IFC2INDOORGML: AN OPEN-SOURCE TOOL FOR GENERATING INDOORGML FROM IFC.

A. A. Diakite[1,*], L. Díaz-Vilariño[2], F. Biljecki[3,4], Ü. Isikdag[5], S. Simmons[6], K. Li[7], S. Zlatanova[1]

[1] UNSW, School of Built Environment, Sydney, New South Wales, Australia – (a.diakite, s.zlatanova)@unsw.edu.au
[2] Universidade de Vigo, CINTECX,GeoTECH group, 36310 Vigo, Spain – lucia@uvigo.es
[3] Dept. of Architecture, National University of Singapore, Singapore – filip@nus.edu.sg
[4] Dept. of Real Estate, National University of Singapore, Singapore
[5] Dept. of Informatics, Mimar Sinan Fine Arts University, 34380 Istanbul, Turkey – umit.isikdag@msgsu.edu.tr
[6] Open Geospatial Consortium, USA – ssimmons@ogc.org
[7] Pusan National University, Dept. of Computer Science and Engineering, South Korea -– lik@pnu.edu

Commission IV, WG IV/5

KEY WORDS: 3D Indoor, BIM-GIS, LBS, Implementation, ISPRS Scientific Initiative

ABSTRACT:

The interest in 3D indoor models has been continuously growing. Most such models are made available as point clouds or BIM (e.g., IFC), the former being generally provided as unstructured information while the latter comes highly structured and rich in semantic information. IFC models are consequently more suitable for direct use, but they can be very complex and contain too many details, which often raises privacy concerns. IndoorGML is one of the standards for describing 3D indoor space with the purpose of supporting Location Based Services (LBS). It relies on solid scientific concepts and offers a high flexibility with extension mechanisms. It provides a geometric, topological, and semantic description of the indoor which facilitates specifically applications like indoor navigation or facility management. Additionally, it can represent complex indoor environments without compromising privacy, thanks to its high level of abstraction. However, despite its solid conceptual basis, IndoorGML is suffering from a lack of practical tools and remains hard to produce, making it largely unavailable. In this project, we developed an open-source tool named *ifc2indoorgml* allowing to automatically generate IndoorGML models from IFC data. We discuss the workflow and the different development approaches. By making such tool available to the wider public, we expect more 3D IndoorGML models to be created and made freely available for research and development within the spatial community and beyond.

1. INTRODUCTION

IndoorGML is an Open Geospatial Consortium (OGC) standard that focuses on the representation and exchange of data representing the 2D and 3D indoor environment. While other standards dealing with indoor (e.g., IFC, CityGML) target the structural elements of buildings (e.g., walls, floors), IndoorGML relies on a space-based approach (Li et al., 2019), where features are represented as a space. A combination of geometric, topological, and semantic information gives the ability to represent complex indoor environment in a suitable way for several geospatial applications. The first version of the standard (v1.x) was mainly built based on the requirements from indoor navigation (Lee et al., 2014) due to strong and urgent standardization demands for applications such as routing services, indoor emergency response and other related location-based services (LBS). Then it was further improved to clarify some of the concepts (Alattas et al., 2018b) and consequently enriched to include access rights utilising concepts of LADM (Alattas et al., 2017).

In addition to the semantic, geometric, and topological representations, IndoorGML is characterized by powerful concepts such as the cellular description of the space which allows description with different level of granularity, the multilayer concept which allows the representation of different thematic layer (e.g., topographic, sensor coverage), and many more (Lee et al., 2014, Kang and Li, 2017). Throughout the years, it has received considerable attention from the indoor spatial community, for a wide range of use cases, extension proposal, comparison and integration to other standards, and so on (Diakité et al., 2017, Liu et al., 2017, Alattas et al., 2018a, Flikweert et al., 2019, Ledoux, 2020). A second version of the standard is under preparation (Diakité et al., 2020), aiming to address some limitations of the first version and strengthen the coverage of a variety of navigation cases, as well as other LBS applications such as retail, facility or asset management.

However, despite its solid conceptual basis, IndoorGML is lacking practical tools to boost its use by the wider public. Few open-source projects dealing with the standard can be found on GitHub, with the STEM Lab from Pusan University[1] being the most active contributor of the ecosystem. However, those projects generally have different focuses and do not help generating IndoorGML from other existing sources. More recently, a plugin proposing to extract IndoorGML from AutoCAD and Revit[2] models was introduced, going well in the direction of our work, although relying on proprietary formats. In fact, most of the published works related to IndoorGML are at conceptual level, failing to reveal the real strength of the model in real world situations. This is mostly because there is currently no straightforward way to create IndoorGML models. Consequently, there is a considerable lack of available IndoorGML datasets, hindering

* Corresponding author

[1] github.com/STEMLab
[2] github.com/intratech/KICT-Autocad-RevitToIndoorGML

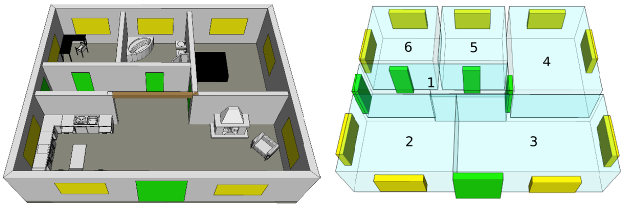


Figure 1. Space representation in IFC. An IFC model that represents a simple house (left) and the 6 room (cyan) as well as doors (green) and windows (yellow) spaces that give adequate IndoorGML representation for spaces and openings. Image from (Diakité et al., 2017).

its wider discovery and adoption.

To mitigate those issues, a project in the scope of the ISPRS Scientific Initiative 2021 brought together several actors of the IndoorGML ecosystem to conduct the development of a basic and robust software tool named *ifc2indoorgml* and aiming to support IndoorGML production. The tool focuses on producing IndoorGML models from common 3D building standards such as IFC, taken advantage of in complementary domains (Biljecki et al., 2021, Noardo et al., 2021), while leveraging existing developments. The consistent analogy between IndoorGML and IFC has been well identified in the literature (Teo and Yu, 2017, Diakité et al., 2017). Both standards provide dedicated classes for the representation of indoor spaces, along with relevant attributes. Other standards such as LADM (van Oosterom and Lemmen, 2015) or more recently, CityGML v3 (Kutzner et al., 2020) are also dealing with spaces, but they were not considered in this project. This paper discusses the workflow adopted to generate IndoorGML from IFC and the different development approaches used in the project. By making such tool available, we expect more 3D IndoorGML models to be created and made freely available for research and development within the spatial community and beyond.

2. APPROACH

IFC models generally contains more information than needed by an IndoorGML model. Nonetheless, relevant matches can be identified between the two standards, making the mapping between their classes feasible. We therefore adopt a direct derivation of IndoorGML classes[3] from IFC classes by identifying their correspondences, similarly to previous work (Teo and Yu, 2017, Diakité et al., 2017).

IndoorGML deals mainly with three types of information: geometry, topology and semantic of the indoor space. The geometry is mainly an explicit description of the extent of the spaces, coming in the form of boundary representation (B-Rep). The topology is the spatial relationship (connectivity) between the spaces which allows the derivation of navigation network for example. Finally, the semantic is the labelling or a more specialized categorization that allows to know the type of a space (room, door, etc.). Note that the core module of IndoorGML does not include semantic. All this information can be found in IFC models, as they may provide explicit representations of spaces, e.g., rooms and openings, as illustrated in Fig.1. IndoorGML can directly use such geometric representations and potentially derive topological and semantic information.

[3] For the IndoorGML classes, we mostly rely on the terminology of the second version of the standard.

IFC classes	Direct/Indirect corresp.	IndoorGML classes
IfcSpace	Direct	CellSpace, NavigableSpace, GeneralSpace
	Indirect	Node
IfcWindow, IfcDoor	Indirect	CellSpace, NavigableSpace, TransferSpace
IfcOpeningElement	Direct	
IfcRelSpaceBoundary	Indirect	CellBoundary
	Indirect	Edge

Figure 2. Possible direct and indirect correspondences between IFC and IndoorGML classes (based on v2).

Figure 3. The three main steps of our implementation.

In the scope of this project, we focused on the following classes of IndoorGML's core module: *CellSpace*, *CellBoundary*, *Node* (or *State*) and *Edge* (or *Transition*). The table in Fig. 2 shows the basic mapping that we used between the two standards. The *CellSpace* class is the one that fits the best to most of the IFC classes because it is used by IndoorGML to represent any space. However, the details provided by IFC (e.g., differentiation between openings and rooms) allows us to include basic semantic classes of the navigation module (e.g., *TransferSpace*).

All the other classes of IndoorGML may require additional processing for the IFC classes to match with their properties. For example, the Node class of IndoorGML needs to be computed from the geometric information of the corresponding space (e.g., centroid of the vertices). Similarly, classes like *CellBoundary* and Edge need to be derived from the *IfcRelSpaceBoundary* class, that describes relevant topological information. While the mapping between the classes may not be a big issue, it is not guaranteed to retrieve the information in an IFC model. For example, the *IfcSpace* class is often missing, making it impossible to derive the rest. Therefore, a hard constraint in enabling the IFC to IndoorGML production will be to assert the presence of *IfcSpace*. All the rest can be derived, even if their corresponding classes are missing in the IFC (e.g., *CellBoundary* and Edge can be derived even if *IfcRelSpaceBoundary* is not provided).

3. IMPLEMENTATION

There are three main steps involved in our implementation workflow (Fig. 3):

1. IFC import to LCC: this part consists in the deployment of an IFC model parser which will ingest the input files and collect geometric, topological, and semantic information from them. The collected information needs to be organised in a structured way for efficient operations on the entities. For this reason, we will use the Linear Cell Complex (LCC), a specialisation of the Combinatorial-Maps data structure (Damiand and Lienhardt, 2014, Damiand, 2022).

2. Data processing: The data organised in the LCC is processed to derive adapted IndoorGML information. For example, the stored geometry will be used to compute the

nodes, the semantic data will be used to classify the *CellSpace* entities properly (e.g., into *CellSpace* or *NavigableSpace*, etc.). But mostly, the topological relationships between the *CellSpaces* are maintained in the LCC.

3. IndoorGML export: once the IndoorGML information is generated, IndoorGML files can be exported. We took into consideration the differences between the versions 1 and 2 (which is still a beta version until its final release).

3.1 IFC import to LCC

3.1.1 IFC parser IFC is an open, international standard (ISO, 2018), meant to be vendor-neutral, or agnostic, and usable across a wide range of hardware devices, software platforms, and interfaces for many different use cases. The IFC schema specification is the primary technical deliverable of buildingSMART International (buildingSMART International, 2020). The most widely used format for IFC in practice is the STEP Physical Format (SPF or IFC-SPF). While there are several other formats possible (XML, JSON, etc.), it is the most compact one that can be read as an ASCII file (plain text). IFC-SPF is based on the ISO standard for clear text representation of EXPRESS data models ISO 10303-21 (buildingSMART International, 2020).

There is an increasing number of open-source libraries offering tools to work with IFC files. In our case, we used IFC++ (Gerold, 2022), an open source IFC implementation for C++, originally developed at the Bauhaus University Weimar and available on GitHub. It provides C++ class models, as well as a reader and writer for IFC files in STEP format. It relies on other C++ libraries (e.g., Carve, OpenSceneGraph, etc.) to handle robustly Constructive Solid Geometry (CSG) operations that may be necessary for explicitly representing some IFC entities.

When parsing an IFC file, we run a pre-loader algorithm to check if *IfcSpace* entities are available so that an IndoorGML can be reconstructed. If no *IfcSpace* is detected, the process will simply stop. Our interest is on a specific set of classes that represent indoor spaces and openings, or the structural or virtual components containing them:

- *IfcBuildingStorey*
- *IfcSpace*
- *IfcDoor*
- *IfcWindow*
- *IfcOpeningElement*
- *IfcVirtualElement*

As the IFC schema follows a strict hierarchy, we use the *IfcBuildingStorey* class to process the building model storey by storey to query and parse the features of interest. Those features are specifically the *IfcSpace*, *IfcDoor* and *IfcWindow* entities. The geometric data of *IfcSpace* entities is directly used as is, after any necessary transformation into explicit B-Rep geometry is done through CSG operations beforehand.

However, *IfcDoor* and *IfcWindow* entities are treated differently. Firstly, in IndoorGML, openings are only considered as intermediate spaces between other indoor spaces. Therefore, an opening not connected to any space is ignored, as it will not contribute to the network. Consequently, in our process, if an *IfcDoor* or *IfcWindow* is not detected in the surrounding of any *IfcSpace*, through the *IfcRelSpaceBoundary* relationship, it will be skipped, unless it is connected to an *IfcOpeningElement* (which is unlikely when *IfcRelSpaceBoundary* is missing).

IfcOpeningElement represents a void within elements with physical manifestation (buildingSMART International, 2020). For a door or a window, it corresponds to the void within the wall containing them, for example. Such void is more adapted to IndoorGML's cell representation because it represents directly the space occupied by the opening and is generally the transition space between two other spaces. For all those reasons, we always check if *IfcDoor* and *IfcWindow* entities are related to any *IfcOpeningElement*, and if so, the geometry of the latter is used to stand for the opening. Related *IfcOpeningElement* to *IfcDoor*/*IfcWindow* entities can be retrieved through their *FillsVoids* attributes. Figure 4 illustrates the difference between the geometry of a regular *IfcDoor* and the void (space) of an *IfcOpeningElement*.

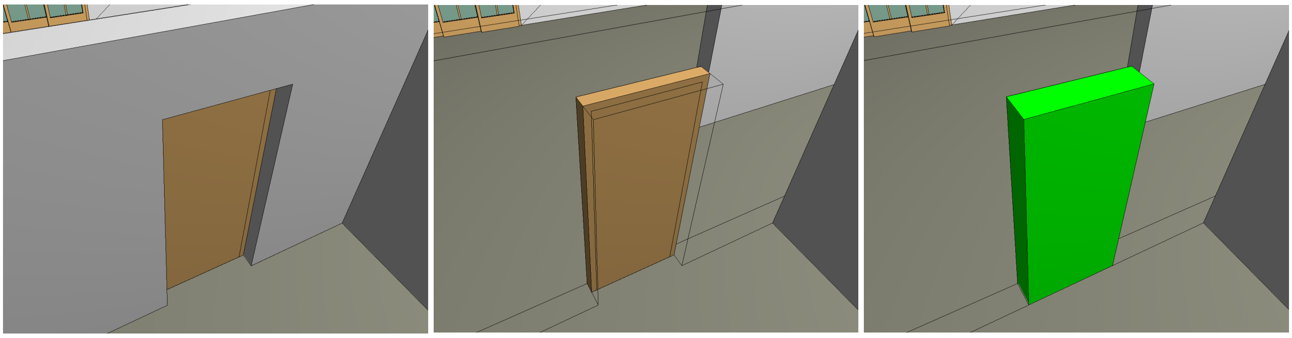
The *IfcVirtualElement* class is meant to represent imaginary boundaries, such as between two adjacent, but not separated, spaces. Nevertheless, it can also be used to fill gaps between spaces that are supposed to be connected but are yet separated because they belong to different storeys. This is illustrated in Fig. 5 where a staircase linking two levels goes through an opening that is represented as an *IfcVirtualElement*. Therefore, to take into account such cases, we have included *IfcVirtualElement* in the classes to inspect for the detection of openings.

3.1.2 IFC to LCC Once the parsing of the IFC file is done, the collected information needs to be organised in a handy data structure for further processing. Since we are only dealing with linear B-Rep geometries, we used a Linear Cell Complex (LCC), a linear embedding of the Combinatorial-Map (C-Map) data structure. One of the main motivations for this choice is the availability of this data structure in the robust C++ library for computational geometry CGAL (The CGAL Project, 2022).

A C-Map is a topological data structure allowing to represent orientable objects of n dimension. In $2D$, it is comparable to the well-known Half-Edge data structure, but it can be extended to n dimensions. A $3D$ LCC is a linear embedding of a $3D$ C-Map, for representing orientable subdivided $3D$ objects with linear geometry: each vertex of the subdivision is associated with a point. The geometry of each edge is a segment whose end points are associated with the two vertices of the edge, the geometry of each face is obtained from all the segments associated to the edges describing the boundary of the face (Damiand, 2022). The subdivision of objects into volumes (3-cells), faces (2-cells), edges (1-cells) and vertices (0-cells) that an LCC provides is suitable to the description of an indoor spaces into cells, as required by the cellular space paradigm of IndoorGML (Lee et al., 2014). Furthermore, it allows performing operations on models while preserving their topological validity and other attributes (e.g., semantic information). Figure 6 shows an LCC resulting from the geometric data in an IFC file, with the 3-cells representing the *CellSpaces* of rooms and openings.

For the purpose of the ifc2indoorgml tool, we defined a $3D$ LCC class that can handle the semantic information that the two standards are using, besides the $3D$ geometric and topological requirements. The following attributes have therefore been defined for the 3-cells of the LCC:

https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-295-2022 | © Author(s) 2022. CC BY 4.0 License.



(a) (b) (c)

Figure 4. (a) A door in an IFC model. (b) The components corresponding to the *IfcDoor* class (brown). (c) The space (green) corresponding to the *IfcOpeningElement*.

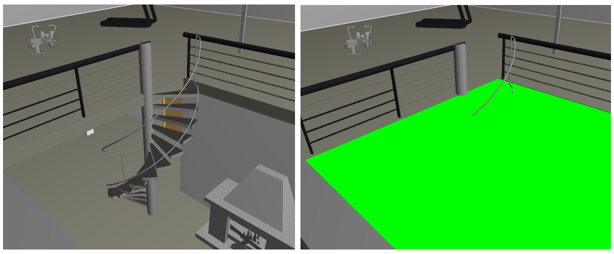


Figure 5. Case of an *IfcVirtualElement*(green, right) linking two storeys through stairs (left).

- *uid*: a unique ID;
- *semClass*: to store the corresponding IFC semantic class (e.g., *IfcSpace*);
- *label*: to store the corresponding IndoorGML class (e.g., *NavigableSpace*);
- *relatedCells*: to store the IDs of all the 3-cells that are adjacent to a given one.

Most of these attributes are populated on the fly, relying on the table in Fig. 2 (direct correspondences are used for *semClass* and *label*), when parsing the IFC file. The *relatedCells* attribute needs more attention and is discussed in the next subsection.

3.2 Data processing (LCC to IndoorGML)

The data processing phase is where collected information from the IFC files is processed to generate valid IndoorGML data. At this stage, we generally have basic information of an IndoorGML's primal space (*CellSpaces*) and eventually some components from the navigation extension (*NavigableSpace*, *TransferSpace*, etc.). Now we mostly need to generate basic information of the dual space (network), but we will also discuss other missing primal space data. In this project, we considered the current version 1.1 of IndoorGML as well as the second version which is still under preparation (Diakité et al., 2020). therefore, two classes are maintained in the tool, one dedicated to IndoorGML 1.1 and another one that inherit from it and brings the IndoorGML 2 specificities available to date.

3.2.1 CellBoundary *CellBoundary* or *CellSpaceBoundary* (v1) is an IndoorGML class used to semantically describe the boundary of each geographical feature in space (Lee et al.,

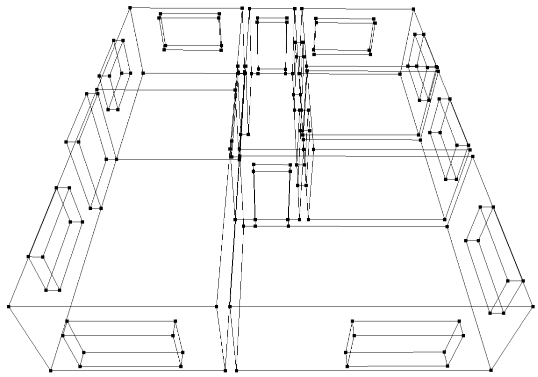


Figure 6. An LCC obtained from an IFC model.

2014). It is the feature that represents the exact adjacency area between two *CellSpaces*, and is thereby linked to the *Edge / Transition* of the network. Because it is not a mandatory feature, an IndoorGML can still be valid without it. This is due to the fact that it can be mitigated by less accurate adjacency checks between *CellSpaces*, which is enough to perform basic navigation for example. The current version of our tool does not process *CellBoundary* yet, but the choice of libraries like CGAL was made to account for future improvements in that direction. The *IfcRelSpaceBoundary* class is what comes the closest to the *CellBoundary*, it is therefore used here to identify adjacency relationships (see section 3.2.3).

3.2.2 Node The computation of *Node* (or *State* in IndoorGML 1.x) is straightforward. For a given 3-cell of the LCC, an iteration through all its vertices (0-cells) is performed to access to their $3D$ coordinates. Their mean coordinate is then computed and used as the centroid of the corresponding *CellSpaces* in the dual space. The risk of such simplified approach obviously is to end up with a centroid located out of the interior of the *CellSpace*. However, for now the focus is on the generation of logical rather than geometric networks (the *Nodes*/*States* are not representing rigorous geographic locations, just their respective *CellSpaces* in the dual space).

3.2.3 Edge The *Edge* (or *Transition* in v1.x) class, as its name suggests, represent the edge in the network, which is the adjacency link between *CellSpaces* in the dual space. *Edge* entities would normally be computed from the *CellBoundary* entities as mentioned above, but since those latter are not implemented yet, we deduce them from information in the IFC files.

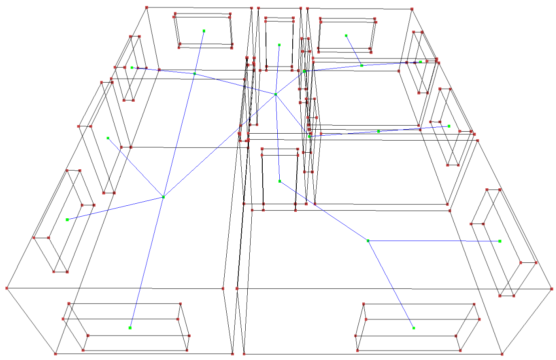


Figure 7. IndoorGML core module extracted from the LCC. The red vertices and black edges are those of the *CellSpaces* (primal space), while the green *Nodes* and the blue *Edges* are the corresponding network (dual space).

The *IfcRelSpaceBoundary* class is defined as an objectified relationship that handles the element to space relationship by objectifying the relationship between an element and the space it bounds (buildingSMART International, 2020). In other words, it provides the information of all related 3-cells to a given 3-cell of an *IfcSpace* in the LCC. For example, it is required for identifying the doors or windows surrounding every spaces, hence its usage for populating the *relatedCells* attribute mentioned in section 3.1.2. Based on that attribute, the *Edges* between any pair of 3-cell can be computed by simply linking their *Nodes*.

However, our experiments have shown that it is not uncommon to have IFC files that are failing to provide the *IfcRelSpaceBoundary* information in their models. In such cases, the tool is likely to miss the *CellSpaces* of the openings (as explained in sec.3.1.1) and will not be able to generate a full adjacency network for the model. Nevertheless, it is still possible to detect adjacent 3-cells by using geometric heuristic (e.g., detecting coplanar faces between 3-cells), although similar approaches are more error-prone. They can be considered in future versions of the tool to mitigate missing IFC data.

3.3 IndoorGML export

As a result from the processes detailed above, we obtain at this stage a valid IndoorGML core module and an additional semantic extension module from the IFC input[4]. The result is shown in Fig. 7 where the primal and dual space elements can be distinguished.

The IndoorGML data is stored in the corresponding IndoorGML classes (depending on the version considered). To handle the import and export of IndoorGML files, we used RapidXML (Kalicinski, 2009), a lightweight XML library for C++. While the import is not implemented yet, we designed specific classes for managing the export of the two versions. However, unlike the exports of the v1.1, the produced v2 files cannot yet be validated and should be considered as beta versions likely to change by the time of an official release of the standard. Another important aspect that will need to be considered in future exporting options of the ifc2indoorgml tool, is the production of other technical models such as the JSON format or SQL for databases (Alattas et al., 2018a).

[4] In IndoorGML, the semantic data for a given application comes as an extension module; although the navigation module is considered as the default one, it is still an extension, thus not part of the core module.

4. INTERFACE AND OPERATIONS

The tool is currently available for download on GitHub[5]. Instructions are provided to guide users for the compilation steps required to build it from scratch. Binaries for 64bits Windows operating system are also provided for easier access. A simple user interface (UI) is provided to simplify the use of the tool. The UI is built on the one provided by the LCC demo that comes with the CGAL library package, which has been customised for the purpose of the project (Fig. 8).

Few options are offered for the manipulation of IFC and IndoorGML files. The File tab provides functions to open an ifc file, or a 3-map file (an XML format for LCC models that can be handy for saving/loading a format other than IndoorGML). The IndoorGML tab regroups operations specific to the standard. For now, only two options are available, one for generating the IndoorGML data from the LCC loaded in the scene and another one for exporting a selected IndoorGML version.

Other general operations on the LCC are implemented under the Operations tab (e.g., merging of coplanar 2-cells (faces) to reduce tessellation, triangulation of 2-cells, deletion of volumes from the model, etc.). The View tab has an option to reset the zoom on the loaded scene for now, while the Help provides some details on the project. It also allows to enable or disable the Volume list on the right-hand side. The latter is provided to help the scene manipulation by listing all the *CellSpaces* that could be retrieved in the input data. The user can also fill or unfill the volumes to switch between plain and wireframe views, or simply hide/unhide them.

5. CONCLUSION AND FUTURE IMPROVEMENTS

We developed ifc2ndoorgml, an open-source tool allowing to generate IndoorGML models from IFC data, with the aim to boost the availability and usage of the standard in the spatial community. The tool can be used to generate valid IndoorGML files describing both primal and dual space elements. Robust C++ libraries were used to ensure smooth functions and a simple user interface is provided for convenience of use. Despite the limited size of the project, the main goal which was to produce a first version of a tool ensuring a successful generation of IndoorGML from IFC is successfully completed. The outcome is a significant step forward for the IndoorGML community and potentially for the indoor LBS applications in general, considering the growing volume of research focused on indoors, leveraging IFC, and 3D modelling at the building scale (Wu et al., 2020, Lim et al., 2020, Liu et al., 2021, Palliwal et al., 2021, Yan et al., 2021).

As this is an initial development, there are opportunities for future improvements. An important improvement will be the implementation of the *CellBoundary* class. This may require sensitive boolean operations for which the exact calculation kernel of CGAL will be of great help for robustness issues. Future versions of the tool should enable the generation of partial IndoorGML models (e.g., a model with the primal space only, or network only, etc.) as discussed in (Diakité et al., 2020). Similarly, options should be provided to add or remove more layers, to benefit from the multi-layering mechanism of IndoorGML. Those options will give more flexibility to the users and help cover more use cases. The UI would also benefit from some

[5] github.com/grid-unsw/ifc2indoorgml

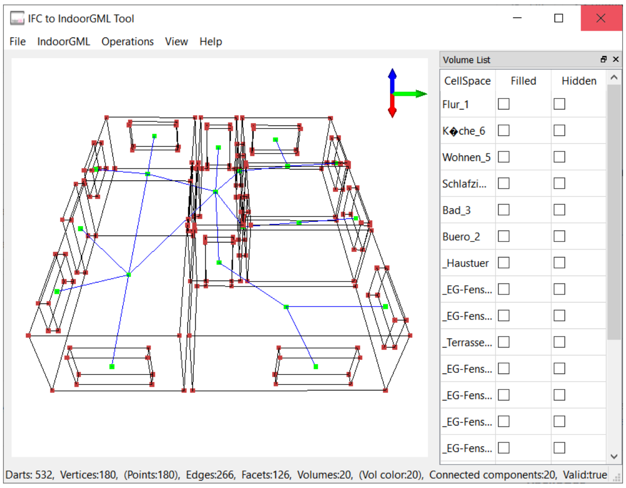


Figure 8. User interface of the ifc2indoorgml tool, with a model loaded in the scene.

improvements, mostly the visualisation and the manipulation of the objects of the scene. By providing more editing functions, the tool can become a full editing tool for IndoorGML models and help enabling the full strength of the standard.

ACKNOWLEDGMENTS

This work was supported by the ISPRS Scientific Initiatives 2021 and the Ministry of Land, Infrastructure, and Transport of Korean government through the iNOUS initiative leb by Pusan National University. The second author has received funding through human resources grant RYC2020-029193-I funded by MCIN/AEI/10.13039/501100011033 y FSE El FSE invierte en tu futuro.

REFERENCES

Alattas, A., van Oosterom, P., Zlatanova, S., Diakité, A. A., Yan, J., 2018a. Developing a database for the ladm-indoorgml model. *6th International FIG Workshop on 3D Cadastres*, International Federation of Surveyors (FIG), 261–277.

Alattas, A., Zlatanova, S., Van Oosterom, P., Chatzinikolaou, E., Lemmen, C., Li, K.-J., 2017. Supporting indoor navigation using access rights to spaces based on combined use of IndoorGML and LADM models. *ISPRS international journal of geo-information*, 6(12), 384.

Alattas, A., Zlatanova, S., Van Oosterom, P., Li, K.-J., 2018b. Improved and more complete conceptual model for the revision of indoorgml (short paper). *10th International Conference on Geographic Information Science (GIScience 2018)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Biljecki, F., Lim, J., Crawford, J., Moraru, D., Tauscher, H., Konde, A., Adouane, K., Lawrence, S., Janssen, P., Stouffs, R., 2021. Extending CityGML for IFC-sourced 3D city models. *Automation in Construction*, 121, 103440.

buildingSMART International, 2020. Industry foundation classes 4.0.2.1. Industry Foundation Classes (IFC) - An Introduction. technical.buildingsmart.org/standards/ifc (08 March 2022).

Damiand, G., 2022. Linear cell complex. *CGAL User and Reference Manual*, 5.4 edn, CGAL Editorial Board.

Damiand, G., Lienhardt, P., 2014. *Combinatorial maps: efficient data structures for computer graphics and image processing*. CRC Press.

Diakité, A. A., Zlatanova, S., Li, K. J., 2017. About the subdivision of indoor spaces in IndoorGML. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(4W5), 41–48.

Diakité, A. A., Zlatanova, S., Alattas, A., Li, K., 2020. Towards IndoorGML 2.0: updates and case study illustrations. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43, 337–344.

Flikweert, P., Peters, R., Díaz-Vilariño, L., Voûte, R., Staats, B., 2019. Automatic Extraction of a Navigation Graph Intended for IndoorGML from an Indoor Point Cloud. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.

Gerold, F., 2022. IFC++, Open source IFC implementation. IFC++ - Open source IFC implementation. ifcquery.com (08 March 2022).

ISO, I., 2018. 16739-1: 2018: Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries—Part 1: Data Schema. *International Organisation for Standardisation: Geneva, Switzerland*.

Kalicinski, M., 2009. Rapidxml. rapidxml.sourceforge.net (09 March 2022).

Kang, H.-K., Li, K.-J., 2017. A standard indoor spatial data model—OGC IndoorGML and implementation approaches. *ISPRS International Journal of Geo-Information*, 6(4), 116.

Kutzner, T., Chaturvedi, K., Kolbe, T. H., 2020. CityGML 3.0: New functions open up new applications. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1), 43–61.

Ledoux, H., 2020. Are your indoorGML files valid? *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 6, 109–118.

Lee, J., Becker, T., Nagel, C., Kolbe, T. H., Sisi, Z., Li, K.-J., 2014. OGC® IndoorGML, Version 1.0.

Li, K.-J., Conti, G., Konstantinidis, E., Zlatanova, S., Bamidis, P., 2019. Ogc indoorgml: A standard approach for indoor maps. *Geographical and fingerprinting data to create systems for indoor positioning and indoor/outdoor Navigation*, Elsevier, 187–207.

Lim, J., Janssen, P., Biljecki, F., 2020. Visualising detailed CityGML and ADE at the building scale. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLIV-4/W1-2020, 83–90.

Liu, L., Li, B., Zlatanova, S., Oosterom, P. v., 2021. Indoor navigation supported by the Industry Foundation Classes (IFC): A survey. *Automation in Construction*, 121, 103436.

Liu, L., Zlatanova, S., Zhu, Q., Li, K., 2017. Towards the integration of IndoorGML and IndoorlocationGML for indoor applications. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 343.

Noardo, F., Krijnen, T., Arroyo Ohori, K., Biljecki, F., Ellul, C., Harrie, L., Eriksson, H., Polia, L., Salheb, N., Tauscher, H., Liempt, J., Goerne, H., Hintz, D., Kaiser, T., Leoni, C., Warchol, A., Stoter, J., 2021. Reference study of IFC software support: The GeoBIM benchmark 2019—Part I. *Transactions in GIS*, 25(2), 805–841.

Palliwal, A., Song, S., Tan, H. T. W., Biljecki, F., 2021. 3D city models for urban farming site identification in buildings. *Computers, Environment and Urban Systems*, 86, 101584.

Teo, T.-A., Yu, S.-C., 2017. The extraction of indoor building information from BIM to OGC IndoorGML. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 167.

The CGAL Project, 2022. *CGAL User and Reference Manual*. 5.4 edn, CGAL Editorial Board.

van Oosterom, P., Lemmen, C., 2015. The land administration domain model (LADM): Motivation, standardisation, application and further development. *Land use policy*, 49, 527–534.

Wu, Y., Shang, J., Chen, P., Zlatanova, S., Hu, X., Zhou, Z., 2020. Indoor mapping and modeling by parsing floor plan images. *International Journal of Geographical Information Science*, 1–27.

Yan, J., Zlatanova, S., Diakité, A., 2021. A unified 3D space-based navigation model for seamless navigation in indoor and outdoor. *International Journal of Digital Earth*, 1–19.