

## Towards indoorgml 2.0

### Updates and case study illustrations

Diakit , A. A.; Zlatanova, S.; Alattas, A. F.M.; Li, K. J.

#### DOI

[10.5194/isprs-archives-XLIII-B4-2020-337-2020](https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-337-2020)

#### Publication date

2020

#### Document Version

Final published version

#### Published in

International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives

#### Citation (APA)

Diakit , A. A., Zlatanova, S., Alattas, A. F. M., & Li, K. J. (2020). Towards indoorgml 2.0: Updates and case study illustrations. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 43(B4), 337-344. <https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-337-2020>

#### Important note

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

#### Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

#### Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

## TOWARDS INDOORGML 2.0: UPDATES AND CASE STUDY ILLUSTRATIONS

A. A. Diakité<sup>1,\*</sup>, S. Zlatanova<sup>1</sup>, A.F.M. Alattas<sup>2</sup>, K. J. Li<sup>3</sup>

<sup>1</sup> Faculty of Built Environment, University of New South Wales, Kensington, NSW 2052, Australia - (a.diakite, s.zlatanova)@unsw.edu.au

<sup>2</sup> Faculty of Architecture and the Built Environment, Delft University of Technology, Julianalaan 134, 2628 BL Delft, The Netherlands A.F.M.Alattas@tudelft.nl

<sup>3</sup> Pusan National University, Kumjeong-Gu, Pusan 46241, South Korea - lik@pnu.edu

### Commission IV, WG IV/5

**KEY WORDS:** IndoorGML, Indoor LBS, Standard, 3D Indoor, Topological Network, Semantic

### ABSTRACT:

Indoor environments differ from outdoor in many aspects. This, added to the limitations faced by other common standards for urban features reinforced the need of setting a dedicated standard for indoor applications. IndoorGML was born in this context to provide the basic concepts, data models, and standard that meet the requirements of indoor spatial applications. Indoor spatial information can be generally classified into two categories: indoor objects such as architectural components (walls, stairs, slabs) and interior facilities (furniture); indoor spaces such as cavities (rooms and corridors) or virtual subdivision (sensor and legal spaces). Handling both information is necessary to support applications ranging from Indoor location-based services (LBS), indoor route analysis or indoor geo-tagging to building and asset management. In this paper, we present the proposed changes to the second version of IndoorGML, under preparation and intended to provide the necessary support for applications using information from those two categories. IndoorGML 2.0 is open to all applications that rely on indoor spaces and require analysis that can be performed on a network, extracted from those spaces utilizing neighbourhood relationships. It follows a model-driven approach, i.e. all concepts are presented by the Unified Modelling Language, from which technical implementations are derived (GML, JSON, SQL, etc.). We present the proposed changes to the previous version, illustrate a way of representing indoor objects other than spaces and discuss several use cases of the standard.

### 1. INTRODUCTION

IndoorGML is an Open Geospatial Consortium (OGC) standard that focusses on the representation and exchange of the 2D and 3D indoor environment (Li, 2016, Li et al., 2019). While the other standards dealing with indoor (e.g. IFC, CityGML) target the structural elements of buildings (walls, floors, etc), IndoorGML relies on a space-based approach, where everything is 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 (and current) version of the standard (v1.0.3) was mainly built based on the requirements from indoor navigation (IndoorGML, 2014) due to strong and urgent standardization demands for applications such as routing services, indoor emergency response and other related location-based services (LBS).

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 to describe the space with different level of granularity and the Multilayer concept (Becker et al., 2009) which allows the representation of different thematic layer types (e.g. topographic, sensor coverage, etc.) for the same space (IndoorGML, 2014, Kang, Li, 2017). Throughout the years, it received considerable attention from the indoor spatial community, for a wide range of use cases, extension proposal and comparison/integration to other standards, etc. (Hwang et al., 2012, Li, Lee, 2013, Salvetti et al., 2015, Ryoo et al., 2015, Mirvahabi, Abbaspour, 2015, Diakité,

Zlatanova, 2016, Liu et al., 2017, Flikweert et al., 2019). However, a lack of details in the standards description, a focus on GML scripting and a strong focus on navigation-related applications, limited the strength of the standard in covering a variety of navigation cases, as well as other LBS applications such as retail, facility or asset management. Despite the strong core concept, several notions of the v1.0.3 need to be corrected or adjusted, to account for a growing interest and user community.

To address those limitations, the new version of the standard, IndoorGML 2.0 is in preparation (Alattas et al., 2018b). While most of the concepts are maintained (cellular space, semantic, geometry, topology, multilayer), several aspects are clarified and revisited for improvement. The changes are reflected in the new UML diagram of the standard, where all the classes, their attributes and their interactions are represented (see Fig. 1). Some of the main changes include: renaming of classes, redefinition of *TransferSpace* and *GeneralSpace* notions, exclusion of *Thin door* concept, introduction of the *level* attribute in the *CellSpace* class and redefinition of the geometry as an attribute rather than a separated class. The representation of the topological relationship between spaces by the Poincaré Duality is still maintained for its convenience in domains beyond navigation. This implies the preservation of spatial constraint that the Poincaré duality involves, such as the non-overlapping of the *CellSpace* elements.

Furthermore, other aspects of the standard are also being developed to support new applications. For example, the current version of IndoorGML focuses on navigable spaces and does not provide a representation of indoor features that are not spaces by essence and that are critical to LBS applications, such

\* Corresponding author

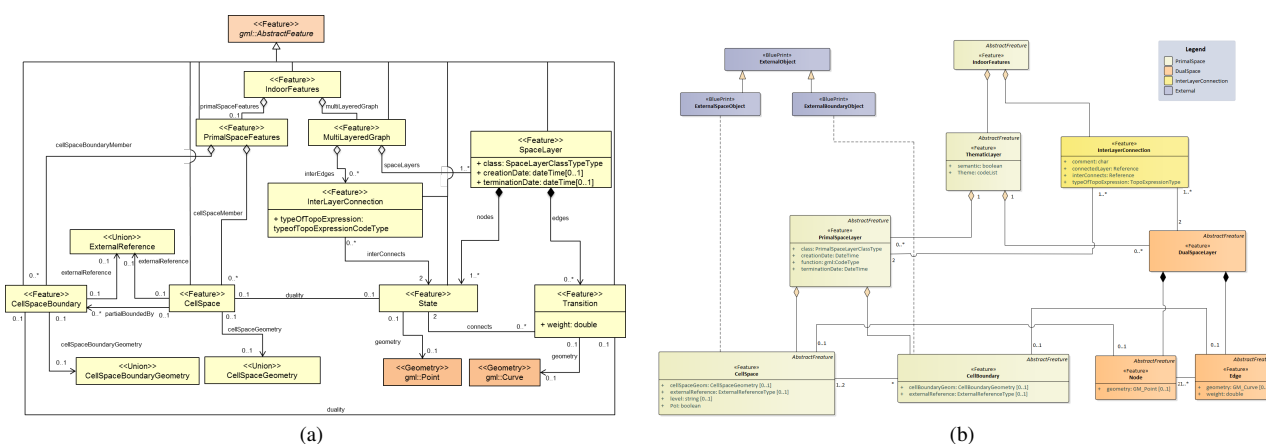


Figure 1. UML diagrams of the Core module of (a) IndoorGML 1.x and (b) IndoorGML 2.0.

as Point of Interests (PoI). This paper presents a concept for handling PoI through space subdivision, which allows a CellSpace to be subdivided to a suitable granularity level. The approach is based on the Flexible Space Subdivision (FSS) framework (Diakité, Zlatanova, 2018) in combination with the multilayer concept. The FSS-based space subdivision allows to introduce the description of complex indoor features solely on the basis of the space that they occupy (Object-Space). The FSS framework allows to explore many possibilities for the *Non-NavigableSpace* class, which was mentioned but not elaborated in the previous version.

This paper further informs the geospatial community about the modifications considered for the next version of the IndoorGML standard. Details of the changes will be provided along with their main motivations. Additionally, several use case scenarios will be presented in order to discuss how they can be implemented on the basis of the new changes. Those use cases include: the extraction of *CellSpace* elements from common compatible standards such as IFC, the generation of different type of models with different complexity (topology only, topology and semantic, geometry and semantic, etc.) to illustrate the flexibility of the standard.

## 2. OVERVIEW OF CHANGES

Several changes was brought compared to IndoorGML1.x. Some classes were and attributes were renamed, modified or added in the standard for the sake of simplicity, clarity and genericness. To facilitate the comparison, Figure 1 illustrates the core module of IndoorGML1.x and IndoorGML2.0 that integrates the changes detailed in the following subsections.

## 2.1 Renamed classes

This subsection addresses the classes that have renamed relations. Generally, naming of relations (as it can be seen in Fig. 1(a)) is removed and when needed added as attribute.

**PrimalSpaceLayer** and **DualSpaceLayer** The two principal classes of the core module are the dedicated to the representation of the primal space (3D geometry) and the dual space (3D topology). While they used to be referred to as *PrimalSpaceFeatures* and *MultiLayeredGraph* in IndoorGML1.x, they are now renamed respectively as *PrimalSpaceLayer* and

*DualSpaceLayer*. This choice is obviously motivated by a clarification goal.

**CellBoundary** According to the definition of IndoorGML, the primal space describes the indoor space as an aggregation of space units called cells. Those 3D space cells and their boundaries were respectively called *CellSpace* and *CellSpaceBoundary*. In the IndoorGML2.0 proposal, the latter is renamed as *CellBoundary*.

**Node and Edge** One of the common confusion involved in the use of IndoorGML1.x is related to the description of the topological network in the dual space. In fact, the network, which is derived based on the Poincaré Duality is composed of nodes representing the cells of the primal space and edges representing their adjacency and, under certain conditions, their connectivity. For the sake of completeness, one should note that ‘connectivity’ relationship, which is of importance for navigation applications, is a type of neighbourhood relationship that corresponds to the adjacency graph of navigable spaces only.

However, those graph components were called *State* for a node and *Transition* for an edge in IndoorGML1.x. While such naming has not been motivated in the standard definition, it refers to navigation cases in which people are commonly staying in spaces, which gives their state. While the movement from one space to another is short-term and represents their transition. These notations then becomes confusing for non-navigation applications. Therefore, they are renamed into *Node* and *Edge* in IndoorGML2.0.

## 2.2 Modified classes and attributes

This subsection discusses the classes and attributes that have been moved from one place to another of the standard, to improve its efficiency and flexibility.

**Geometry of cells and external references** The UML diagram of IndoorGML1.x suggests that there are dedicated classes for the geometric information related to the cell spaces and their boundaries, as well as the network. The concept of keeping geometry in separate classes has its roots in topological representations, i.e. common faces are to be maintained once in the model and they are 'reused' by the neighbouring solids. However, the indoor spaces used for navigation and other LBS applications most commonly do not share faces, which results in storing all

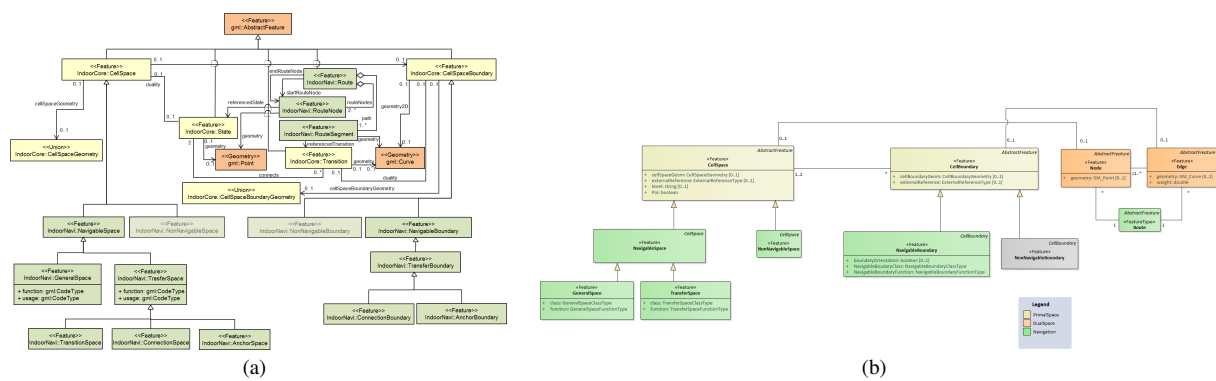


Figure 2. UML diagrams of the Navigation module of (a) IndoorGML 1.x and (b) IndoorGML 2.0.

faces of solids. Hence, the classes *CellSpaceGeometry*, *CellSpaceBoundaryGeometry* and *Geometry* (for the network) can be simply considered as attributes of those features. This is therefore clarified in the UML of IndoorGML2.0, where any *CellSpace*, *CellBoundary*, *Node* and *Edge* is provided with an attribute *CellSpaceGeom*, *CellBoundaryGeom* and *Geometry* (for *Node* and *Edge*). Similar process is done for the external references for which an attribute *externalReference* has been added to *CellSpace* and *CellBoundary* classes.

**InterLayerConnection** The *InterLayerConnection* class is dedicated to the description of the links between different layers. While in IndoorGML1.x it was specifically dedicated to the connection between network components (*MultiLayeredGraph*), it is extended in IndoorGML2.0 to also account for interlayer connections between elements of the primal space. This is motivated by the fact that the notion of layer is improved in IndoorGML2.0. A layer can be solely composed of primal space features, dual space features or a combination of both. In the previous version of the standard, only interconnection between dual spaces of different primal spaces was possible.

### 2.3 New class and attributes

The need to improve the information organisation in the standard and its support for indoor features other than spaces led to the introduction of one new class and several attributes in the core module.

**ThematicLayer** With IndoorGML1.x, a model is composed of one *IndoorFeatures* instance which aggregates two lists: one instance of *PrimalSpaceFeatures* which aggregates all the *CellSpace* elements existing in the model, and one instance of *MultiLayeredGraph* aggregating *SpaceLayer* components as well as *InterLayerConnection* instances. As mentioned previously, this enables the multi-layer mechanism only for the dual space (the networks). To make it possible to efficiently store several layers of geometry (*CellSpace*) and/or topology (*Node* and *Edge*), a reorganisation is needed. The *ThematicLayer* is proposed for IndoorGML2.0, as an aggregation of *PrimalSpaceLayer* and *DualSpaceLayer* instances to help defining layers separately with the ability to contain fully or partly components of the core module. The class comes with two attributes: *semantic* and *Theme*. The *semantic* is set as a boolean as it is simply an indication that there is semantic information associated to the *PrimalSpaceLayer*. Because the navigation module is currently the only semantic module available, a boolean is enough to indicate its presence for now. This is however susceptible to

evolve in the future (e.g. into a *codeList*). The *Theme* attributes determines whether the layer is of type *TOPOGRAPHIC*, *SENSOR*, *LOGICAL*, *TAGS* or *UNKNOWN*. This *codeList* is already in use to determine the class type of a *SpaceLayer* entity in IndoorGML1.x. Therefore, it may also be subject to change in the future, once more relevant themes will be identified (e.g. we could introduce a *LEGAL* theme considering related work in a land administration standard (Alattas et al., 2017)).

**PoI** The notion of PoI occupies a central place in LBS applications. As its name says, it corresponds to locations of interest for a specific purpose on a map. A 3D indoor map such as IndoorGML should therefore be able to represent such notion (Park et al., 2016). While recent works are investigating potential additional IndoorGML modules for this purpose (Claridades et al., 2019), we present here a preliminary approach that can directly fit to the current core module of the standard.

Because every space comes down to a *CellSpace* in indoorGML and any space can be a location of interest, may it be empty or occupied by a physical object (see Section 3), we propose to append the attribute *PoI* to the *CellSpace* class of the core module. For the moment, the attribute is just a boolean allowing to tag a cell as a PoI and implement specific considerations with respect to it in the corresponding application. Thanks to the relations between the primal and dual space in the standard, one can use the information on a navigation network for example, and put a higher weight of edges leading to nodes associated with PoI cells.

### 2.4 Changes related to the Navigation module

Several changes were considered for the navigation module as well, alongside with the core module's modifications. Figure 2 provides a global illustration of the changes by putting the two UML diagram against each other. The most prominent change is related to several classes aggregating to the *TransferSpace* class and that have been removed; namely the *TransitionSpace*, *ConnectionSpace* and *AnchorSpace* classes. While willing to provide more details, these latter are not making a fundamental difference with their mother class, leading to more confusion in their use. Instead, the *TransferSpace* is simplified to only doors and windows, where an agent can stay for a very short time to get to another space. Everything else will be simply considered as a *GeneralSpace*, including corridors, stairs, elevators, etc. Attributes associated to them, such as the *function* attribute can be used to provide more detail regarding their specific nature.

Furthermore, the concept of 'Thin door' which was discussed in IndoorGML1.x and motivates the children classes of *Navig-*

*ableBoundary* (see Fig.2(a)) is removed in the new UML. This is to keep the (more intuitive) concept of considering everything as a space strong throughout the standard. However, cases that the notion of 'Thin door' was meant to address can still simply be handled by the *NavigableBoundary* class.

Additionally, two other classes dedicating to the routing were also removed, namely the *RouteSegment* and the *RouteNode*. While they are just convenient classes to explicitly store routing information, they are adding complexity to the standard for a simple task. We believe the information they provide can be directly embedded in the *Route* class as attributes. This option will therefore be tested and the output will determine the relevant attributes to add to the class.

### 3. REPRESENTING INDOOR OBJECTS THROUGH SPACE SUBDIVISION

IndoorGML1.x does not provide any mean to represent indoor features other than spaces. For this reason, the indoor objects are neglected in the model, and only the room containing them are generally represented. This is a considerable limitation for any application requiring a more precise information regarding location and properties of such indoor features, as they generally stand for PoIs in LBS applications. On the other hand, a room, or a specific part of a room can also be PoIs and provide specific functions. On the other hand, IndoorGML supports two key concepts that can be used to overcome this gap, namely the multi-layer (previously mentioned) and the space subdivision concepts (Zlatanova et al., 2013, Jung, Lee, 2015).

It is under all those considerations that the Flexible Space Subdivision (FSS) framework was first introduced in (Diakit , Zlatanova, 2018). While the focus of the paper was more in the definition of the framework's principles with respect to fine-grained indoor navigation, here we discuss with more details the integration of the FSS in IndoorGML, and illustrate an extension proposal for the navigation module, allowing the consideration of indoor features.

The FSS first assumes that the indoor environment can be subdivided on the basis of the objects that it contains. These latter can be categorized into three types, depending on their locomotion (ability to move independently or be moved): static, semi-mobile and mobile. Based on the objects, their locomotion, properties and functions, the FSS subdivides the indoor space into three main subspaces:

- Object spaces (O-Spaces) representing the spaces physically occupied by static and semi-mobile objects;
- Functional spaces (F-Spaces) corresponding to the spaces involved by the function of an O-Space;
- Remaining free spaces (R-Spaces) corresponding to the spaces that can be assumed freely available for agents' navigation.

The advantage of this framework is in its ability to represent the full indoor environment solely on the basis of spaces, which comes very convenient for a space-based standard such as IndoorGML. Figure 3 shows the proposed UML diagram to integrate the FSS into IndoorGML. This UML is different from the one proposed in (Diakit , Zlatanova, 2018) in the sense that

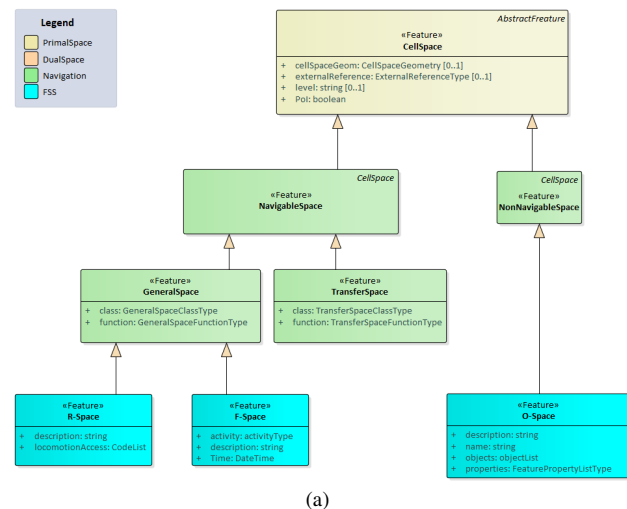


Figure 3. UML diagrams of the FSS integration to the Navigation module.

it takes into account the classes of the navigation module of IndoorGML, rather than just its core, and provides more attributes to the new classes. Furthermore, as mentioned before, the attribute *Poi* has been added to the *CellSpace* class, as any cell space can effectively be called as such. Further details of the PoI will be handled by the specialized classes derived from the FSS.

#### 3.1 Indoor objects as *NonNavigable* spaces

Because the O-Space concept involves physical occupancy, that class *O-Space* becomes a specialization of the *NonNavigableSpace* class that was not implemented in IndoorGML1.x. The *NonNavigableSpace* and *O-Space* become thereby the classes that describe indoor physical objects on the basis of the indoor space that they occupy. Several attributes are proposed to enable the association of more details to such objects. The *description* attribute is a string that can be used to describe any detail concerning the O-Space. The *name* attribute is a sting providing a specific name to the O-Space. the *objects* attribute provides a list of the physical objects that the O-Space contains. This can be one or several objects, depending on the aggregation approach adopted at the creation of the O-Space (e.g. several nearby component can be aggregated into one O-Space). Finally, the *properties* attribute allows to describe a list of specific properties for each object in the O-Space (e.g. weight, state of use, etc.). All those attributes can be subject to change in future versions of the standard, as the research for a comprehensive understanding of their necessity is still undergoing.

Figure 4(b) illustrates an O-Space resulting from the aggregation of the O-Spaces of the chair and the table in Fig. 4(a), because these latter intersect each other when not aggregated (which is an invalid configuration in IndoorGML). The resulting O-Space could carry the following information for example:

- description: "Office desk configuration"
- name: "Bureau"
- objects: [chair, desk]
- properties: [weight: 5kg, state: functional, weight: 15kg, state: functional]



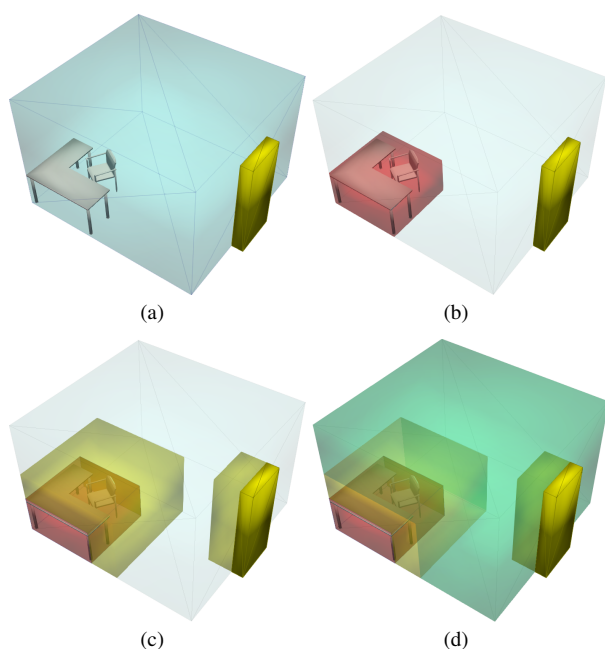


Figure 4. Example of the FSS implementation on a simple 3D room model scenario. (a) Room space (light blue) with a chair, a desk and a door (yellow). (b) O-Space encapsulating the furniture. (c) F-Spaces (yellow) of the O-Space and the door. (d) R-Space (green) of the room.

### 3.2 Navigable spaces as F-Spaces and R-Spaces

The two other subspaces induced by the FSS, although related to the O-Space, concern mainly the spaces not occupied physically. In fact, F-Spaces represent the space that may be physically occupied as a consequence of the function of an indoor object, located in an O-Space. The R-Spaces are just the free space remaining once the O-Spaces and F-Spaces are identified and subtracted from the space of the room. For this reason, both *F-Space* and *R-Space* classes are specialization of the *NavigableSpace* class. More specifically, they are specialization of the *GeneralSpace* class which now covers (in IndoorGML2.0) any type of indoor space while the *TransferSpace* class only describes openings.

According to the definition of the F-Spaces in (Diakit , Zlatanova, 2018), any opening space can be considered as an F-Space in itself. However, we do not consider yet consider the *TransferSpace* class as the generalization of any FSS subspace. This may also change in the future. Attributes considered for an *F-Space* are: the *activity*, which describe an activity type that may occur within the space (e.g seated work, queue, gathering, manoeuvre, etc.); the attribute *description* is similar to the one of O-Spaces and the attribute *Time* determines a specific time and date when the F-Space is susceptible to be occupied, thus not reliable for navigation, unless it is the destination itself.

Figure 4(c) provides an illustration of such subspaces (in yellow). We can see that a corresponding F-Space is computed for the door, as well as the O-Space containing the furniture. For this illustration, it was computed as a simple buffering space around the objects in order to ensure a space for activities. Deeper investigations on automatic approaches to generate better customised F-Spaces based on the properties of the objects are still ongoing.

R-Spaces also come with specific attributes. Apart from the *description*, it can carry the *locomotionAccess* information, which gives hint on the type of suitable locomotion that can take place within the concerned R-Space (pedestrian, wheelchair, drone, etc.). Figure 4(d) shows (in green) the R-Space of the example model and all the FSS subspaces resulting from it.

### 3.3 Multi-Layer representation

The multi layer mechanism of IndoorGML provides a convenient way to implement the FSS framework for adequate indoor features description and space granularity, without compromising the original topographic data. Indeed, it gives the ability to the standard to store different geometric, topological and semantic information in different layers, so as to allow subdivisions while avoiding incoherences or hard constraints within a same layer. In fact, the permitted topological relationships in one layer are 'meet' and 'disjoint' only. In contrast, the relationships between two layers can have all eight possible relationships between solids (in 3d) and polygons (in 2D), as described in (Zlatanova, 2000).

Figure 5 shows two different layers (b) and (c) created from the same input 3D model (a). Practically speaking, *CellSpace* instances are created in Fig.5(b) to store the geometry and related information of the spaces. Note that the furnishing elements are ignored and only remains the space of the room and the one of the door. One could go further and create *GeneralSpace* instance for the former and *TransferSpace* for the latter. Those instances are then aggregated to a *PrimalSpaceLayer* instance, optionally along with their corresponding *Node* and *Edge* instances aggregated to a *DualSpaceLayer* instance. Both primal and dual layers would then be aggregated to a specific *ThematicLayer* with a topographic *Theme* and true *semantic* value, if any, as attributes.

A similar process can then be applied for a new topographic *ThematicLayer* with the FSS classes as semantic, resulting to what is illustrated in Fig. 5(c). Note that the furniture are still not physically represented, but the space that they occupy and that may be implied by their use can be distinguished. These two different layers can then be linked with an *InterLayer-Connection* instance (see Fig. 1). The attribute *typeOfTopo-Expression* can describe the topological relationship between them, without being restricted to the adjacency link used by the Poincar  duality (e.g. *CONTAINS*, *OVERLAPS*, *INTERSECTS*, etc.). It then becomes possible to specify that the first layer *CONTAINS* the one with the FSS classes.

## 4. USE CASES

In this section, we provide some case illustrations of anticipated use case of IndoorGML2.0 and demonstrate the flexibility enabled by the new UML diagram of the standard. We explore different scenarios that involve the use of other common standards as data source and lead to specific use of the classes available. For the matter, we will use the BIM (IFC) model shown in Fig. 6, focusing mainly on its components that are relevant to IndoorGML. Note that the furniture are deliberately skipped here as the examples shown in the previous section are sample of the same model.

### 4.1 Geometry and semantic

In this example, we show the case of data extraction from a BIM model (IFC) that is stored as geometry with the *CellSpace* class

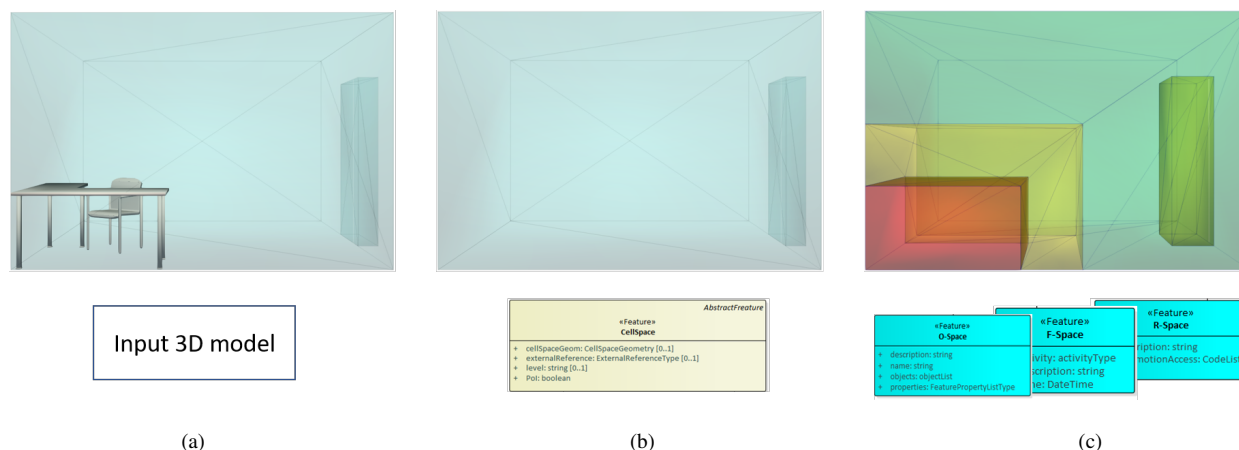


Figure 5. Different IndoorGML layers of a same space. (a) Original 3D model of a room space with furniture. (b) CellSpaces forming the first topographic layer. (c) Second topographic layer composed of FSS subspaces of the original model.

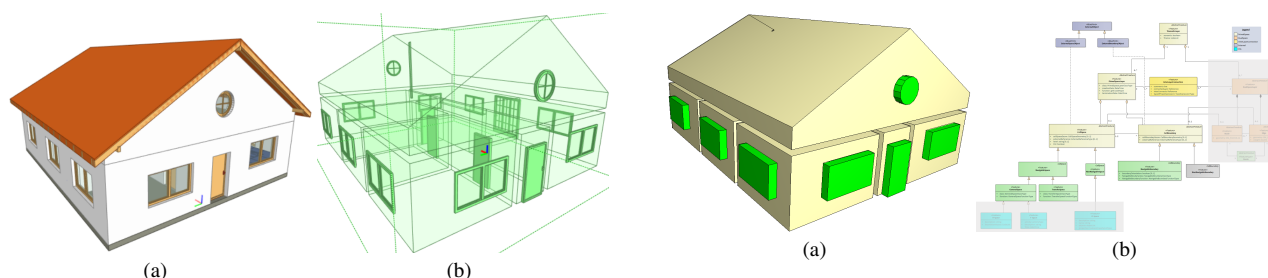


Figure 6. Input IFC model. (a) The full BIM model. (b) The spaces and opening of the model.

of IndoorGML and its associated semantic (using the Navigation module). This approach is already quite well investigated in the indoor spatial science community (Kim et al., 2014, Teo, Yu, 2017, Diakit  et al., 2017), and is becoming more and more reliable for several applications. This scenario can be useful when only the geometry of the spaces and openings are needed from a 3D building model.

Figure 7(a) shows the captured cell spaces and illustrates in (b) the parts of the UML that are solicited in such type of scenario. Typically, every *IfcSpace* entity coming from an IFC model is a direct correspondence to a *CellSpace* in IndoorGML. The 3D geometry can be directly stored in the *cellSpaceGeom* attribute, and additionally, *CellBoundary* entities can also be directly stored if available. This can be the case if the Second Level space boundaries are provided directly in the IFC model (Bazjanac, 2010, Lilis et al., 2017), which provides a subdivision of the boundaries of *IfcSpace* entities in a way that reflects any common surface with adjacent spaces. But this is not common practice, except in specialized areas such as building performance simulation.

The association with the semantic information from the navigation module is also straightforward, as any *IfcSpace* can be associated with a *GeneralSpace* instance, while the opening (*IfcOpeningElement*) of the doors (*IfcDoor*) and windows (*IfcWindow*) can be stored as *TransferSpace* instances.

## 4.2 Topology only

The proposed UML diagram for IndoorGML2.0 allows the storage of the topological information only (i.e. graph built based

Figure 7. Importing IFC geometric data into IndoorGML. (a) The *IfcSpace* entities are stored as *CellSpace* or specialized as *GeneralSpace* instances, and openings are stored as *TransferSpace*. (b) Part of the UML that is required for storing geometry and semantic of cell spaces.

on adjacency relationship of cells). In a scenario where the geometry is not needed, this could be a convenient option to use, guaranteeing lightweight files. Figure 8 shows an example of rooms and openings adjacency network computed from the input BIM model. The *Node* entities are computed using the centroid of the *IfcSpace* elements, while the *Edge* elements are obtained by connecting the nodes of adjacent spaces. Similarly to space boundaries, the information of the connectivity between the *Ifc* entities may be directly available in the model (e.g. using *IfcRelSpaceBoundary* relations).

## 4.3 Topology and Semantic

Another interesting case is the scenario where only topology and semantic are needed. This can happen for example if one wants to share a navigation networks with specified PoIs without necessarily sharing the geometry of the rooms (e.g. for file size reasons). This can be effectively implemented in the proposed IndoorGML2.0 model thanks to the links between the primal and the dual spaces along with the definition of the geometry as a optional attribute. Indeed, despite the geometry and semantic information being solely associated to *CellSpace* instances, these latter do not have to carry geometric information. This choice is motivated by the fact that any dual space is created from a primal space at the first place. Therefore, the user has the option to pick what is needed from the primal space once the network is built. As a consequence, this implies that any geometric information needed for the nodes and edges should

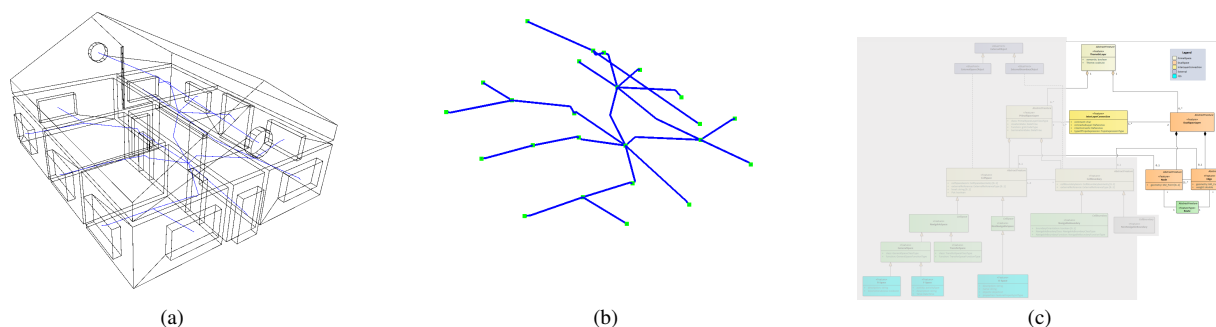


Figure 8. Storage of topological information only. (a) Ifc entities and their computed adjacency graph. (b) Network to store in IndoorGML. (c) Part of the UML required for topology (network) storage.

be stored in the *DualSpaceLayer* instance, as their will be no mean to compute them from a *PrimalSpaceLayer* where no 3D geometry is defined.

## 5. CONCLUSION

In this paper, we introduced several changes and improvements that are being considered and implemented in the IndoorGML2.0 standard proposal. In general, the previous version of the standard is simplified and number of inconsistencies are removed, some classes are renamed for the sake of clarity, some classes are now represented as attributes, and these latter and their related code lists are specified. Furthermore, new advanced space subdivision concepts are also introduced, enabling support for essential LBS components such as PoIs. The scope of IndoorGML2.0 is thereby expected to incorporate FSS which, in combination with the Multi-Layer concept, makes it possible to refine the granularity of the indoor spaces and allow fine-grained indoor LBS, while respecting all the topological constraints defined by the standard (e.g. no two cell space shall overlap). Those concepts has been presented and illustrated and several use cases has been discussed as well.

As a work in development, many other aspects will need further work in the future. One of them could be the investigation of more relevant themes for the newly introduced *ThematicLayer* class, for which a legal or property-related themes, such as defined in standard like LADM, could be good candidates. The introduction of the FSS framework to the standard is also relatively new, so all the related attributes need to be further investigated and tested against real world cases to be validated. Related to this is the concept of PoI, which needs further testing and perhaps amendments. Last but not least, one of the main mission of IndoorGML2.0 is to be easily implementable and widely supported by open source tools and software packages and libraries. To this end, on of the main goal in the standard will be a clear guideline in deriving GML, SQL and JSON implementation of the defined UMLs, following a MDA (model-driven approach). With respect to this, initial experiments with MDA to derive technical SQL implementation has been presented in (Alattas et al., 2018a). All this together, in addition to the expected future contributions of the whole indoor spatial information community, will make IndoorGML a strong and reliable open standard to support any LBS-related application.

## REFERENCES

- Alattas, A., van Oosterom, P., Zlatanova, S., 2018a. Deriving the technical model for the indoor navigation prototype based on the integration of indoorGML and LADM conceptual model.
- 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. *GIScience 2018: 10th International Conference on Geographic Information Science*, 114, Dagstuhl.
- Bazjanac, V., 2010. Space boundary requirements for modeling of building geometry for energy and other performance simulation. *CIB W78: 27th International Conference*.
- Becker, T., Nagel, C., Kolbe, T. H., 2009. A multilayered space-event model for navigation in indoor spaces. *3D geo-information sciences*, Springer, 61–77.
- Claridades, A. R., Park, I., Lee, J., 2019. Integrating IndoorGML and Indoor POI Data for Navigation Applications in Indoor Space. *Journal of Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, 37(5), 359–366.
- Diakit , A. A., Zlatanov, S., Li, K.-J., 2017. About The Subdivision of Indoor Spaces in IndoorGML. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, IV-4/W5, 4.
- Diakit , A. A., Zlatanova, S., 2016. Extraction of the 3D Free Space from Building Models for Indoor Navigation. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, IV-2/W1, 4.
- Diakit , A. A., Zlatanova, S., 2018. Spatial subdivision of complex indoor environments for 3D indoor navigation. *International Journal of Geographical Information Science*, 32(2), 213–235.
- Flikweert, P., Peters, R., D  az-Vilarino, L., Vo  te, R., Staats, B., 2019. Automatic extraction of a navigation graph intended for IndoorGML from an indoor point cloud. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W5, 5, 271–278.



- Hwang, J.-R., Kang, H.-Y., Choi, J.-w., 2012. Development of an editor and a viewer for indoorgml. *Proceedings of the Fourth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, 37–40.
- IndoorGML, O., 2014. OGC Standard for Indoor Spatial Information. Internet: <http://www.indoorgml.net/>, [May 03, 2020].
- Jung, H., Lee, J., 2015. Indoor subspacing to implement IndoorGML for indoor navigation. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40, 25.
- Kang, H.-K., Li, K.-J., 2017. A standard indoor spatial data modelOGC IndoorGML and implementation approaches. *ISPRS International Journal of Geo-Information*, 6(4), 116.
- Kim, J.-S., Yoo, S.-J., Li, K.-J., 2014. Integrating indoorgml and citygml for indoor space. *International Symposium on Web and Wireless Geographical Information Systems*, Springer, 184–196.
- Li, K.-J., 2016. IndoorGML—A Standard for Indoor Spatial Modeling. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 41, 701.
- 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.
- Li, K. J., Lee, J. Y., 2013. Basic concepts of indoor spatial information candidate standard IndoorGML and its applications. *Spatial Information Research*, 21(3), 1–10.
- Lilis, G. N., Giannakis, G. I., Rovas, D. V., 2017. Automatic generation of second-level space boundary topology from IFC geometry inputs. *Automation in Construction*, 76, 108–124.
- 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*, IV-2/W4, 4, 343.
- Mirvahabi, S., Abbaspour, R. A., 2015. Automatic extraction of IndoorGML core model from OpenStreetMap. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(1), 459.
- Park, J., Kang, H.-Y., Lee, J., 2016. A spatial-temporal POI data model for implementing location-based services. *Journal of Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, 34(6), 609–618.
- Ryoo, H.-G., Kim, T., Li, K.-J., 2015. Comparison between two ogc standards for indoor space: Citygml and indoorgml. *Proceedings of the Seventh ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, 1–8.
- Salveti, M., Li, K.-J., Conti, G., 2015. The indoorgml plug-in for josm. *Proceedings of the Seventh ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, 1–3.
- Teo, T.-A., Yu, S.-C., 2017. The Extraction of Indoor Building Information from BIM To OGC IndoorGML. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 42.
- Zlatanova, S., 2000. On 3d topological relationships. *Proceedings 11th international workshop on database and expert systems applications*, IEEE, 913–919.
- Zlatanova, S., Liu, L., Sithole, G., 2013. A conceptual framework of space subdivision for indoor navigation. *Proceedings of the fifth ACM SIGSPATIAL international workshop on indoor spatial awareness*, 37–41.