with regard to the geometric representations of these *AbstractVoid*. We expect refinement of these concepts as a result of discussion in the SWG.

For these reasons, we focus on variants A1, A2, B1, and B2 for our first implementation as follows:

- A1: projected storey volumes as surfaces
- A2: projected outer wall surfaces as curves
- B1: projected room volume as surfaces
- B2: projected interior wall surfaces as curves

Figure 4 shows an examplary CityGML object graph with one storey (*Storey*) and one room (*Room*), each represented according to both paradigms: as a projected volume (*MultiSurface*) and as projected boundary surface (*MultiCurve*). Only the boundary surface representation allows for the assignment of semantic surface types. Note that all room boundaries are interior (*InteriorWallSurface*) while all storey boundaries are exterior (*WallSurface*).

3.3 Semantic attributes

The semantic aspect of interior features in CityGML is basic, e.g. denoting the function of a room and elevation of a storey. This might fall short with use cases that require additional semantics about the interior. For this reason, we draft a sample Application Domain Extension (ADE) to showcase the preservation of a

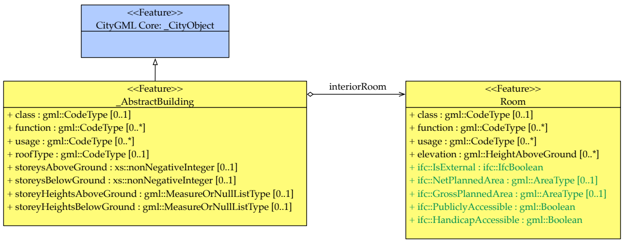
<<Feature>>
CityGML Core: _CityObject
<<Feature>>
_AbstractBuilding
+ class : gml::CodeType [0..1]
+ function : gml::CodeType [0..*]
+ usage : gml::CodeType [0..*]
+ roofType : gml::CodeType [0..1]
+ storeysAboveGround : xs::nonNegativeInteger [0..1]
+ storeysBelowGround : xs::nonNegativeInteger [0..1]
+ storeyHeightsAboveGround : gml::MeasureOrNullListType [0..1]
+ storeyHeightsBelowGround : gml::MeasureOrNullListType [0..1]
interiorRoom
<<Feature>>
Room
+ class : gml::CodeType [0..1]
+ function : gml::CodeType [0..*]
+ usage : gml::CodeType [0..*]
+ elevation : gml::HeightAboveGround [0..*]
+ ifc::IsExternal : ifc::IfcBoolean
+ ifc::NetPlannedArea : gml::AreaType [0..1]
+ ifc::GrossPlannedArea : gml::AreaType [0..1]
+ ifc::PubliclyAccessible : gml::Boolean
+ ifc::HandicapAccessible : gml::Boolean

Figure 5. The UML representation of the sample ADE (additions in green) to conserve relevant information from the IFC source supporting potential use cases such as indoor navigation.

few potentially useful attributes found in the source IFC dataset (Figure 5). We selected a couple of attributes that might benefit indoor navigation, e.g. whether a room is accessible for the public.

4. EXTRACTION OF LOD0 FLOORPLAN INFORMATION FROM IFC

In this section we demonstrate how the CityGML structure introduced in the previous section can be populated from IFC input.

4.1 Implementation, technologies and sample data

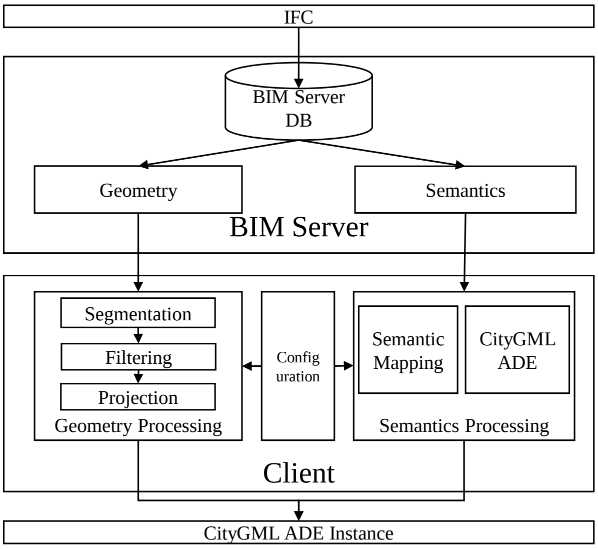

Figure 6. Implementation Architecture to generate an instance of CityGML ADE from IFC.

To generate floor plans in CityGML we employ a conversion utility that is being developed in the broader context of BIM and GIS interoperability. To support varied versions of the IFC to CityGML conversion including custom ADEs, we aim for a runtime-configurable process. To this end, we utilize the software design concept of inversion of control (IoC), also known as dependency injection (DI). DI can increase the software design quality by improving its modifiability, extensibility, and reusability Yang et al. (2008). In addition, DI minimizes coupling between objects making them loosely coupled, which in turn improves testability of the software application Veng (2014). Java supports DI using its full service DI framework called Spring.

We have chosen the Opensource Building Information Modelserver (BIMServer) to read, write and manage IFC. BIMServer provides an API and client libraries to remotely and simultaneously access the IFC data residing in its database, which can be utilized to develop custom applications (Beetz et al., 2010). As BIMServer is a Java implementation, Java became a natural first choice as a programming language for the development of the conversion utility. With Java, we can also utilize citygml4j [2], an open source Java API to read, write and process CityGML data.

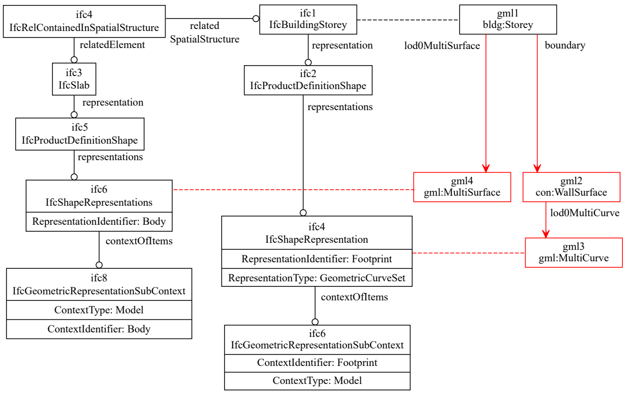

Figure 7. Population of CityGML storeys from IFC storeys.

The developed application consists of three basic modules: The configuration module reads the configuration. Based on this configuration, the geometry processing module performs operations like segmentation, filtering and generalization etc. on the preprocessed geometry to generate a valid surface based geometry representation as required by the intended CityGML instance. The third, the semantic processing module, also takes the configuration as input. Based on the configuration and the input IFC data, it generates the GML object relation graph using the classes defined within the CityGML schema and ADE and assigns the geometry to the GML object.

To verify the extraction and conversion, we created a dedicated IFC4 [3] dataset for a two storey residential building, that includes all required source data. In addition we used the Revit advanced tutorial building exported to IFC2x3 Coordination View.

4.2 Object graph mapping

The IFC concept corresponding to CityGML features is that of elements and spatial elements, both subtypes of products. The spatial structure (upper part in Figure 4) is straight-forward to create from the corresponding structure in IFC: *CityModel* is populated from *IfcProject*, *bldg:Building* from *IfcBuilding*, *bldg:Storey* from *IfcBuildingStorey*, *bldg:Room* from IfcSpace. The relations *bldg:subDivision* and *bldg:roo*m are derived from the aggregation of spatial elements in IfcRelAggregates. *bldg:interiorRoom* is the transitive closure of the two aforementioned relations.

Next, we are focussing on the different variants of attaching geometry to the *bldg:Storey* (A1, A2) and *bldg:Room* (B1, B2) features. IFC allows for multiple geometric and topological representations of products. Here, we are considering the 3D body representation in IFC to populate the projected surface in CityGML (A1, B1) and the footprint representation in IFC to populate the projected boundary curve (A2, B2). To identify the respective geometric contexts, we resort to the implementer's agreement [4] for IFC2x3, assuming this is still applicable for IFC4.

[2] https://github.com/citygml4j/citygml4j
[3] http://www.buildingsmart-tech.org/ifc/IFC4/Add2/
[4] http://www.buildingsmart-tech.org/implementation/ifc-implementation/ifc-impl-agreements/cv-2x3-106

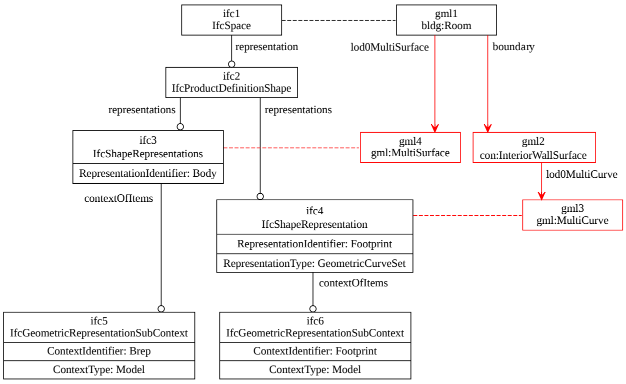

Figure 8. Population of CityGML rooms from IFC spaces.

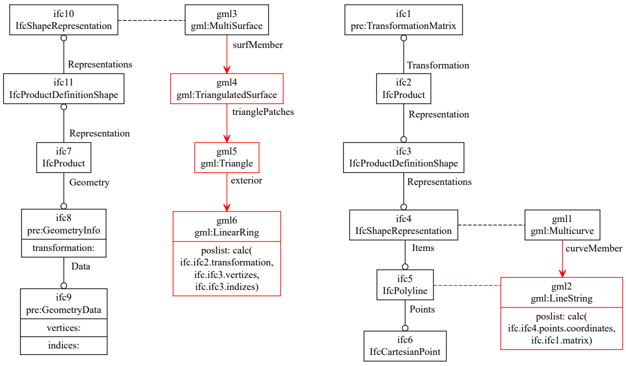

Figure 9. Geometry extraction variants: preprocessed geometry to triangulated surface (A1 and B1, left) or polyline to line string (A2 and B2, right).

Figures 7 and 8 show excerpts of the IFC (left) and GML (right) object graphs. In these diagrams, corresponding entities of the two main graphs are connected through dashed edges. Graph nodes and edges to be created are highlighted in red.

Variant A1: According to the schema, storeys in IFC can have their own 3D body representation, but in practice this is rarely the case. Therefore we use the 3D body representation of the slabs assigned to a storey to extract the required surfaces for the respective storey as shown in Figure 7, left IFC/CityGML branches.

Variant A2: Footprint representations are more common for storeys in IFC as their own geometrical representation. They can be easily converted to the curve geometry representing outer wall surfaces, as shown in Figure 7, right IFC/CityGML branches.

Variant B1: Contrary to storeys in A1, IFC spaces commonly have their own 3D body representation. Therefore we can use this representation directly, see Figure 8, left IFC/CityGML branches.

Variant B2: Similar to storeys in A2, IFC spaces can have footprint representations, see Figure 8, right IFC/CityGML branches.

Figure 9 shows how geometric detail is aquired by further traversing the IFC and CityGML object graphs. The conversion from IFC footprint representation to CityGML MultiCurves (Variant A2, B2, right side of figure) is implemented for the straightforward case of IfcPolylines only. The conversion of IFC 3D body representation to CityGML MultiSurfaces (Variant A1, B1, left side of figure) involves more extensive geometry processing, which will be described in detail in the following section.

4.3 Geometry computation

The IFC geometry, when imported into BIMServer, undergoes preprocessing and is converted to 3D Triangulated Irregular Network (TIN) representation. The developed BIM client retrieves the geometry as a TIN representation and the floorplan in CityGML is a projection of the building elements in 2D for each floor arranged in 3D. To map the geometry in IFC to floor plan in CityGML, we developed an algorithm. It consists of three steps, namely segmentation, filtering and projection.

Step 1: Segmentation As the input solid geometry is represented in triangulated form, the segmentation was performed using the surface normals for each triangle. The developed algorithm as shown in Figure 10 consists of two parts and three possible outcomes as below,

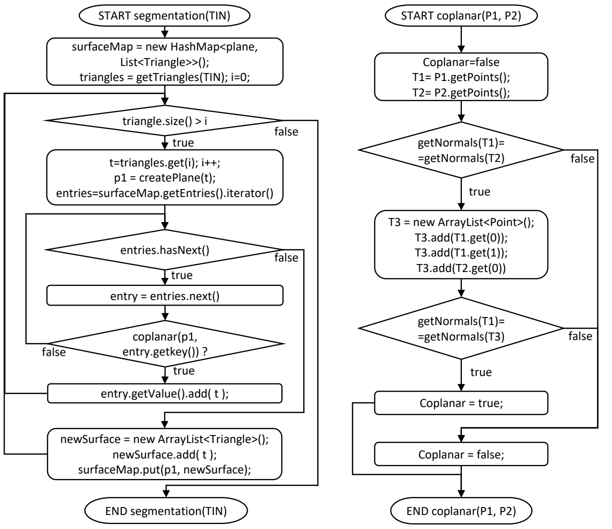
START segmentation(TIN)
surfaceMap = new HashMap<plane, List<Triangle>>(); triangles = getTriangles(TIN); i=0;
triangle.size() > i
false
true
t=triangles.get(i); i++; p1 = createPlane(t); entries=surfaceMap.getEntries().iterator()
entries.hasNext()
false
true
entry = entries.next()
coplanar(p1, entry.getkey()) ?
false
true
entry.getValue().add(t);
newSurface = new ArrayList<Triangle>(); newSurface.add(t); surfaceMap.put(p1, newSurface);
END segmentation(TIN)
START coplanar(P1, P2)
Coplanar=false T1= P1.getPoints(); T2= P2.getPoints();
getNormals(T1)= =getNormals(T2)
false
true
T3 = new ArrayList<Point>(); T3.add(T1.get(0)); T3.add(T1.get(1)); T3.add(T2.get(0))
getNormals(T1)= =getNormals(T3)
false
true
Coplanar = true;
Coplanar = false;
END coplanar(P1, P2)

Figure 10. Segmentation of TIN into surface members.

Neither coplanar nor parallel: In the first part of the algorithm, the surface normals of each triangle T1, are matched with the normals of the surface member S as shown in Figure 11. If the normals are not matched, the triangle is neither coplanar nor parallel to the surface member hence, new surface member is created.

Coplanar: If the surface normals are matched in the first part, the second part is executed. In this, two points from the surface member and one distinct point from the triangle are used to generate another triangle 'abe' as shown in Figure 11. The surface normals of this triangle are then matched with the surface normals of the surface member S. If the normals are matched the triangle T1 and surface member S are coplanar.

Parallel but not coplanar: If the surface normals are not matched in the second part, the triangle T1 and surface member S are parallel but not coplanar, hence new surface member is created.

Step 2: Filtering Multiple functions have been implemented for filtering. Such as, some of the geometry filters generate output based on the area and some based on the elevation of the surfaces. The floor surfaces, rooms, windows and doors were generated in CityGML by extracting the highest elevation surface members of the respective geometry elements in IFC, which are parallel to the ground.

https://doi.org/10.5194/isprs-annals-IV-4-W6-25-2018 | © Authors 2018. CC BY 4.0 License.

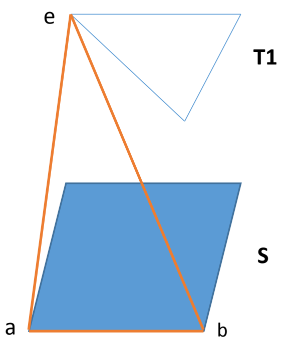

Figure 11. Identification of Parallel and Coplanar Triangles.

Step 3: Projection Projection is applied on the filtered surface members as a final geometry processing step. To represent a floor plan in CityGML, the filtered geometry of all the required building elements like rooms, windows and doors was projected on the floor surface by modifying the elevation (z), so that, it is equal to the elevation of the floor surface.

Figure 12, represents the result of applying the conversion utility with above mentioned geometry processing approach.

Variant A1: Figure 12 (a) and (c) represent the boundary of floors in a two storey building and the ground floor of the revit advanced tutorial model respectively.

Variant B1: Figure 12 (b) and (d) represent the floor plan for both by adding the projected geometry of rooms in the output model.

Variant C1: Figure 12 (e) represents the floor plan of the ground floor of the revit advanced tutorial model by projecting the geometry of rooms, windows and doors on the floor surface.

4.4 Property set extraction

Attributes of a CityGML feature are populated from the IFC attributes of the correlating entity or from property sets assigned to the entity. Figure 13 shows the mapping for some sample attributes of the CityGML3 room feature and the respective ADE extension suggested in Section 3.3.

5. POTENTIAL USE CASES

Since our work is one of the first generally focusing on floor plans in CityGML, we dedicate this section on discussing their usability in terms of different applications. We identify a few applications that may benefit from such data and briefly discuss them.

Emergency, resilience and safety planning Providing ways to further exploit the information contained within detailed BIM (for example, the construction materials and/or fire ratings on doors and fixtures as an obvious starting point) would enable new types of visualisation, spatial analysis and simulations that could lead to various uses. For example, safer design, improvements in emergency response, resilience and safety planning, and also help drive the identification of features and development of requirements for regulatory building plan submission.

Indoor navigation As above, and in the context of individual buildings, the same applies to indoor navigation/routing, where information from detailed building information models can be used to assess mobility routes and barrier free accessibility, leading to improved mobility, living environment and design of indoor spaces, for example. This use case has been supported by our sample ADE (Section 3.3).

Macro-scale energy demand estimation Similar to the previous examples, but in the context of broader scale application and use cases, the abstraction and harmonisation of information contained within detailed BIM spanning multiple buildings and assets, into a simplified floor plan data model also provides potential for new types of visualisation and analysis at the macro-scale. In the energy domain for example, information from detailed building models, could be combined with sensor observations in ways that lead to new insights, and the development of better energy performance targets.

6. DISCUSSION

6.1 Evaluation of the results

CityGML Variant 2 is the only variant, that allows for semantic annotation of surfaces. Variant 3 provides a custom representation of walls, which can facilitate the modeling of openings in floor plans. Moreover, because of its stronger generalization, this variant can keep a very low LoD, even with more features added.

Currently, we use IFC body geometry to extract variant 1 and IFC footprint geometry to extract variant 2, although with a bit more geometry processing, it would be possible to use either for either case. We have shown two ways to extract CityGML LoD geometry from IFC: a) using IFC geometry directly and b) relying on preprocessed triangulated geometry. In first case, every IFC geometry type has to be covered (we only considered polylines), whereas b) is generally applicable due to the unified geometry. However the price of that approach is a dependency on preprocessing engines and larger CityGML files due to triangulation.

6.2 Integration with outdoor data

The generated LoD0 datasets with floor plans can be integrated with the exterior data (Fig. 14). This particular example shows the integration of LoD0-B1 of the interior with the LoD2 of the exterior shell of the building. Combining different LoDs for the interior and exterior was already discussed by researchers in the past a few years (Löwner et al., 2013; Benner et al., 2013). As one can imagine, the permutations of all different combinations are a topic for a separate paper, e.g. conflicts between different granularities and variants would be a complex discussion.

While such a possibility allows flexibility, it entails terminological difficulties. It is a question on how to denote such a combination, e.g. LoD2/LoD0-B1. Coupled with different forms of exterior data (especially those that have been subject of refinements, e.g. see (Biljecki et al., 2016) for an extension defining 16 LoDs for the exterior), it is clear that there are numerous instances of how to model buildings, e.g. the example above would be possible to denote as LoD2.2/LoD0-B1. Obviously, the simple labelling of CityGML LoDs is not possible in such a case.

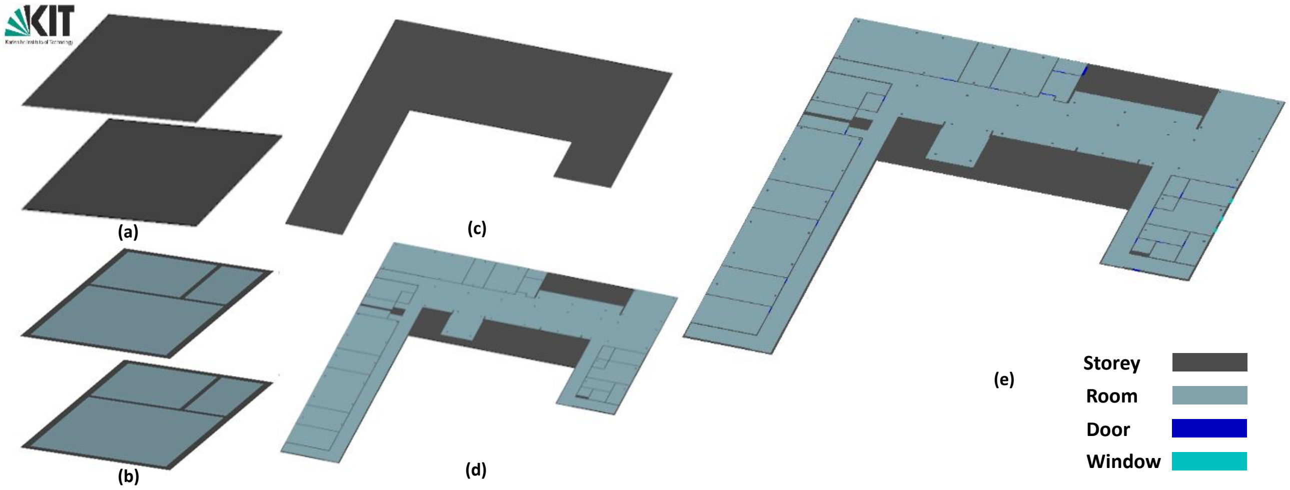

Figure 12. Various variants generated as a result of implemented geometric computation (FZK Viewer).

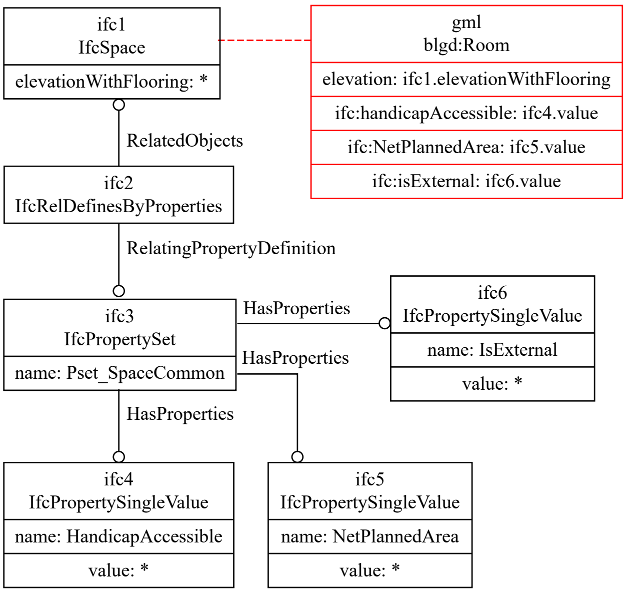

Figure 13. Sample property and attribute extraction for rooms

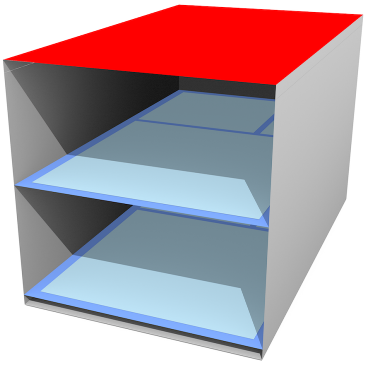

Figure 14. Integration of one of our LoD0 variants (LoD0-B1; see (b) in Fig. 12) of the interior LoD0 with the exterior LoD2. Different combinations of non-equal LoDs have advantages, but entail caution, such as terminological confusion and incompatibility (e.g. protrusions of geometry).

6.3 Future work

Variant C, openings, were only extracted from triangulated geometry. The direct IFC geometry extraction and their proper integration into the CityGML3 model are yet to be investigated. Variant 3, the wall axis representation, is not covered by any CityGML version, thus for its realization we would have to extend the ADE.

For the semantic attribute mapping from IFC property sets, we only proposed a first draft of mapping selected attributes. Further refinement needs to tackle unit conversion and calculations of derived values as well as complex CityGML data types.

To make the conversion generally applicable, we would have to check ‘in the wild’ IFC data for the existence of the required input data and update the IFC2x3 specific part to IFC4.

7. CONCLUSION

This paper discusses several topics related to modelling floor plans in CityGML, a subject that has not been given much attention in the 3D GIS community (e.g. our literature review did not identify a paper solely focused on floor plans). We identified multiple ways to model floor plans in CityGML (as different realisations of the newly proposed LoD0, set to become part of the forthcoming CityGML 3.0 standard), raising awareness of different interpretations of the specifications. Even though we are dealing with relatively simple information such as basic building interior, there may be numerous ways to realise the data. We have shown that it is possible to produce multiple variants of LoD0 floor plans, and moreover – multiple ways to store each variant.

Most importantly, the paper demonstrates a proof of concept on the sourcing of CityGML LoD0 floor plans by leveraging the availability of detailed architectural models in IFC. Because of the limited information, we determine that developing an ADE is a logical step to preserve potentially valuable information from the IFC. A byproduct of the work is that we generate one of the first datasets consistent with the current proposals for CityGML 3.0.

As discussed later in the paper, CityGML datasets containing basic interior have uses in multiple application domains, such as indoor navigation and energy modelling. Thus we hope that giving more consideration to basic forms of indoor data in CityGML will increase their usability in existing or new application domains.

We plan continuing the work by exploring the conversion of IFC to more detailed LoDs of the interior.

ACKNOWLEDGEMENTS

The authors are grateful to the reviewers for their constructive feedback. This material is based on research/work supported by the National Research Foundation under Virtual Singapore Award No. NRF2015VSG-AA3DCM001-008.

References

Atazadeh, B., Kalantari, M., Rajabifard, A. and Ho, S., 2017. Modelling building ownership boundaries within BIM environment: A case study in victoria, australia. *Computers, Environment and Urban Systems* 61, Part A, pp. 24–38.

Beetz, J., van Berlo, L., de Laat, R. and van den Helm, P., 2010. bimserver.org – An open source IFC model server. In: *Proceedings of the CIP W78 conference*, p. 8.

Benner, J., Geiger, A., Gröger, G., Häfele, K.-H. and Löwner, M.-O., 2013. Enhanced LoD concepts for virtual 3D city models. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* II-2/W1, pp. 51–61.

Biljecki, F., Ledoux, H. and Stoter, J., 2016. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems* 59, pp. 25–37.

Biljecki, F., Ledoux, H., Stoter, J. and Zhao, J., 2014. Formalisation of the level of detail in 3D city modelling. *Computers, Environment and Urban Systems* 48, pp. 1–15.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S. and Çöltekin, A., 2015. Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information* 4(4), pp. 2842–2889.

Billen, R., Zaki, C., Servières, M., Moreau, G. and Hallot, P., 2012. Developing an ontology of space: Application to 3D city modeling. In: T. Leduc, G. Moreau and R. Billen (eds), *Usage, Usability, and Utility of 3D City Models – European COST Action TU0801*, EDP Sciences, Nantes, France.

Boeters, R., Arroyo Ohori, K., Biljecki, F. and Zlatanova, S., 2015. Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry. *International Journal of Geographical Information Science* 29(12), pp. 2248–2268.

Diakité, A. A. and Zlatanova, S., 2016. Extraction of the 3d free space from building models for indoor navigation. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-2/W1, pp. 241–248.

Gröger, G. and Plümer, L., 2012. CityGML – Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing* 71, pp. 12–33.

Hagedorn, B., Trapp, M., Glander, T. and Döllner, J., 2009. Towards an Indoor Level-of-Detail Model for Route Visualization. In: *10th International Conference on Mobile Data Management: Systems, Services and Middleware*, IEEE, Taipei, Taiwan, pp. 692–697.

ISO, 2013. Industry foundation classes (IFC) for data sharing in the construction and facility management industries. International Standard ISO 16739, International Organization for Standardization, Geneva, Switzerland.

Kang, H.-Y. and Lee, J., 2014. A Study on the LOD(Level of Detail) Model for Applications based on Indoor Space Data. *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography* 32(2), pp. 143–151.

Kemec, S., Zlatanova, S. and Duzgun, S., 2012. A new LoD definition hierarchy for 3D city models used for natural disaster risk communication tool. In: T. Bandrova and C. König (eds), *Proceedings of the 4th International Conference on Cartography & GIS*, Albena, Bulgaria, pp. 95–104.

Kutzner, T. and Kolbe, T. H., 2018. CityGML 3.0: Sneak preview. In: *38. Wissenschaftlich-Technische Jahrestagung der DGPF und PFGK18 Tagung*, Publikationen der DGPF, Vol. 27, Munich, Germany.

Ledoux, H. and Meijers, M., 2011. Topologically consistent 3D city models obtained by extrusion. *International Journal of Geographical Information Science* 25(4), pp. 557–574.

Loch-Dehbi, S., Dehbi, Y. and Plümer, L., 2017. Estimation of 3D Indoor Models with Constraint Propagation and Stochastic Reasoning in the Absence of Indoor Measurements. *ISPRS International Journal of Geo-Information* 6(3), p. 90.

Löwner, M.-O., Benner, J., Gröger, G. and Häfele, K.-H., 2013. New Concepts for Structuring 3D City Models – An Extended Level of Detail Concept for CityGML Buildings. In: *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science*, Springer Berlin Heidelberg, pp. 466–480.

Löwner, M. O., Gröger, G., Benner, J., Biljecki, F. and Nagel, C., 2016. Proposal for a new LOD and multi-representation concept for CityGML. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* IV-2/W1, pp. 3–12.

Oldfield, J., van Oosterom, P., Beetz, J. and Krijnen, T. F., 2017. Working with open BIM standards to source legal spaces for a 3d cadastre. *ISPRS International Journal of Geo-Information* 6(11).

Open Geospatial Consortium, 2012. OGC City Geography Markup Language (CityGML) Encoding Standard 2.0.0. International standard.

Pipelidis, G., Moslehi Rad, O., Iwaszczuk, D., Prehofer, C. and Hugentobler, U., 2018. Dynamic Vertical Mapping with Crowdsourced Smartphone Sensor Data. *Sensors* 18(2), pp. 480–25.

Tardy, C., Moccozet, L. and Falquet, G., 2012. Semantic alignment of documents with 3D city models. In: *Usage, Usability, and Utility of 3D City Models – European COST Action TU0801*, EDP Sciences, Les Ulis, France, pp. 02011–11.

Tuncer, B., Benita, F. and Tay, H., 2017. Identification of building floors in a 3D city model. In: *14th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT)*, IEEE, Irbid, Jordan, pp. 16–20.

Veng, M., 2014. Dependency injection and mock on software and testing. Master's thesis, Uppsala University, Information Systems.

Yang, H. Y., Tempero, E. and Melton, H., 2008. An empirical study into use of dependency injection in java. In: *19th Australian Conference on Software Engineering (aswec 2008)*, pp. 239–247.