**Open Geospatial Consortium**

Submission Date: <2023-01-24>

Approval Date:   <yyyy-mm-dd>

Publication Date:   <yyyy-mm-dd>

External identifier of this OGC® document: <<http://www.opengis.net/doc/is/IndoorGML/>2.0>

Internal reference number of this OGC® document:   OGC 22-045

Version: 1.0

Category: OGC® Standard

Editor:   Sisi Zlatanova, Abdoulaye Diakité, and Ki-Joune Li

OGC IndoorGML 2.0 Part I – Conceptual Model

**Copyright notice**

Copyright © 2022 Open Geospatial Consortium  
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

**Warning**

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

Document type:    OGC® Standard

Document subtype:    if applicable

Document stage:    Draft

Document language:  English

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER’S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR’s sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

Contents

[II. KEYWORDS 6](#_Toc146459149)

[III. PREFACE 6](#_Toc146459150)

[IV. SECURITY CONSIDERATIONS 7](#_Toc146459151)

[V. SUBMITTING ORGANIZATIONS 7](#_Toc146459152)

[VI. SUBMISSION CONTACT POINTS 7](#_Toc146459153)

[1 Scope 8](#_Toc146459154)

[2 Conformance 8](#_Toc146459155)

[3 References 8](#_Toc146459156)

[4 Terms and Definitions 9](#_Toc146459157)

[4.1 Indoor Space 9](#_Toc146459158)

[4.2 Cellular Space 9](#_Toc146459159)

[4.3 Graph 9](#_Toc146459160)

[4.4 Adjacency Graph 10](#_Toc146459161)

[4.5 Connectivity Graph 10](#_Toc146459162)

[4.6 Logical Network 10](#_Toc146459163)

[4.7 Geometric Network 10](#_Toc146459164)

[4.8 Multi-Layered Space Model 10](#_Toc146459165)

[5 Conventions 10](#_Toc146459166)

[5.2. UML Notation 12](#_Toc146459167)

[5.4 Identifiers 13](#_Toc146459168)

[6 OVERVIEW OF INDOORGML 13](#_Toc146459169)

[6.1 Motivation for defining IndoorGML 13](#_Toc146459170)

[6.2 Modularisation 14](#_Toc146459171)

[7 GENERAL CONCEPTS OF INDOORGML 15](#_Toc146459172)

[7.1 Space 15](#_Toc146459173)

[7.2 Cellular space 16](#_Toc146459174)

[7.2.1 Geometry 17](#_Toc146459175)

[7.2.2 Topology 18](#_Toc146459176)

[7.2.3 Subdivision, aggregation, and selection 19](#_Toc146459177)

[7.3 Poincaré Duality 20](#_Toc146459178)

[7.4 Structured space model 22](#_Toc146459179)

[7.5 Semantics 23](#_Toc146459180)

[7.6 Thematic layers 23](#_Toc146459181)

[7.6.1 Multiple-Layered Space representation 25](#_Toc146459182)

[7.6.2 Inter-Layer Relations 25](#_Toc146459183)

[7.7 Levels of Detail (LoD) 🡪 to be included in IndoorGML 2.0.1 (or 2.1) 28](#_Toc146459184)

[8 Data model 28](#_Toc146459185)

[8.4 IndoorGML Core Module 29](#_Toc146459186)

[8.4.1 CellSpace 30](#_Toc146459187)

[8.4.2 CellBoundary 31](#_Toc146459188)

[8.4.3 PrimalSpaceLayer 33](#_Toc146459189)

[8.4.4 Node 34](#_Toc146459190)

[8.4.5 Edge 35](#_Toc146459191)

[8.4.6 DualSpaceLayer 36](#_Toc146459192)

[8.4.7 InterLayerConnection 37](#_Toc146459193)

[8.4.8 ThematicLayer 38](#_Toc146459194)

[8.5 Navigation extension module 39](#_Toc146459195)

[8.5.9 NavigableSpace 40](#_Toc146459196)

[8.5.1 GeneralSpace 41](#_Toc146459197)

[8.5.2 TransferSpace 41](#_Toc146459198)

[8.5.3 NavigableBoundary 42](#_Toc146459199)

[8.5.4 NonNavigableSpace 42](#_Toc146459200)

[8.5.5 ObjectSpace 43](#_Toc146459201)

[8.5.6 NonNavigableBoundary 43](#_Toc146459202)

[8.5.7 Route 44](#_Toc146459203)

[9 Data dictionary and requirements 44](#_Toc146459204)

[9.1 Feature Types in Core Module 45](#_Toc146459205)

[9.1.1 IndoorFeatures 45](#_Toc146459206)

[9.1.2 ThematicLayer 45](#_Toc146459207)

[9.1.3 PrimalSpaceLayer 46](#_Toc146459208)

[9.1.4 CellSpace 47](#_Toc146459209)

[9.1.5 CellBoundary 48](#_Toc146459210)

[9.1.6 DualSpaceLayer 49](#_Toc146459211)

[9.1.7 Node 50](#_Toc146459212)

[9.1.8 Edge 50](#_Toc146459213)

[9.1.9 InterLayerConnection 51](#_Toc146459214)

[9.2 Feature Types in Navigation Module 51](#_Toc146459215)

[9.2.1 NavigableSpace 51](#_Toc146459216)

[9.2.2 NonNavigableSpace 52](#_Toc146459217)

[9.2.3 GeneralSpace 52](#_Toc146459218)

[9.2.4 TransferSpace 52](#_Toc146459219)

[9.2.5 ObjectSpace 53](#_Toc146459220)

[9.2.6 NavigableBoundary 53](#_Toc146459221)

[9.2.7 NonNavigableBoundary 54](#_Toc146459222)

[9.2.8 Route 54](#_Toc146459223)

I. ABSTRACT

This OGC® IndoorGML standard specifies an open data model for indoor information in UML and technical implementation schemas in GML, SQL and JSON. While there are several 3D building modelling standards such as CityGML, KML, IFC, LADM and IMDF which deal with interiors of buildings from geometric, cartographic, and semantic viewpoints, IndoorGML focuses on modeling indoor spaces and their neighbourhood relationships to support indoor location-based services. This version of IndoorGML addresses spaces and networks for indoor navigation.

# II. KEYWORDS

The following keywords are to be used by search engines and document catalogues.

ogcdoc, ogc documents, indoor, navigation, indoorgml, gml, sql, json

# III. PREFACE

The goal of IndoorGML is to represent and allow for exchange of geoinformation that is required to build and operate systems that rely on spaces and topological relationships between them such as path computation, sensor coverage, property accessibility, etc. Several standards such as CityGML (OGC, 2012), KML (OGC, 2015), LADM (ISO, 2012) and IFC (ISO,2018) have been published to describe 3D geometry and semantics of building features, but they are not readily appropriate to derive spaces and their topological relationships. The navigation standard IMDF (OGC, 2021) provide a comprehensive model to compute path between features located on a map, but the derived network is application specific. This standard aims to provide unified, standardised and flexible approach for indoor spatial information required for space-graph based applications such as indoor navigation.

This version of the OGC standard consists of two components: 1) a core data model to describe topological connectivity and different contexts of indoor space, and 2) a data model for navigation in indoor space.

This version of IndoorGML covers geometric and semantic properties of indoor spaces relevant for indoor navigation. These indoor spaces may differ from the spaces described by other standards such as CityGML, KML, IFC, LADM and IMDF. In this respect, IndoorGML is a complementary standard to CityGML, KML, IFC, LADM and IMDF to support location-based services for indoor navigation.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.

*Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.*

# IV. SECURITY CONSIDERATIONS

No security considerations have been made for this standard.

# V. SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

* The University of New South Wales
* Pusan National University
* Ordnance Survey
* University of Seoul
* All4Land
* Delft University of Technology

The initial concepts have been developed with the support of

* Technical University of Berlin
* Technical University of Munich

# VI. SUBMISSION CONTACT POINTS

Questions regarding this submission should be directed to the editor or the submitters:

|  |  |  |
| --- | --- | --- |
| **Name** | **Affiliation** | **Contact** |
| Sisi Zlatanova | University of New South Wales | s.zlatanova at unsw.edu.au |
| Ki-Joune Li | Pusan National University | lik at pnu.edu |
| Abdoulaye Diakite | The University of New South Wales | diakite.abdoulaye at gmail.com |
| Jeremy Morley | Ordnance Survey | Jeremy.Morley at os.uk |

# Scope

IndoorGML is an OGC**®** standard for the representation and exchange of indoor navigation network models. IndoorGML is a UML conceptual schema and implementation schema of the Geography Markup Language version 3.2.1.

This version of IndoorGML establishes a common schema for indoor navigation applications. It models topology and semantics of indoor spaces, which are needed for the components of navigation networks. IndoorGML contains a minimum set of generic, unified modelling concepts for indoor environments as follows:

* Spaces and space subdivision contexts;
* Geometric and semantic properties of spaces;
* Types of connectivity between spaces;
* Navigation networks (logical and metric) and their relationships.

# Conformance

Conformance targets of this OGC® Standard are IndoorGML instance documents. Conformance with this standard shall be checked whether IndoorGML instance documents achieve the criteria as defined in clause 7 to 9.

In order to conform to IndoorGML, and schema document should:

a) conform to the rules, specifications, and requirements in clauses 7 to 9; and

b) pass all relevant test cases of the abstract test suite given in Annex A.

The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site1.

# References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

ISO: ISO 8601:2004, Data elements and interchange formats – Information interchange – Representation of dates and times, 2004. ISO (2004).

ISO/TS: ISO/TS 19103:2005, Geographic Information – Conceptual Schema Language, 2005. ISO and TS (2005).

ISO: ISO 19105:2000, Geographic information – Conformance and testing, 2000. ISO (2000).

ISO: ISO 19107:2003, Geographic Information – Spatial Schema, 2003. ISO (2003).

ISO: ISO 19109:2005, Geographic Information – Rules for Application Schemas, 2005. ISO (2005).

ISO: ISO 19111:2003, Geographic information – Spatial referencing by coordinates, 2003. ISO (2003).

ISO: ISO 19115-1:2014, Geographic Information – Metadata – Part 1: Fundamentals, 2014. ISO (2014).

ISO/TS: ISO/TS 19139:2007, Geographic Information – Metadata – XML schema implementation, 2007. ISO and TS (2007).

OGC 08-126, OGC® Abstract Specification Topic 5, The OpenGIS Feature, 2009

OGC 99-108r2, OGC® Abstract Specification Topic 8, Relations between Features, 1999

OGC 99-110, OGC® Abstract Specification Topic 10, Feature Collections, 1999

OGC 07-036, OGC® Geography Markup Language Implementation Specification, Version 3.2.1, 2007

# Terms and Definitions

This document used the terms defined in Policy Directive 49[[1]](#footnote-1), which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this Standard and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

For the purposes of this document, the following additional terms and definitions apply.

## Indoor Space

A space within one or multiple buildings.

## Cellular Space

A cellular space *S* is a set of identifiable sells *ci* grouped according to theme *T* and is noted *ST* is *ST* = {*c*1, *c*2, …, *cn*}

## Graph

A graph *G* (*V*, *E*) where *V* is a set of nodes representing cells and *E* is the set of edges indicating topological relationships.

## Adjacency Graph

A graph *Gadj* (*V*, *Eadj*) where *V* is a set of nodes representing cells and *Eadj* is the set of edges indicating the adjacency relationship.

## Connectivity Graph

A graph Gcon(*V*, *E*con) where *V* is a set of nodes representing cells in indoor space and *Econ* is the set of edges indicating the connectivity relationship.

## Logical Network

A graph *G* (*V*, *E*), where node *v* in *V* and edge *e* in *E* do not contain any geometric properties.

## Geometric Network

A graph *G* (*V*, *E*) where node *v* in *V* and edge *e* in *E* contain both geometric properties.

## Multi-Layered Space Model

A model representing multiple themes of cellular spaces and/or graphs and inter-layer connections between them.

# Conventions

**5.1. Symbols (and abbreviated terms)**

The following symbols and abbreviated are used in this standard.

|  |  |
| --- | --- |
| **Abbreviation** | **Word or Phrase** |
| BIM | Building Information Modeling |
| CityGML | City Geographic Markup Language |
| GPS | Global Positioning Systems |
| CRS | Coordinate Reference System |
| GML | Geographic Markup Language |
| IndoorGML | Indoor Geographic Markup Language |
| IFC | Industry Foundation Classes |
| ISO | International Organization for Standardization |
| KML | Keyhole Markup Language |
| LOD | Level of Detail |
| MLSM | Multi-Layered Space Model |
| OGC | Open Geospatial Consortium |
| RFID | Radio Frequency IDentifier |
| UML | Unified Modeling Language |
| XML | eXtended Markup Language |
| 1D | One Dimensional |
| 2D | Two Dimensional |
| 3D | Three Dimensional |

## 5.2. UML Notation

The diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram. The UML notations used in this standard are described in the diagram below.



Figure 1: UML Notations

In this standard, the following three stereotypes of UML classes are used.

* <<Interface>> A definition of a set of operations that is supported by objects having this interface. An Interface class cannot contain any attributes.
* <<DataType>> A descriptor of a set of values that lack identity (independent existence and the possibility of side effects). A DataType is a class with no operations whose primary purpose is to hold the information.
* <<CodeList>> is a flexible enumeration that uses string values for expressing a list of potential values.

In this standard, the following standard data types are used:

* CharacterString – A sequence of characters;
* Boolean – A binary value of either 1 (true) or 0 (false);
* Integer – An integer number;
* Double – A double precision floating point number; and
* Float – A single precision floating point number.

## Identifiers

The normative provisions in this specification are denoted by the URI

http://www.opengis.net/spec/{standard}/{m.n}

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

# OVERVIEW OF INDOORGML

IndoorGML has been designed to support applications developers in providing Location-based services applications. Figure 2 illustrates the place of IndoorGML in the ecosystem of standards, models and files formats and end-user applications. IndoorGML provides simplified yet standardised notations for indoor spaces and networks, which can be used in different application contexts such as navigation, monitoring, asset and property management. IndoorGML can be linked to and derived from any geometric model that a building owner may have (floor plans, CAD models, BIM models, laser scans, measurements). The semantics notations of IndoorGML are generic and therefore allowing to protect some sensitive building information.

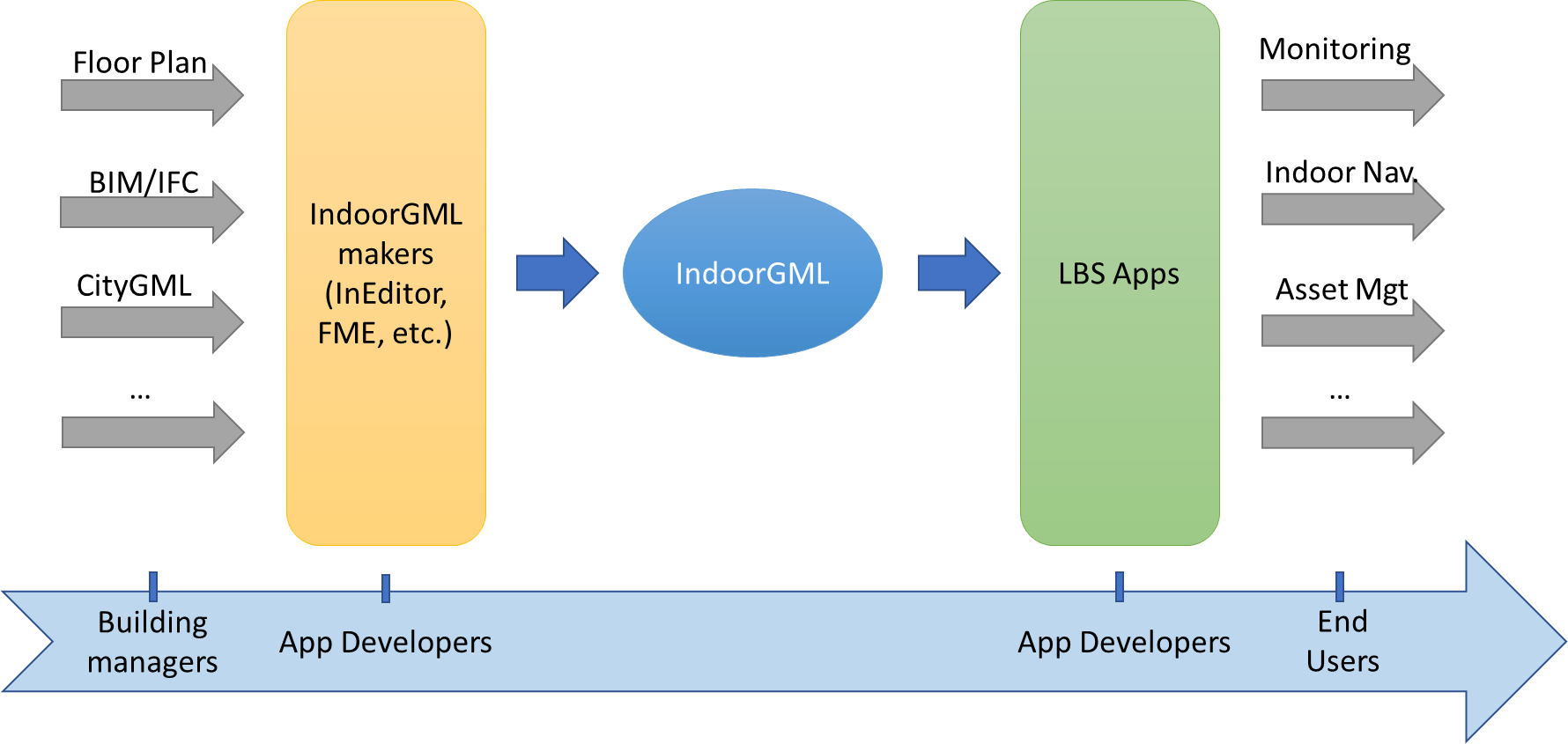


Figure 2: IndoorGML in the overall application development ecosystem

## Motivation for defining IndoorGML

Indoor environments differ from outdoor in many aspects. The indoor spaces have less structures lanes and directions to move; they are multi-levelled and reachable via different vertical connectors such as stairs, elevators, escalators, and ramps; they have large number of obstacles such as furniture columns, fences, decorations. The spaces are enclosed and accessible via different types of openings (normal doors, emergency doors, sliding doors, one-way doors, portals). The height of the indoor spaces might vary to such extend that some spaces become not accessible for certain type of users. This has led to the existence of variety of approaches for modelling indoor environments and providing services. Therefore, well-known concepts, data models, and standards need to the be refined and unified to reflect specifics of indoor environments.

In general, indoor spatial information can be classified into two large categories as follows:

* Architectural components (walls, stairs, slabs) and interior facilities (furniture).
* Cavities (rooms and corridors) or virtual subdivision (sensors coverage and legal spaces)

Building and facility management application require mostly information from the first category. Indoor location-based services (LBS), indoor route analysis or indoor geo-tagging services require mostly information from the second category.

IndoorGML is intended provide a unified modelling approach that is necessary to support indoor applications using information from those two categories. The leading concepts in IndoorGML are the Indoor spaces and the topological relationships between them (Section 7.1), which are grounded in the Poincaré duality. The space notations are kept as generic as possible to reflect the variety and complexity of indoor environments. The entire indoor environment - objects and spaces - constitutes the Cellular space (Section 7.2). Cells have attributes, one of which is their geometry. The cell units can be subdivided or aggregated (Section 7.2). The Cell Spaces are the basis for deriving an adjacency/connectivity/accessibility network (Section 7.3). Cell Spaces of the same characteristics are non-overlapping and form a thematic layer (Section 7.6). For example, architectural components (walls, slabs, stairs) and the corresponding cavities (rooms, corridors) form a Topographic thematic layer.

IndoorGML 2.0 follows a model-driven approach. All concepts are organised in a UML class diagram (Section 8), from which the implementation schemas for GMLis provided (Annex A).

## Modularisation

Following the guidance in the OGC’s policy (OGC, The Specification Model – A Standard for Modular specifications, 2009)￼, IndoorGML is organised into a Core module and Extension modules that have mandatory dependency on the core (see Figure 3). ￼Figure 1The IndoorGML core module comprises the basic concept and each extension module covers a specific application, which requires extension of the core module semantics. IndoorGML 2.0 contains one extension named *Navigation*. Each IndoorGML module is specified by an implementation schema definition (XML, SQL and JSON).

The dependency relationships among IndoorGML’s modules are illustrated in Figure 3. Each module is represented by a package in UML. The package name corresponds to the module name. A dash arrow in the figure indicates that the schema at the tail of the arrow depends upon the schema at the head of the arrow. In the following sections the modules are described in detail.

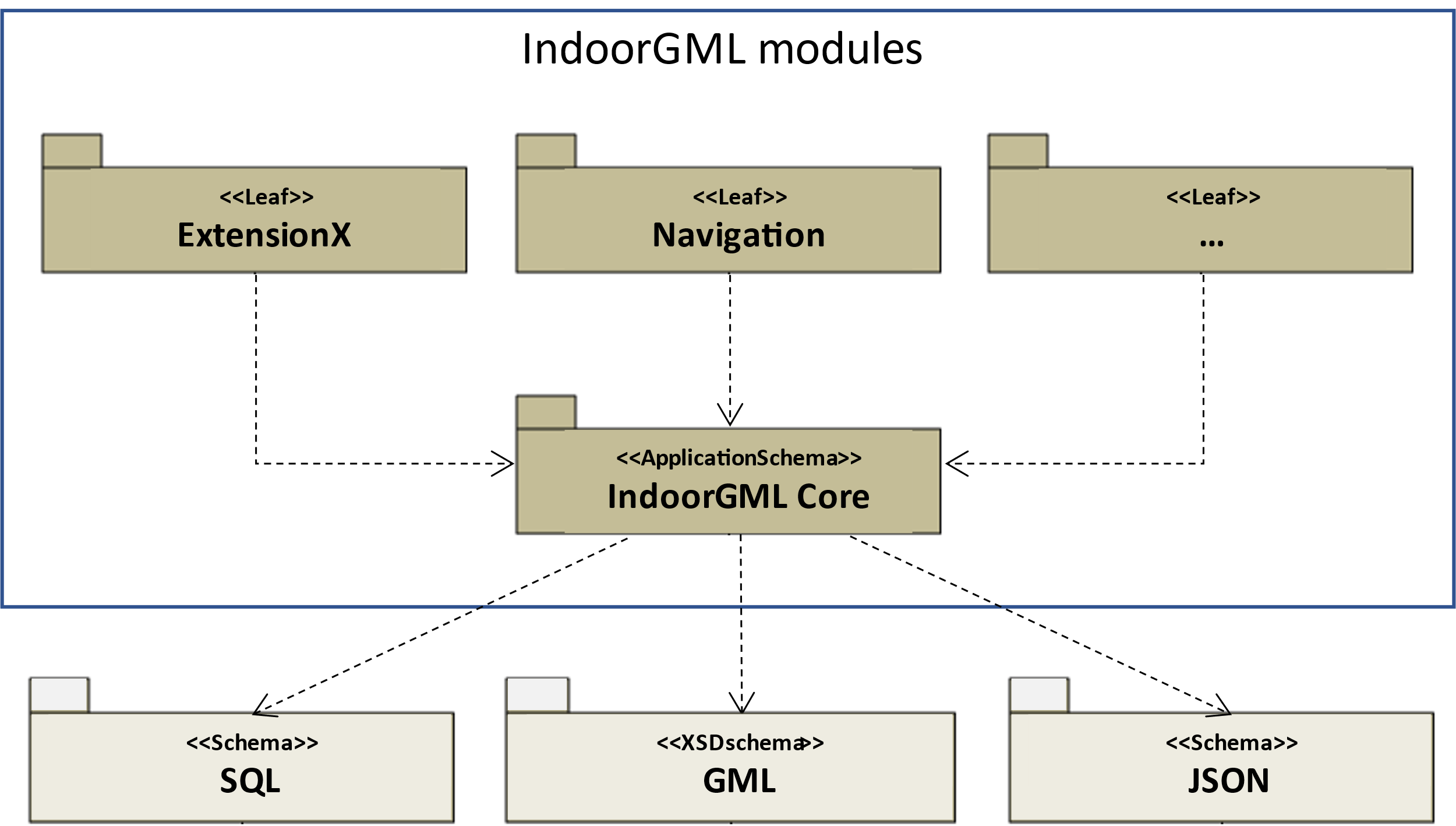


Figure 3: Modular organisation of IndoorGML

# GENERAL CONCEPTS OF INDOORGML

IndoorGML is a space-centred standard. As so, it focuses on the three main types of information of spaces (2D or 3D): geometry, topology and semantic. In order to define the space and its suitable properties under the consideration of those three types of information, the standard relies on the following concepts:

* Cellular space,
* Poincaré Duality,
* Semantic extension,
* Thematic layering.

The cellular space provides the geometric description of an IndoorGML model. The Poincaré duality describes the topological relations such as adjacency and connectivity between the spaces. Together, they form the key concept of Primal-Dual model that defines the core part of an IndoorGML model. The semantic extension mechanism, as its name suggests, allows to add more details to the basic semantics of the core module. Thematic layering mechanism allows to organise an IndoorGML model as a collection of layers with different themes. Those concepts are elaborated in the following subsections.

## Space

The notion of space is widely explored in spatial science and urban applications in general (Zlatanova, et al., 2020). Among its diverse definitions that can be found in dictionaries and related literature, one definition of the space encapsulates most of the concepts attached it:

*“Space is either unlimited expand or an empty area usually bounded in some way between things (e.g., the architect left space in front of the building) or an area reserved for some particular purpose (e.g., the laboratory’s floor space)”* (Princeton University, 2010).

That definition acknowledges three main aspects of the space: (a) its ability to expand infinitely, (b) its intuition to be generally empty and eventually bounded (particularly in the built environment) and (c) its functional property. In IndoorGML, the space is characterized by all those properties, except IndoorGML space is not necessarily empty. Depending on the IndoorGML extension (indoor navigation, sensors coverage, ownership, etc.) spaces can be empty, non-empty or partially empty.

The indoor space is commonly perceived as a space within a building. It incorporates architectural components such as walls, slabs, doors, etc, furniture such as chairs, tables, desks and the remaining empty spaces as in rooms, corridors, halls, etc. IndoorGML 2.0 focusses on the empty spaces where objects can be located, and activities can be hosted for indoor navigation or LBS. Consequently, the relationships between spaces are of critical importance.

Spaces in the built environment are not always sharply distinguishable. Many spaces cannot be strictly categorised as indoor or outdoor, but rather as semi-spaces often linking indoor and outdoor environments (Yan, Diakité, & Zlatanova, 2019; Zlatanova, et al., 2020). For example, an inner court, a veranda, a balcony, or an open bridge can belong to a building, without being entirely enclosed within the shell of the building. Nevertheless, for a matter of completeness, IndoorGML can account for all types of space within the built environment, as long as they can be represented with the IndoorGML Cellular space concept.

## Cellular space

A *cellular space* is a set of *cells* (or *CellSpaces*) defined as the smallest organizational or structural unit (Princeton University, 2010) and grouped according to thematic criteria (e.g. topographic space, sensor coverage space, etc.). A cellular space *S* of thematic layer *T*, noted *ST* is defined as follows:

*ST* = {*c*1, *c*2, …, *cn*}

where *ci* is *i*th cell. Every cell in a cellular space can have the following properties:

* a unique identifier ;
* a name (e.g. a room number);
* a geometry (e.g. solids in 3D or surfaces in 2D)

Figure 4 illustrates a cellular space consisting of cells, which represent rooms, corridors, doors and windows in a building.

 Diagram, engineering drawing

Description automatically generated Diagram, engineering drawing

Description automatically generated

1. b) c)

Figure 4: A building (a) and the corresponding cellural space, containing all empty cells (b) and corresponding cells of a room, a corridor and their openings (c).

Within a cellular space, only the adjacency relationship is allowed between cells, that is, no overlap may occur. Overlapping cells must be organised in a new cellular space. A cellular space may be incomplete coverage, i.e. it is possible to have gaps between cells (Figure 5).

Diagram

Description automatically generated

Figure 5: Cellular space containg disconected cells, i.e. all offices in an university building (Allatas et al 2017)

In IndoorGML, a cellular space can be subdivided into smaller cells or aggregated into larger ones. Those operations detailed in section 7.2.3 allow for both a tailored geometric granularity and functional specification of spaces.

### Geometry

Every cell defining an item in indoor space owns a shape, size and location that can be collected and modelled. It can represent physical features such a room, door, wall, or virtual spaces such as legal rights and access or sensors coverages. Depending on the application, the geometry of a cell can be simplified and generalised into a minmax bounding box. Such approach can be applied when considering highly irregular shapes like furniture. Geometric information can be included in IndoorGML either directly or via an external link. Geometry of cells can be omitted as well.

Geometry of cells is defined in Euclidean space and provides the means for the quantitative description of the spatial characteristics of cell. Metrics is defined as in (Morris, 2019). IndoorGML cells are modelled as features and follow ISO19107 (Spatial Schema) (ISO, 2019) provides conceptual schemas to describe and model real world objects as features.

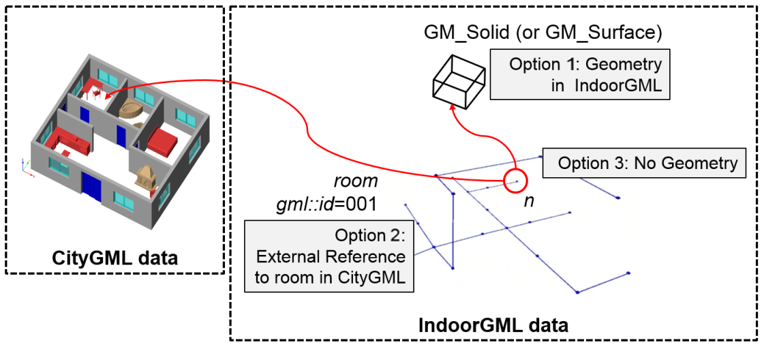


Figure 6: Three options to represent geometry in IndoorGML

As illustrated in Figure 6, there are three options for representing geometry of a cell in indoor space:

1. Geometry in IndoorGML (Option 1): geometric representation of cell may be included within an IndoorGML document. It is GM\_Solid in 3D space and GM\_Surface in 2D space as defined in ISO 19107. Note that solid with holes or surface with holes are allowed in this standard.
2. External Reference (Option 2): instead of explicit representation of geometry, an IndoorGML document can only contain external links to objects defined in other data sets such as IFC or CityGML, where the referenced objects in external data set include geometric information. Then there must be 1:1 or *n*:1 mapping from cells in IndoorGML to corresponding objects in other datasets.
3. No Geometry (Option 3): no geometric information is included in IndoorGML. This means that the shape, extend and location are unknown. The cell is defined by its identifier.

Note that Option 2 can always be used in combination with the other options. When Option 1 is used, three fundamental operations can be applied to cell spaces: subdivision, aggregation, and selection.

### Topology

Beside the geometry of a cellular space, it is also possible to represent cells in the Topological space by a topological model, representing the relationships between points, lines and polygons constructing the geometry (Gröger and George, 2012). Such topological models are dedicated to representing spatial relationships thus their shape and size is not described (Edenhofer et al 1989). This is to say, geometric predicates such as volumes, areas, distances cannot be computed. As mentioned above IndoorGML deals only with cells in 3D and 2D, which are presented as solids or polygons. Consequently, the topological model contains only 3D and 2D objects in 3D cellular space and 2D and 1D objects in 2D cellular space. More information about topology of cells can be found in Section 8.4.3

### Subdivision, aggregation, and selection

The indoor environment is complex and indoor spaces often have hierarchical structures. For several indoor applications, a careful decomposition of an indoor space is required to reflect these hierarchical structures. To support the representation of such configurations, the subdivision, aggregation, and selection processes on the CellSpaces can help achieve them.

Diagram, engineering drawing

Description automatically generated Diagram, engineering drawing

Description automatically generated  
a) b)

Subdivision

Aggregation

A picture containing text, stationary, businesscard

Description automatically generated Surface chart

Description automatically generated  
c) d)

Figure 7: (a) A furnished indoor space. (b) Subdivision of the indoor space into two separate rooms with exclusion of furnishing elements’ spaces. (c) Selection of specific CellSpaces (green) suitable for walking and rolling. (d) CellSpaces (green) suitable for flying.

As illustrated in Figure 7, the subdivision consists in splitting the original cells into several subspaces of different geometry, according to their function. For example, in Figure 7(b), the indoor space is subdivided into several. This subdivision allows to segment Figure 7(a)in subspaces of different functions, e.g. a kitchen and a living room, as well as discriminating the spaces physically occupied by items. The subdivision process could be based on any application-based criteria and all resulting subspaces are CellSpaces of a cellular space. For the purpose of navigation applications, subdivisions may be required because of:

* geometry simplification, e.g. working with spaces that have only convex shapes
* increase of granularity, e.g. in for improving the localisation of people and items.
* need to identify specific functional/perception spaces: e.g. waiting or smoking areas.
* defining free spaces, e.g. spaces free of obstacles.

The aggregation process is the reverse of the subdivision, which leads subspaces to be merged instead of being split. Therefore, the merging of all subspaces of Figure 7(a) allows to retrieve the original cell spaces of Figure 7(a). Similarly, any new cell resulting from this process is a CellSpace of a cellular space. For the purpose of indoor navigation, aggregation may be required when:

* There are CellSpaces of no interest for an application, e.g. induvial toilets or service areas in a building
* There are CellSpaces, which are not accessible for specific users, e.g. restricted areas at hospitals and airports.

Finally, the selection allows to discriminate the CellSpaces of interest from the rest. Figure 7(c) and (d) show a scenario where only CellSpaces that can support a certain type of locomotion mode are considered in the cellular space (see the green CellSpaces). The selection of spaces for indoor navigation applications can take place for many different reasons:

* to reduce the overall number of spaces, e.g. select only empty spaces, such as rooms and corridors and avoid non-empty spaces such as walls, slabs, or too crowded areas.
* to eliminate spaces, which will not be used for a specific user: e.g. select only common spaces for a visitor of a public building
* to eliminate spaces of danger: e.g. in emergency cases, select only spaces which are still safe for users to be in.

## Poincaré Duality

Topological relations between cells is crucial in IndoorGML. They allow establishing links between cell in the same or different thematic layers, which is critical information for several applications such as navigation and LBS. As mentioned above, a topological model of cellular space is partial and represents only relations between cells and their boundaries. The Poincaré duality (Munkres, 2018) is further employed to explicitly describe the relationships between the cells. It provides a theoretical background for mapping cellular space to a graph or network to represent allowed topological relationships. It simplifies the complex spatial relationships, which may occur especially in 3D by a topological model (Lee, 2004).

The Poincaré duality refers to two spaces namely *Primal Space* and *Dual Space*. A *k-*dimensional object in *N-*dimensional Primal Space is mapped to (*N-*k) dimensional object in Dual Space. Thus, solid 3D objects in 3D Primal space, e.g., rooms within a building, are mapped to nodes (0D object) in dual space. 2D surface shared by two 3D objects is transformed into an edge (1D) linking the two nodes in Dual space. The nodes and edges in Dual space form an adjacency graph. The nodes and the edges of Dual space represent abstractions of cells and their adjacency relationships in Primal space.

A picture containing text, clock

Description automatically generated Diagram

Description automatically generated

1. b)

Figure 8: Principles of Poincaré duality. 3D Primal space case (a) and 2D case (b). (Mathematical definition of Poincaré duality in (Munkres, 2018))

Figure 8 illustrates this duality transformation for the case where the primal space is 3D (a) and 2D (b) respectively. Note that the transformations from 1D object (curve) or 0D object (point) in 3D Primal space are not included in IndoorGML since they are not considered as cells in most applications. But the transformation may be applied to 1D or 0D objects of 3D primal space in a similar way if it is required. Then the adjacency graph *Gadj* is defined as follows:

*Gadj* = (*V*, *Eadj*)

where *V* and *Eadj* are sets of nodes and edges in dual space mapped from cells and surfaces in

3D primal space, respectively. The connectivity graph *Gcon* is a subset of the adjacency graph and represent only adjacency that make the spaces connected. For navigation cases connectivity between spaces (i.e. room) is provided via the notion of doors between the rooms. It is defined as:

*Gcon* = (*V*, *Econ*)

where *V* and *Econ* are sets of nodes and edges in dual space mapped from cells and surfaces in

3D primal space, respectively. Figure 9 illustrates cellular space and its connectivity graph.

Diagram, engineering drawing

Description automatically generated Chart, line chart

Description automatically generated Diagram, engineering drawing

Description automatically generated

1. b) c)

Figure 9: Poincaré duality on 3D cells of a building (a); Corresponding adjacency graph in the dual space (b); Combined primal and dual space view (c).

The adjacency graph can be represented as a *logical network* or *geometric network*. While the logical network represents only the relationships between the cells, the geometric networkholds geometry for nodes and edges.

Shape, polygon

Description automatically generated

Figure 10: Structured space model: mapping between Euclidean and Topological spaces, and Primal and Dual Spaces

## Structured space model

The Primal and Dual spaces and the Euclidean and Topological spaces are interlinked in a Structured Space Model as illustrated in Figure 10. The Primal space refers to either Euclidean or Topological space and the Dual space refers to either Geometric network or Logical network. Geometry of Cellular Space and Geometric Network are embedded in the Euclidean space, while Topology of Cellular Space and Logical Network are defined in the Topological space. IndoorGML supports the Primal and Dual models in the Euclidean space and the Logical Network in the Topological space. As mentioned above, the Geometry for Cellular space is not compulsory, as the cellular space can be identified. IndoorGML is valid with at least one of the Primal spaces. See examples in Section 9.

The Euclidean space (Geometry) is estimated to be the most useful for applications such as navigation and LBS. IndoorGML may then containing both Geometry and Geometry Network, or only Geometry, or only Geometric Network. Other types of applications, such as dealing with ownership or sensor coverage, may be better supporter by IndoorGML containing Geometry and Logical Network or Topology and Logical Network.

## Semantics

IndoorGML offers a basic semantic for the Primal and Dual spaces of the core module. The semantics of the core model is generic for all applications as it does not specify any other information about the Primal and Dual Spaces, except some characteristics such as name, level, and PoI. If no extension module is involved, the cells carry the semantics of the core module only.

Further semantic specifications are provided via the Extension modules as specified in Section 8.5. Every cell is further classified according to the semantic introduced by the extension module. IndoorGML 2.0 maintains semantics for Indoor navigation which is provided within Navigation extension module. The semantics, developed through the Navigation extension module is intended for two purposes to: 1) provide a classification of a cell, and 2) determine adjacency relationships that ensure connectivity between cells. Semantics thus allows to define cells that are important for navigation. Thus, a cell can be classified as navigable (room, corridor, hall) , non-navigable (wall, slab, furniture), opening (door, window), etc. (see Section 8.5). The subdivision and classification of Cellular space relies on the architectural layout of a building.

While this may be enough for some cases based on connectivity graph analysis, it can rapidly be limiting for more specialized applications such as sensor managements, legal aspects or security, where advanced, specific semantic needs to be associated to the geometric and topological elements. Examples can be a Legal Extension module, in which a cell might be classified as ‘ownership’, ‘restriction’, ‘responsibility’ etc. or a Security extension module that may offer semantics that would indicate ‘check-in’, ‘boarding’, ‘crew entrance’, etc.

Semantic extension mechanism allows to add more semantic on primal or dual spaces, as long as they follow the modularization principle. Cells can be organised in a hierarchical structure according to their semantics, corresponding properties, and semantic interrelations (specialisation and generalisation). For example, ‘room’ is a specialization of ‘navigable cell’ and ‘non-navigable cell’ is a generalization of ‘walls’ and ‘obstacles’. Cells created for one space representation may be aggregated or subdivided for the purpose of another one. More details about the Navigation extension module are given in Section 8.5

## Thematic layers

A single indoor environment can be organised in many kinds of cellular spaces with distinct subdivision and semantic specifications. Within each Extension module, it is possible to have many different subdivisions and each cellular space is targeted towards specific applications and needs. A cellular space with a specific semantics and/or geometric subdivision, aiming to reflect a group of application can be organised in a *Thematic Layer*. Thematic layers can be defined using the Extension modules and/or Core module. Thematic layers making use of the semantics of Core module only, can be derived applying the principles of space partitioning, i.e. subdivision, aggregation and selection. Examples of such thematic layers are subdivision according to Wi-Fi or RFID coverage (see example below). The Navigation extension module provides additional notions for navigability and connectivity. Therefore, thematic layers that rely of these properties, should include Navigation extension module. Navigation-based themes can be defined using a particular space partitioning with respect to:

* tasks: visitor, staff, facility manager, emergency responder (see Figure 11)
* user characteristics: age, gender
* mode: walking, driving, flying (see Figure 7c and Figure 7d)

IndoorGML 2.0 is organised as a collection of interconnected layers representing different themes of the same physical space. Figure 11 represents a thematic layer ‘Visitors’, which contains all cells, which are accessible to visitors in a university faculty (Alattas et al 2017). Similarly, cellular spaces can be created for students or facility management. All spaces use the semantics of the Navigation extension module, but a selection of spaces is made according to the user tasks. Similarly, cellular spaces from different extension modules can be organised in thematic layers.

Diagram

Description automatically generated

Figure 11: Cellular space for visitors (Alattas et al 2017)

In Figure 12, a physical indoor space is organized according to the Navigation extension module Navigation, named Topographic layer. In addition, two cellular subdivisions called Wi-Fi and RFID are specified, which rely on the semantics of the core model only. The Topographic layer, created under the Navigation extension module, follows the architectural layout of a building, and is composed of rooms, corridors, and stairs. Wi-Fi and RFID cells follow the outlines of the corresponding sensor coverages. The three cellular spaces, although related to distinct semantics and subdivision approaches, form each a thematic layer. These three thematic layers may be appropriate for an application that provides tracking and navigation.

Following the modularization mechanisms, every layer in IndoorGML contains the core module, which is composed of Primal space and Dual space. A valid thematic layer should contain at least one of the four space representations, i.e. Geometry, Topology, Geometric network or Logical network.

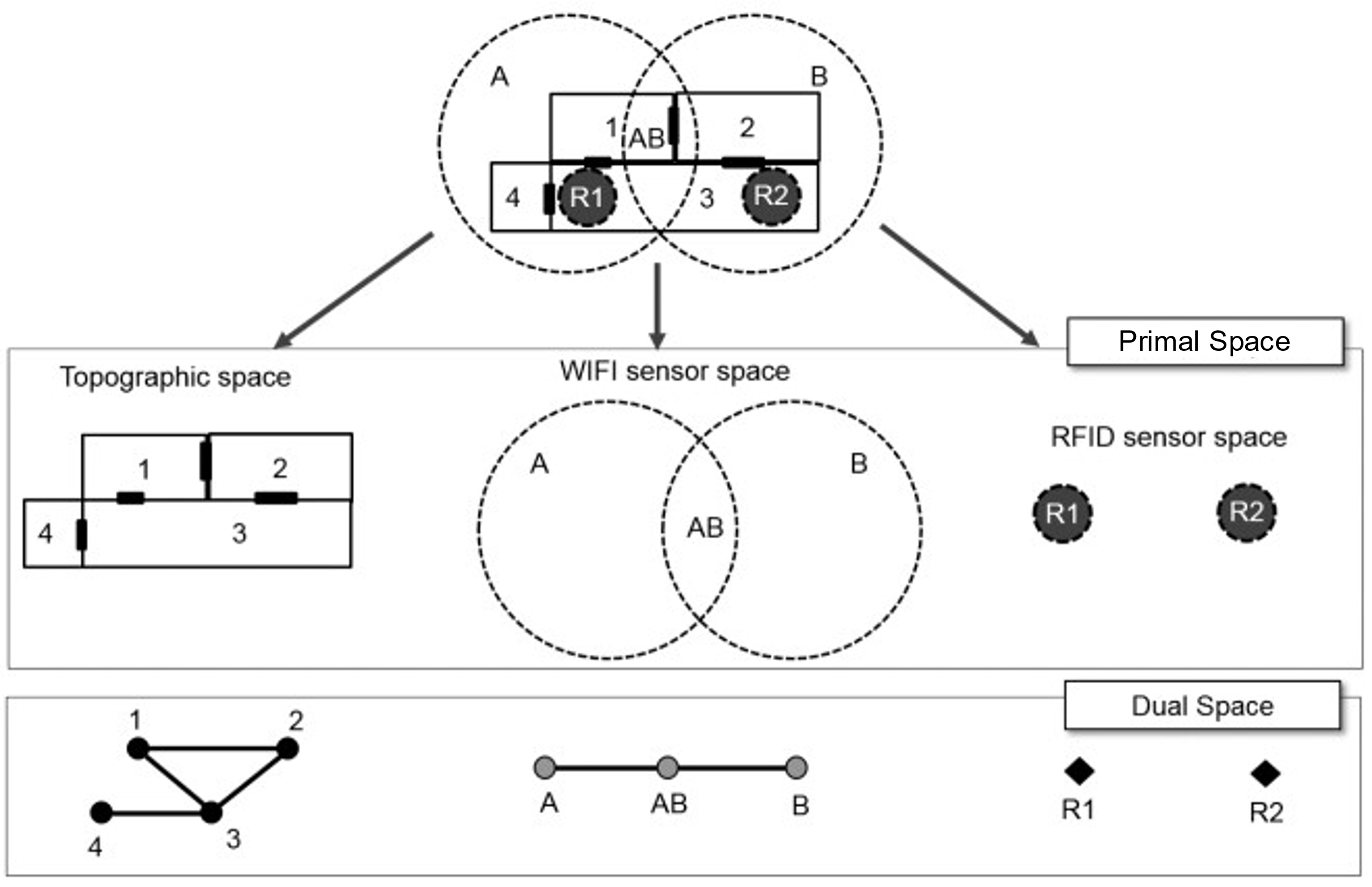


Figure 12: Three different cellural spaces for the same physical space

### Multiple-Layered Space representation

IndoorGML provides mechanisms for maintaining and linking multiple Thematic layers for a same indoor environment. Figure 13 represents the three thematic layers discussed above.

This representation method with multiple cellular space layers is called *Multiple Layered Space Representation* (MLS Representation). The MLS representation is useful for many purposes. For example, we can represent the hierarchical structure of indoor space by MLS representation, where each level is represented as a single space layer. Another application example of MLS representation is indoor tracking with presence sensors such as RFID, as shown in Figure 12. Given an indoor space represented as topographic cellular space layer and RFID sensor coverage layer respectively, we can deduce the movement of a mobile object with a RFID tag by the sequence of RFID coverage cells and corresponding inter-layer space edges.

### Inter-Layer Relations

To handle the interaction between several layers, it is necessary to represent the relationships between them. IndoorGML does this through the Inter-Layer connection which describes the spatial relationships (topology) between two layers. Unlike the topological relationships between cells of a same layer which are ruled by the Poincaré Duality (adjacency only), the inter-layer relations are ruled by the 9-intersection model (Egenhofer 1989). IndoorGML 2.0 concentrates on six relationships namely *contains, within, covers, coveredBy*, *overlaps and equals* between cells in the Primal space and nodes in Dual space and their combinations

Shape

Description automatically generated

Figure 13: Corresponding Primal and Dual spaces of different thematic layers.

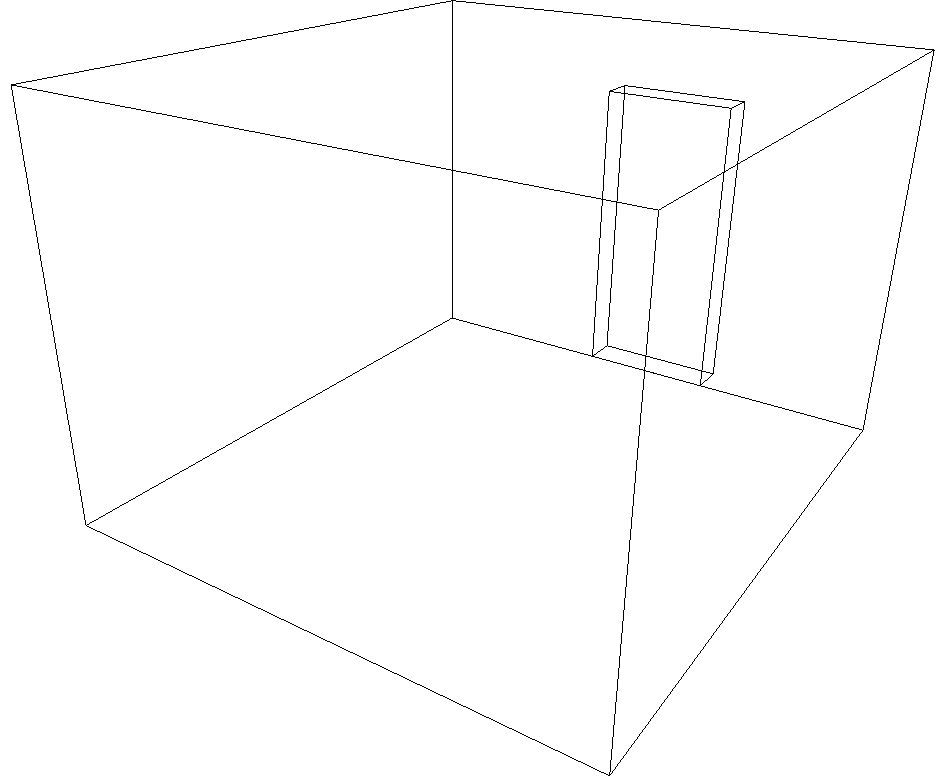
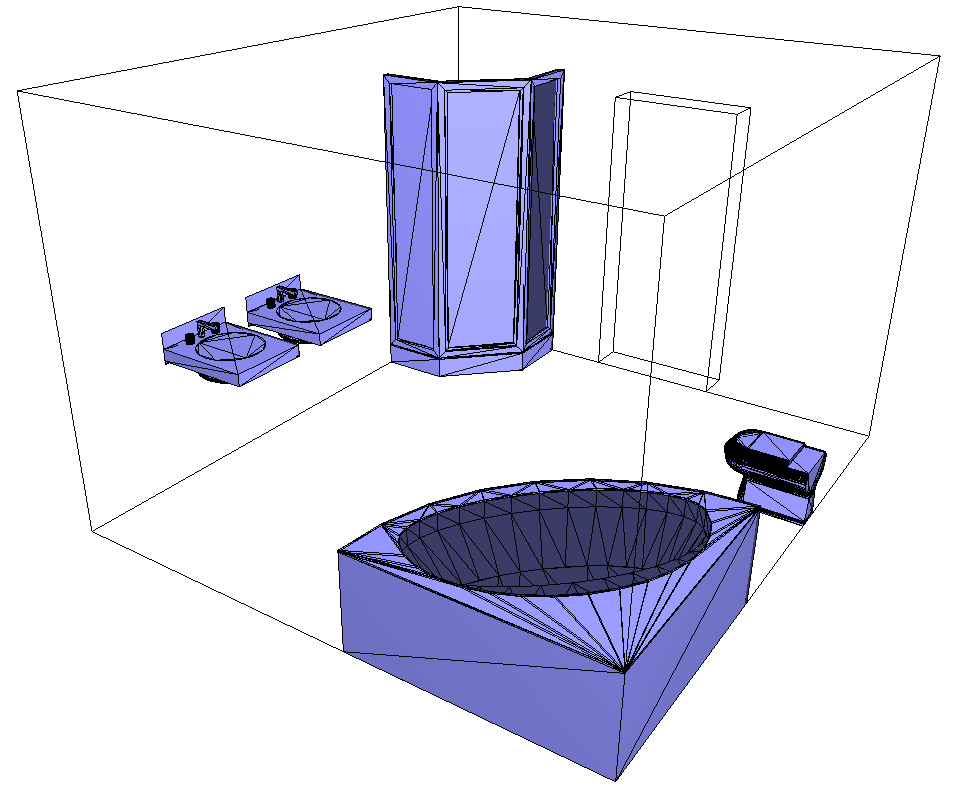
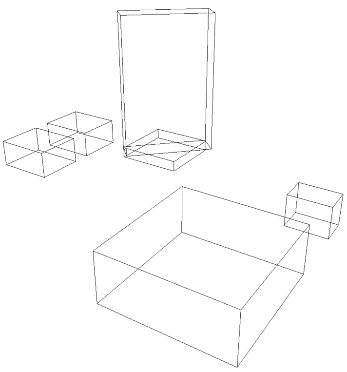
As illustrated in Figure 13, there are three space layers, where each layer has its own primal and dual space representation. Following the same indoor tracking example, Figure 14 illustrates the inter-layer relations between the dual spaces of the layers in Figure 12. In a topographic layer, the nodes represent the possible states of a navigating object and correspond to cells with volumetric extent in primal space (e.g., rooms) while the edges represent state transitions, i.e., the movement of an object from one space to another. They correspond to connectivity relations between the cells in primal space (e.g., neighboured rooms connected with a door). In the sensor space, the graph has a slightly different structure. The nodes represent again the cells with volumetric extend (e.g., the entire coverage space of a Wi-Fi transmitter), while the edges represent the transition from one space to another based on the neighbouring Wi-Fi coverage spaces. Since the layers cover the same real-world space, the separated dual graphs can be combined into a multi-layered graph.

Figure 14 represents relationships in the Dual space between the three Primal spaces given in Figure 13: topographic and two sensors’ spaces Wi-Fi and RFID. A novelty in IndoorGML 2.0 is the possibility to represent an inter-layer connection between two primal spaces. This is illustrated in Figure 15 where a the inter-layer mechanism is used to represent a furnished room with a combination of two layers: a first one describing solely the cells of the room and opening (Figure 15(b)) and a second one describing the CellSpaces of the furniture (Figure 15(c)). The relationship between the two layers can be qualified as a containment (layer 1 *contains* layer 2, or layer 2 is *within* layer 1). This allows describing complex scenes while respecting the non-overlapping constraint of Poincare duality.

Diagram

Description automatically generated

Figure 14: Inter-Layer relations between three different layers of a same environment

a) b) c)

Figure 15: Inter-layer connection between two primal spaces. (a) furnished room. (b) cells of the room and door only. (c) cells of furnishing elements only represented by minmax boxes.

## ~~Levels of Detail (LoD) 🡪 to be included in IndoorGML 2.0.1 (or 2.1)~~

~~(Something like)~~

|  |  |  |
| --- | --- | --- |
| ~~LoD~~ | ~~Description~~ | ~~Geometry~~ |
| ~~LoD 0~~ | ~~2D floorplan of each level~~ | ~~2D Footprint geometry for cells and cell boundary~~ |
| ~~LoD 1~~ | ~~Simplification to cube without detail description about walls. e.g. height of door is equal to cell~~ | ~~Prism model (Extrusion from 2D footprint geometry)~~ |
| ~~LoD 2~~ | ~~Detail description of features on walls. e.g. hieght of door may be independently given from wall~~ | ~~Detail 3D geometry~~ |
| ~~LoD 3~~ | ~~(? Texture ?)~~ |  |

# Data model

After explaining the important concepts on which IndoorGML relies, this section presents the conceptual data model using UML class diagram.

## IndoorGML Core Module

The core module is composed of three main parts:

* the primal space which describes the cellular space (see section 7.2);
* the dual space which carries the network information (see section 7.3);
* the inter-layer connection which makes the link between thematic layers (see section 7.6.2).

In Figure 16, the UML diagram illustrates all the classes associated with those three parts. In the following, the classes are introduced and the data types that they invoke in their attributes are detailed.

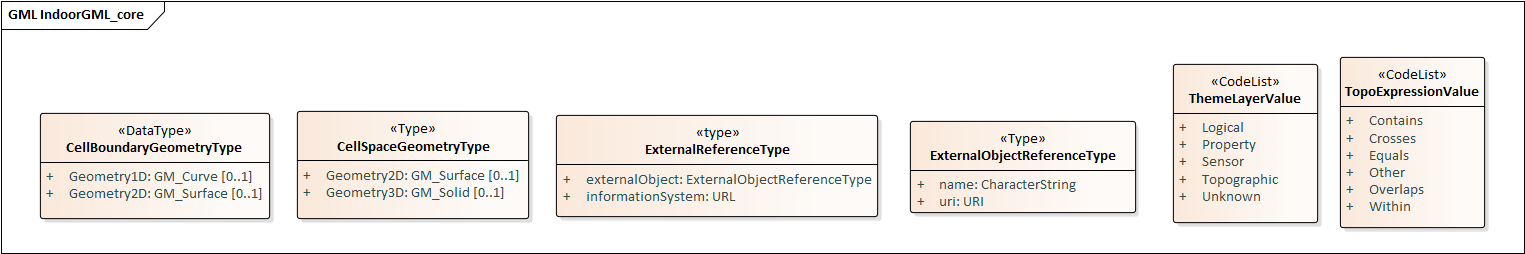
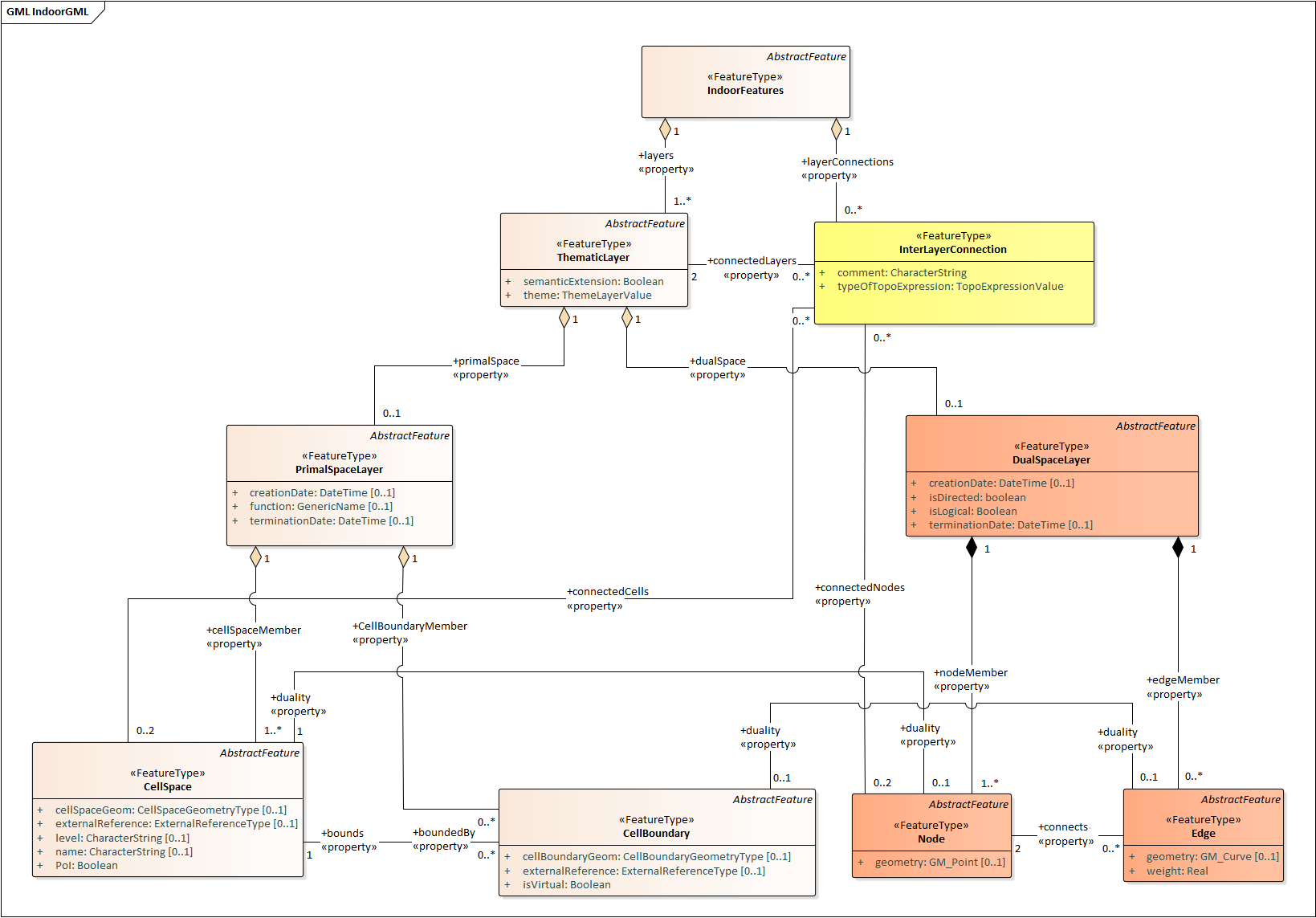
 

Figure 16: UML diagram of the Core module

### CellSpace

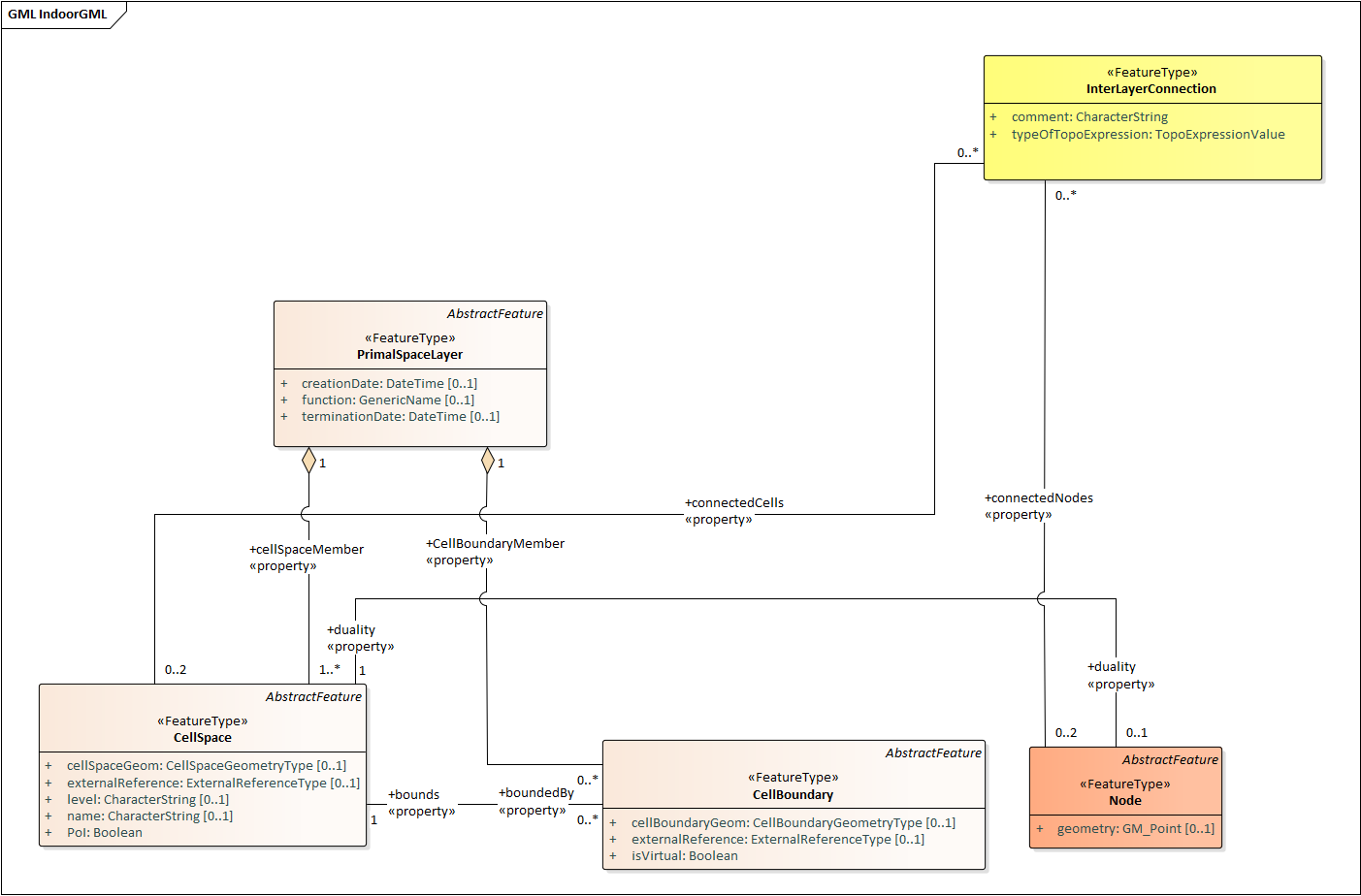


Figure 17: CellSpace and its related classes: PrimalSpaceLayer, CellBoundary, Node and InterLayerConnection

CellSpace is a core module class for representing the environment in terms of cellular space. CellSpace is compulsory class to have a valid IndoorGML2.0. It contains the following attributes (Figure 17):

* *cellSpaceGeom* (*CellSpaceGeometryType*)
* *externalReference (url)*
* *level (string)*
* *name (string)*
* *PoI (boolean)*

The *cellSpaceGeom* attribute carries an instance of type *CellSpaceGeometryType* allowing the description of geometric representations of space. A *CellSpaceGeometryType* is a geometry class type with two possible attributes: *Geometry3D* or *Geometry2D*. They provide 3D and 2D description of a CellSpace instance. The *Geometry3D* attribute describes a representation of type solid, similar to the *GM\_Solid* (ISO 19107:2003) type. It is the default type for describing a 3D CellSpace as one single valid entity. The *Geometry2D* attributes describes a representation of type surface, similar to the *GM\_Surface* type. It is meant for describing a CellSpace in 2D as one single surface (in the case of a 2D IndooGML model). The geometry should be valid according to the ISO 19107 standard terms. If a CellSpace cannot meet those requirements, e.g. be valid 2D or 3D geometry, the option to describe its geometry as a set of CellBoundary entities can be considered. The CellSpace can be defined without geometry as well.

The attribute *externalReference* is used for the reference of an object to its corresponding object in an external data set. A CellSpace also carries a *level* information, which can be left empty when it cannot be clearly identified. This is the case for example for a CellSpace that aggregates several cells spanning across multiple stories. The value of *level* is given as a string rather than an integer because it is sometime given as plain text “M” for mezzanine floor and “RC” for ground floor. A newly introduced attribute is *name*. This is destined to record the name given to a space according to any internal convention (e.g. MR.403 for meeting room 3 at level 4, or coverage of Wi-Fi 234). This is a common practice for large buildings and this attribute helps simplifying space queries for applications. Another new attribute *PoI* is introduced to allow CellSpace elements to be flagged as Point of Interest for LBS applications. The attribute is a simple Boolean allowing the implementation of special considerations for flagged cells.

Note that apart from the *PoI* attribute, all applicable attributes of a CellSpace can be null. For example, a network only IndoorGML model would not need a cellular space with explicit geometric description. However, CellSpace instances should always be described in an IndoorGML model (even without geometry attribute) as they may carry all the important information related to the primal space that other features from the dual space or other layers may need (e.g. a node can be identified as a *PoI* or associated with a *name* thanks to the attribute of its primal space).

In terms of relationships, a CellSpace instance can describe relationship with multiple CellBoundary entities, which represent its surrounding boundaries partially or fully through the *boundedBy* attribute. For example, choice can be made to store only boundaries which are important for the Dual Graph (e.g., boundaries that reflect adjacency between CellSpaces). In the case where a CellSpace does not carry the geometry of type Solid and uses a boundary-based representation instead, then all boundaries might be needed (to derive the geometry of the nodes or for visualisation). Finally, with the *duality* attribute, a CellSpace can describe a reference to one Node instance corresponding to its representation in the dual space.

CellSpace instances are aggregated in a PrimalSpaceLayer according to a specific theme as explained in Section 7.6. In case of multiple PrimalSpaceLayers, the class InterLayerConnection establishes the link between the depended CellSpace instances.

### CellBoundary

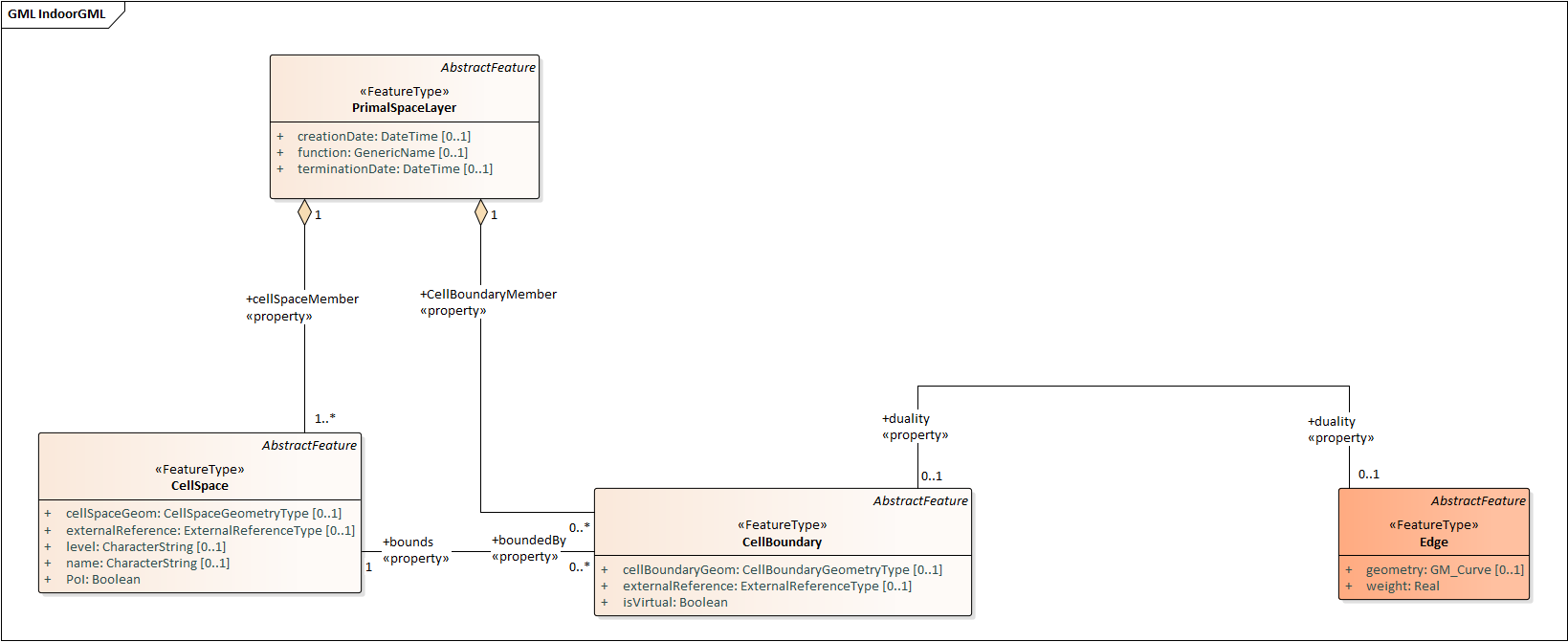


Figure 18: CellBoundary and its related classed: PrimalSpaceLayer, CellSpace and Edge

CellBoundary is a core module class to describe the boundary of each cell in a cellular space (Figure 18). Unlike CellSpace, CellBoundary is not a compulsory class. It is only required when Edge instances exist in the model. It contains the following attributes:

* *cellBoundaryGeom (CellBoundaryGeometryType)*
* *externalReference (url)*
* *isVirtual (boolean)*

The *cellBoundaryGeom* geometry attribute of the CellBoundary carries the geometry (of type *CellBoundaryGeometryType*) which is generally described by a surface in 3D or a curve in 2D. A *CellBoundaryGeometryType* is a geometry class type similar to the *CellSpaceGeometryType*, with two possible attributes: *Geometry2D* and *Geometry1D*. The *Geometry2D* attribute is the same than that of *CellSpaceGeometryType*. Note, in this context, it is embedded in 3D, i.e. it has 3D coordinates and represents a part of the boundary of a CellSpace. The *Geometry1D* attribute describes a representation of type curve, GM\_Curve type. Note, it is meant it is intended for describing a CellBoundary in 2D as one single line/curve and has 2D coordinates. This makes it adequate for representations based on 2D floor plans. CellBoundaryGeom can be omitted. In this case CellBoundaryGeom indicates only if a specific cell boundary is virtual.

The attribute *externalReference* is used for the reference of a geometric object to its corresponding object in an external data set and can be given by the url of the file containing the geometry. The *isVirtual* attribute is a Boolean value used to indicate whether a CellBoundary corresponds to a virtual surface (true) or a physical one (false), which should be the default value. Virtual boundaries are common in 3D indoor models, mainly when a space subdivision is applied.

Additionally, a CellBoundary can be linked to one Edge instance via the *duality* attribute, which corresponds to its dual representation. Unlike CellSpace, it is not a mandatory class in an IndoorGML model. In the case where there are CellSpace entities but no CellBoundary, the network should be derived from the cells using geometric operations.

In the case where there are CellBoundary entities provided without geometric attributes in the model, only logical networks can be safely derived between two CellSpace entities sharing any of those CellBoundary. Therefore, providing geometric networks will still involve similar issues described previously. A final scenario may see an IndoorGML model with geometry information only with CellBoundary instances but not for CellSpace. That case is likely to happen if a solid geometry cannot be provided for a CellSpace, and a set of surface boundaries are provided with no guarantee of closure. In that case the generation of a Node for a CellSpace should be completed from CellBoundary instances, while guaranteeing its position inside the described space.

### PrimalSpaceLayer

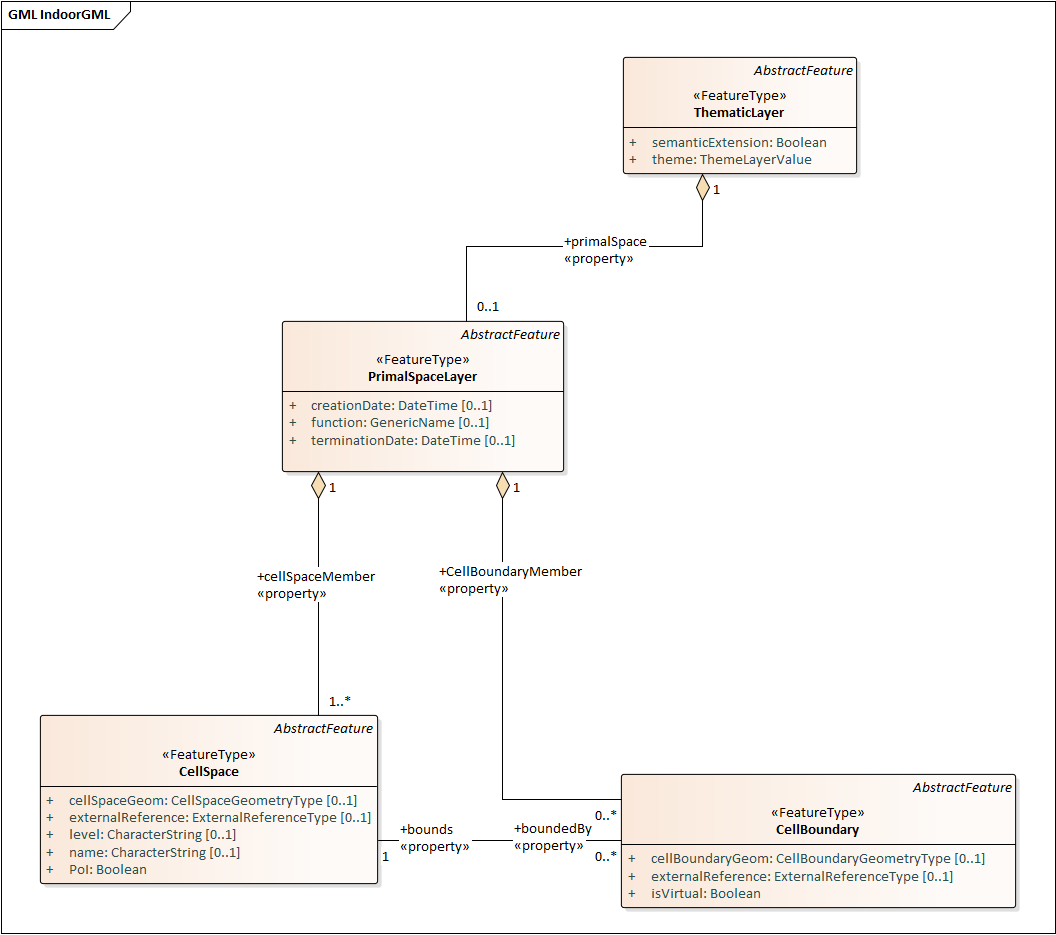


Figure 19: PrimalSpaceLayer and its related classes: CellSpace, CellBoundary and Thematic Layer

PrimalSpaceLayer is a core module class representing the primal cellular spaces of a given thematic layer (Figure 19). It aggregates CellSpace and CellBoundary (which are directly associated with their corresponding geometry attributes) to represent spatial objects in primal space. The PrimalSpaceLayer class has the following attributes:

* *function (CodeList)*,
* *creationDate (DateTime)*,
* *terminationDate (DateTime)*.

With the attribute *function*, nominal and real functions of a space layer are depending on the Thematic layer and can be described as proposed in a CodeList. The *creationDate* and *terminationDate* attributes can be used to describe the chronology of the layer. The points of time refer to real world times.

A PrimalSpaceLayer instance also provides references to its CellSpace and CellBoundary entities through the *cellSpaceMember* and *cellBoundaryMember* elements.

### Node

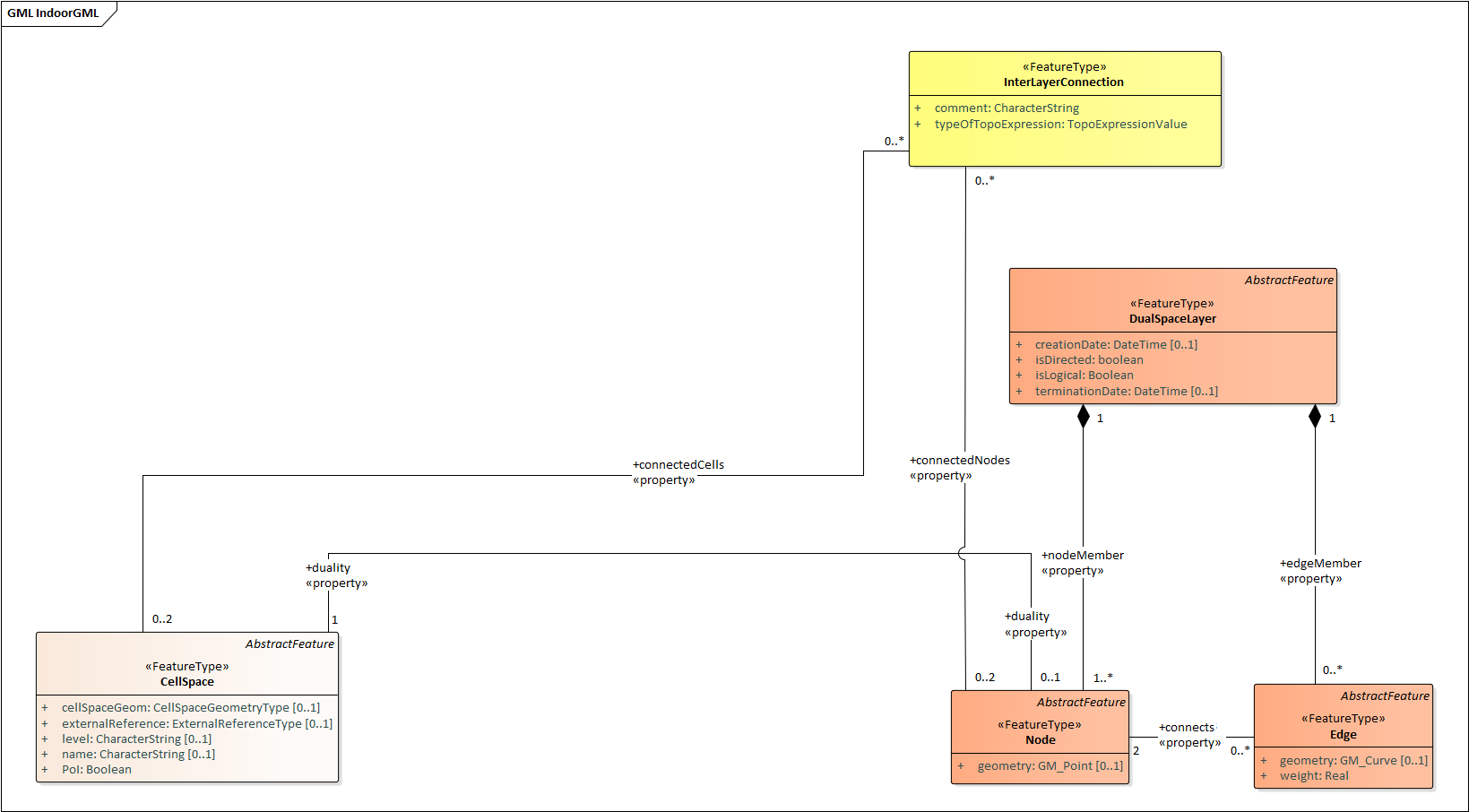


Figure 20: Node and its related classes: CellSpace, Edge, DualSpaceLayer and InterLayerConnection

Node is a core module class to represent a node in dual space (Figure 20). It has one attribute:

* *geometry (GM\_Point)*.

The value of *geometry* corresponds to a 2D or 3D Point in IndoorGML, but its cardinality can be 0 (no geometry provided) or 1. Because a Node is always the dual space abstraction of a primal space cell, it has always an association with its corresponding CellSpace (e.g. room, door, sensor coverage, etc.) through the *duality* attribute. This way, a Node can always access to the information related to the cell it is representing (e.g., geometry, semantic, etc.). Note that the associated CellSpace may not carry any information as well, except the functional information for the specific cellular space. Additionally, a Node is also associated with at least one Edge instance that is linked to it via the *connects* attribute.

### Edge

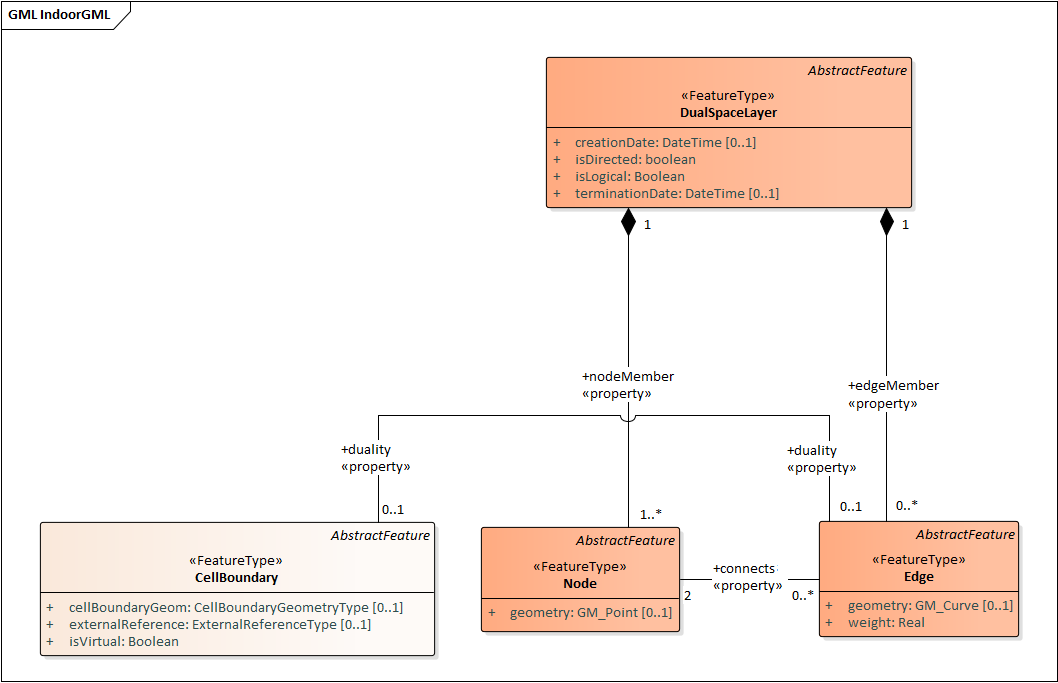


Figure 21: Edge and its related classes: CellBoundary, Node and DualSpaceLayer

Edge is a core module class that represents the adjacency or connectivity relationships among Node elements representing space cells in primal space (Figure 21). It carries the following attributes:

* *geometry (GM\_Curve),*
* *weight (real)*.

The attribute *geometry* provides the description of a 2D or 3D curve, but similarly to Node entities its cardinality can be 0 or 1 as well. The attribute *weight* can be used for graph-based applications (e.g., in order to deal with the impedance representing absolute barriers in transportation problems).

An Edge may be associated with a CellBoundary instance of the primary space via its *duality* attribute. This association can be skipped in situations where a CellBoundary is not necessary to represent the link between two CellSpace entities (e.g. for logical networks or visibility graphs where two CellSpaces connected by visibility may not share a CellBoundary). Finally, an Edge always connects two Nodes.

### DualSpaceLayer

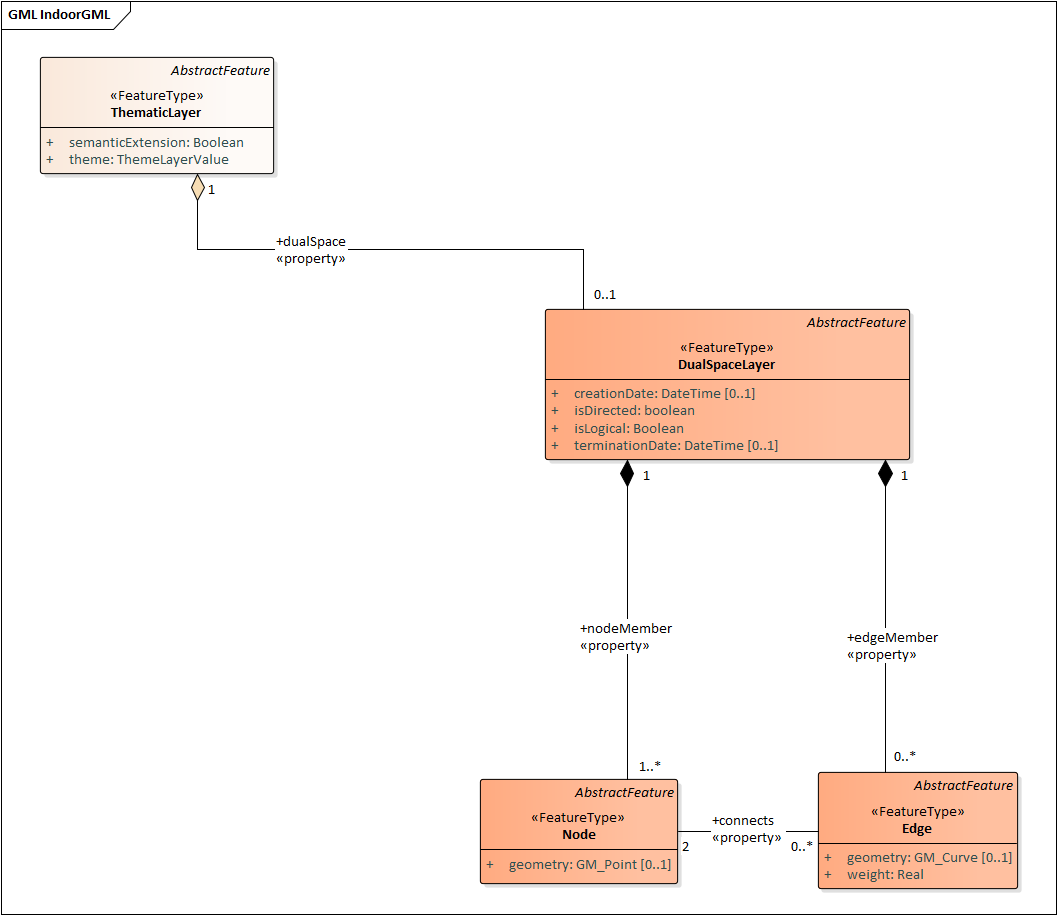


Figure 22: DualSpaceLayer and its related classes: Node, Edge and Thematic Layer.

DualSpaceLayer is a feature class for representing the dual space features (e.g., room network) of a given thematic layer. It is composed of Nodes and Edges to represent the topology of objects from the primal space. It has the following attributes:

* *isLogical* (boolean)
* *isDirected* (boolean)
* *creationDate (DateTime)*
* *terminationDate (DateTime)*

While *creationDate* and *terminationDate* are similar to those of PrimalSpaceLayer, the *isLogical* attribute allows to differentiate whether the provided network is a geometric or a logical network. This difference may matter for certain applications such as navigation, where a logical network would not be sufficient to evaluate travel distances between cells. Similarly, the *isDirected* attribute allows to specify if the graph associated with the DualSpaceLayer is directed or not. A directed graph implies that the node directions should be considered in the applications. Currently, the order of the nodes in the implementation formats determines their direction. Additionally, a DualSpace provides references to all its related Node and Edge entities through its *nodeMember* and *edgeMember* attributes.

### InterLayerConnection

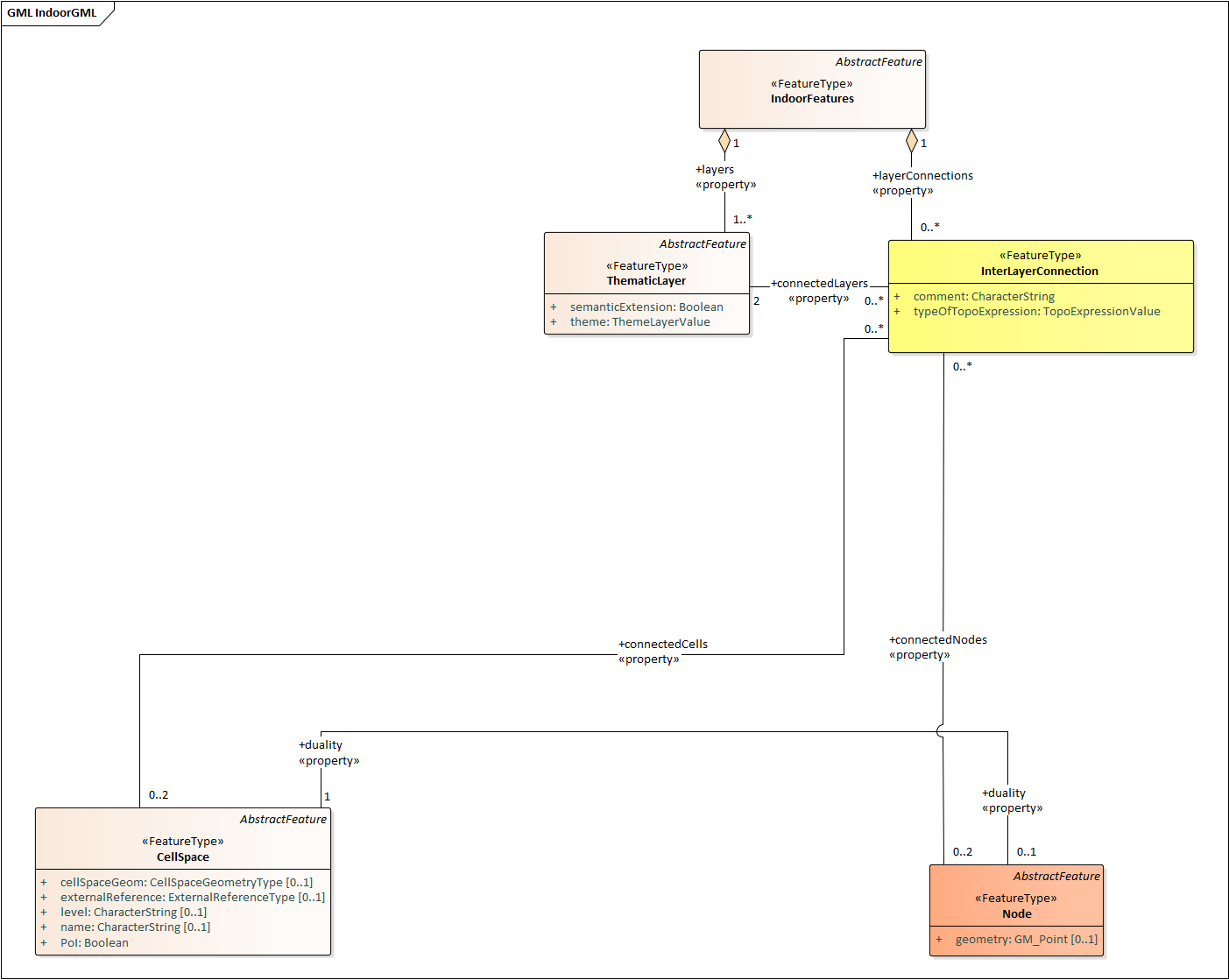


Figure 23: InterLayerConnection and its related classes: CellSpace, Node, ThematicLayer and IndoorFeatures

The InterLayerConnection class describes the connection between two layers in IndoorGML, either of type PrimalSpaceLayer or DualSpaceLayer (Figure 23). It contains the following attributes:

* *typeOfTopoExpression (TopoExpressionValue)*
* *comment* (string)
* *connectedLayers* (ref. to ThematicLayer)

The *typeOfTopoExpression* attribute represents the topological relationship between two layers. It comes as a code list with the following values: *contains, within, covers, coveredBy*, *overlaps and equals*. Those topological values are in the form of verbs for which the subject is the first instance of the *connectedLayers* attribute. In other words, for two layers successively described by the *connectedLayers* attribute, e.g Layer 1 and Layer 2, one should read Layer 1 *typeOfTopoExpression* Layer 2 (e.g., Layer Room *contains* Layer Furniture).

An InterLayerConnection also describes the cells or nodes that are connected between two layers, using the *connectedCells* and/or *connectedNodes* attributes. The former is used when the connection is between two primal spaces and the latter is used otherwise. Finally, the *comment* attribute can contain an additional description for the InterLayerConnection.

### ThematicLayer

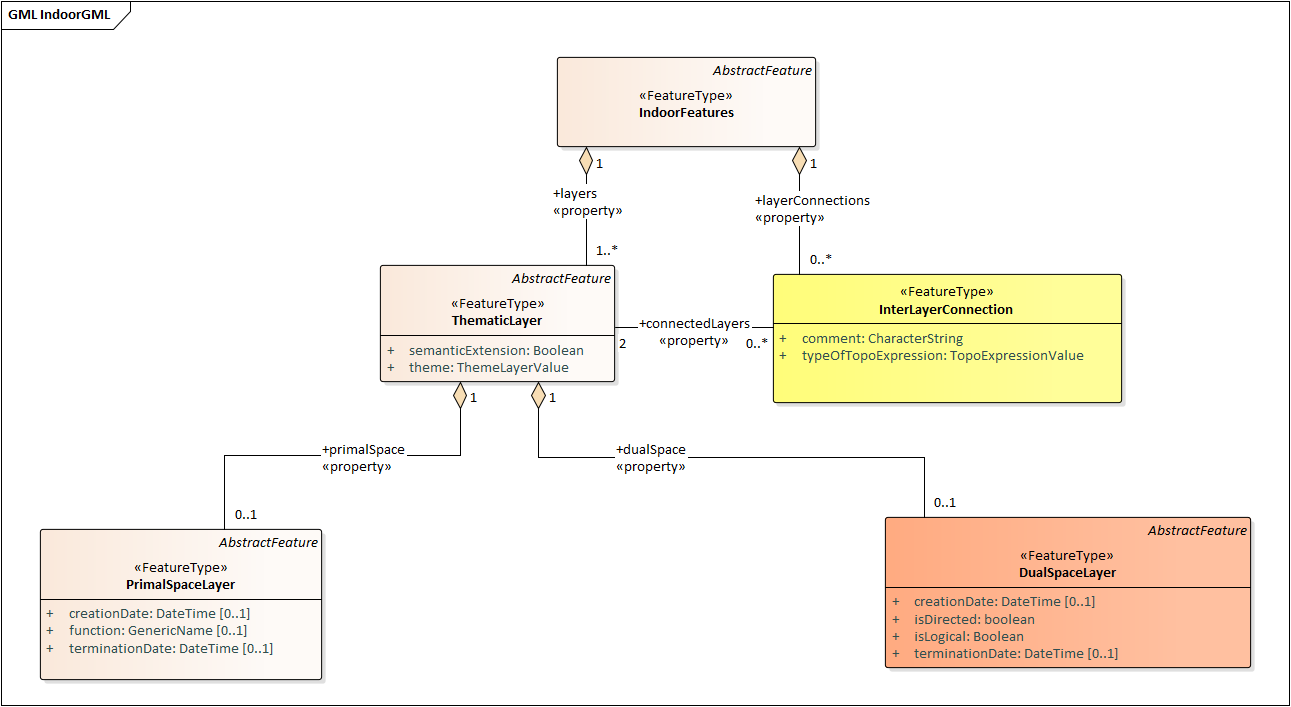


Figure 24: ThematicLayer and its related classes: PrimalSpaceLayer, DualSpaceLayer, InterLayerConnection and IndoorFeatures

The ThematicLayer is a core module class introduced in IndoorGML2.0, as an aggregation of PrimalSpaceLayer and DualSpaceLayer instances to allow definition of Thematic layers separately (Figure 24). Note, IndoorGML1.1 enables the multi-layer mechanism only for the dual space (the networks).

The class comes with the following attributes:

* *semanticExtension* (boolean)
* *theme (ThemeLayerValue)primalSpace* (ref. to PrimalSpaceLayer)

The *semanticExtension* attributeis set as a boolean as it is simply an indication that there is Extension module with additional semantic information associated to the PrimalSpaceLayer. IndoorGML 2.0 maintain only the Navigation extension module (see Section 8.5), a boolean is considered enough to indicate its presence. This is however susceptible to evolve in the future (e.g. into a codeList). The *theme* attribute determines what type of representation of the model can be expected in the corresponding layer (e.g topographic). It comes in the form of a code list which tells whether the layer is of type Physical, Virtual, Tags or Unknown.

A *Physical* layer is a layer that describes the indoor space on the basis of its physical constraints (e.g. the topographic cellular space in Figure 12). It is the most common type of layers for applications like indoor navigation, where the physical elements are highly constraining the use of the space. Similarly, a layer is qualified as *Virtual* when its description of the space relies on exclusively virtual, or a combination of physical and virtual extents. It is the case for example for functional spaces that can represent spaces necessary for some indoor objects to operate or to be used properly (Diakité, 2018). It is also the case for sensor spaces such as the WiFi spaces represented in Figure 12. Finally, the *Tags* type is useful for describing layers that use symbols or tags to represent the cellular space. It is a useful representation when the real geometry of the CellSpaces of a given layer are not relevant for a given application. PoI are often represented in a separate layer with their locations only (e.g., in Dual Space). Finally, any layer the does not fall in those previous categories will take the *Unknown* type.

## Navigation extension module

The Navigation extension module provides semantic information for indoor space to support indoor navigation applications (Figure 25). The IndoorGML 2.0 semantics includes concepts related to navigability and connectivity between cells, obstacles and objects, as well as, routes for specific users. Further specialisation of cell is made available by introducing attributes that can be used for additional navigation constraints such as temporal access related to as opening hours, or constraints resulting from properties of the navigation path.

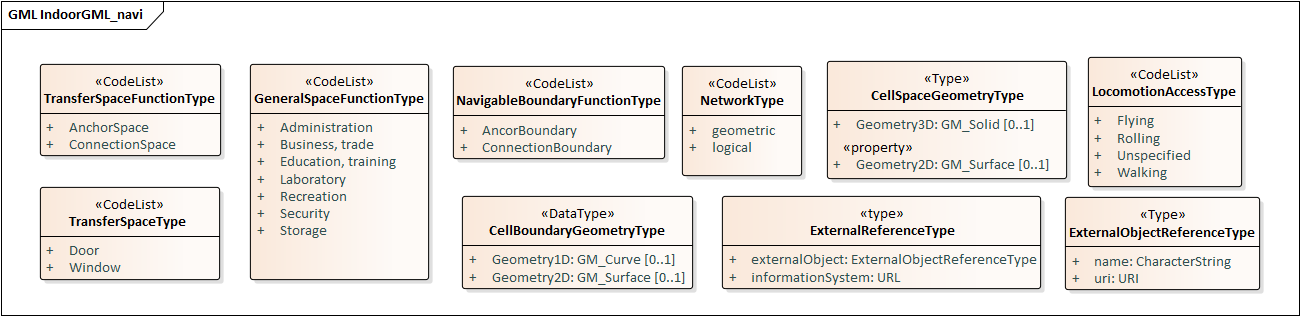
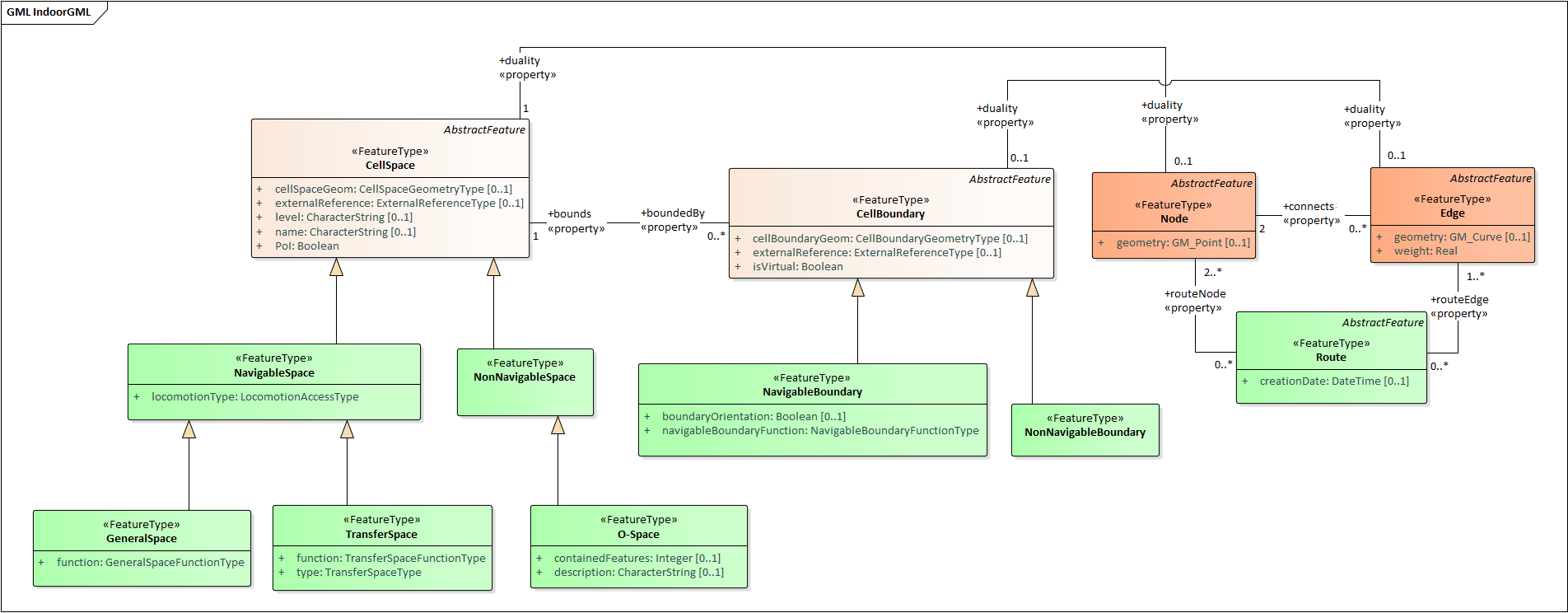
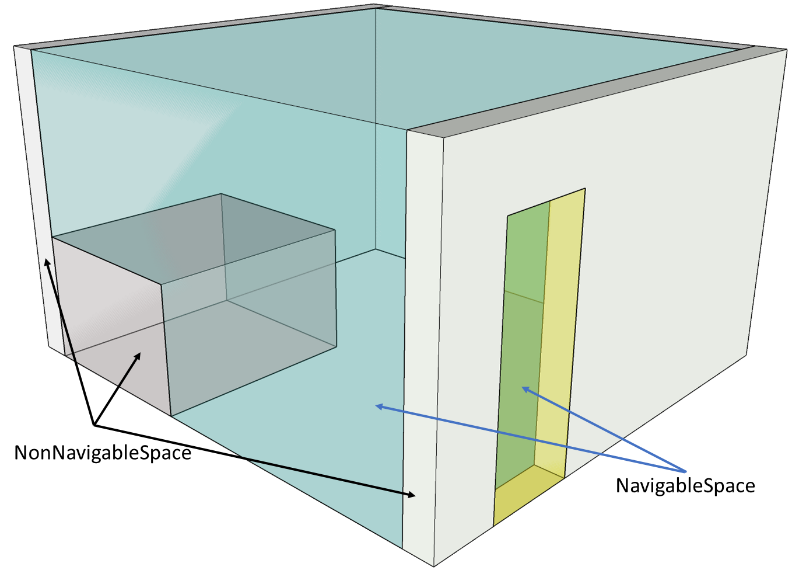
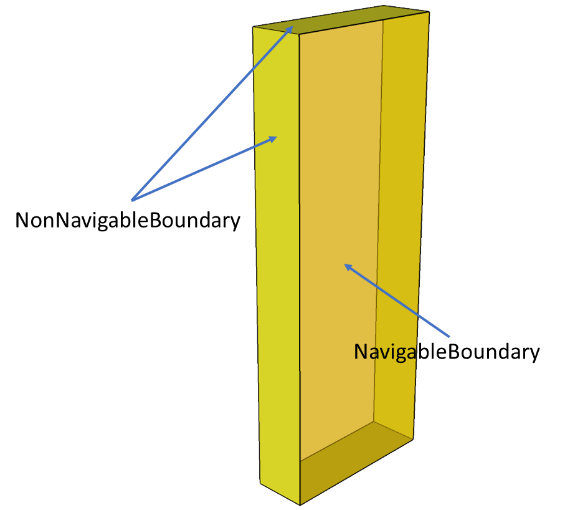
 

Figure 25: UML diagram of the Navigation Extension Module (classes in green)

The space cells are classified into two major groups: *NavigableSpace* and *NonNavigableSpace*. NavigableSpace represents all indoor spaces (e.g., rooms, corridors, windows, stairs) that can be used by a navigation application. Spaces connecting others are also considered by this class (e.g., openings). NonNavigableSpace represents all indoor spaces that are not navigable, either because they are physically occupied by indoor features (e.g., furniture, walls) or because of other navigation constraints (e.g., accessibility). Both NavigableSpace and NonNavigableSpace are child’s classes of CellSpace. Figure 26a illustrates such spaces on a 3D model.

NavigableBoundary and NonNavigableBoundary represents boundaries of NavigableSpace and NonNavigableSpace respectively. They allow to describe the navigability of the spaces’ sides. For example, for the door space in Figure 26b, the sides that are meeting with the walls are of class NonNavigableBoundary, and the rest are NavigableBoundary. They are child’s classes of the CellBoundary class. The association of CellSpace and CellBoundary classes with Node and Edge in IndoorGML core module ensures a link between the navigation module and the dual space.

1. b)

Figure 26: Navigable and Non-navigable spaces (a) and boundaries (b) on a 3D model with walls and furniture (grey), indoor space (blue) and a door space (yellow).

### NavigableSpace

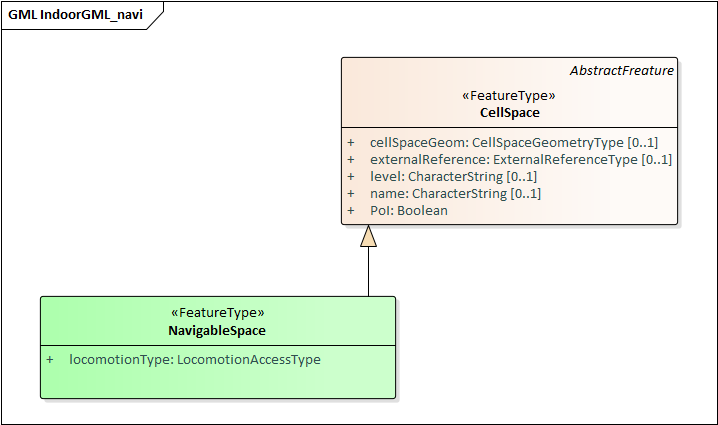


Figure 27: NavigableSpace and its related class: CellSpace

The NavigableSpace class denotes a space in which users can move freely. It has two subclasses GeneralSpace and TransferSpace (Figure 27). The subclasses are classified depending on the purpose of the space. The compartmentalized spaces such as corridor, door, lobby, hallway, big room are represented as NavigableSpace. Note, door is represented as NavigableSpace as shown in Figure 26, especially in 3D. In 2D, doors are commonly represented as boundaries of rooms and have to be considered NavigableBoundaries (see Section 8.5.3)

NavigableSpace entities can carry information about the type of locomotion that they allow, through the *locomotionType* attribute, which is one of the following values: *Flying, Rolling, Unspecified and Walking.* A Navigable space may handle one or several of the locomotion types listed. Note, the class instances inherit the geometry of its parent CellSpace entity and can therefore be represented as gml:Solid on 3D data model or gml:Surface on 2D data model.

### GeneralSpace

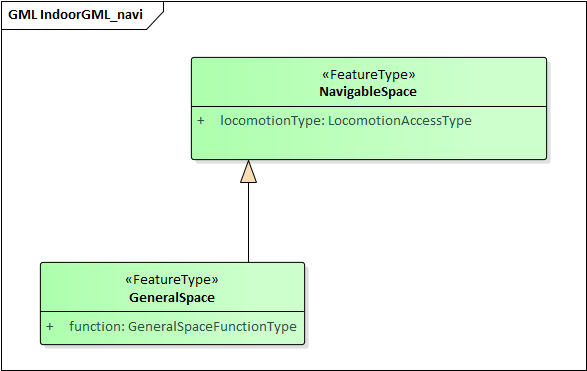


Figure 28: General Space and its related class: NavigableSpace

The GeneralSpace class is one of the two subclasses of NavigableSpace (Figure 28). GeneralSpace is identified as any navigable cells such as rooms, lobbies, kitchen, etc., which agents can use for a longer period of time and can serve as starting and target cell in navigation. It carries the attribute *function* which give details about the function of the cell. In IndoorGML, those functions are described in a code list derived from OmniClass Table 13 (OmniClass, 2021) (see Annexe B).

### TransferSpace

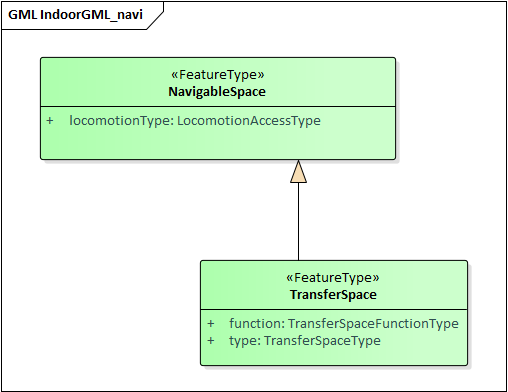


Figure 29: Transfer Space and its related class: NavigableSpace

The class TransferSpace is specialisation of NavigableSpace. It is used to model a space that provide passages between GeneralSpaces. Thereby, it typically describes openings (mainly doors but also windows) for horizontal transfer and entrances to staircase or lift cells for vertical transfers. Similarly to the GeneralSpace class, it carries a *function* attribute that describes whether the space is an AnchorSpace (a space allowing to connect the indoor and the outdoor) or a BoundarySpace (a space connecting two indoor or two outdoor spaces). Another of its attribute is *type* which specified through a codeList the TransferSpaceType (Door or Window).

### NavigableBoundary

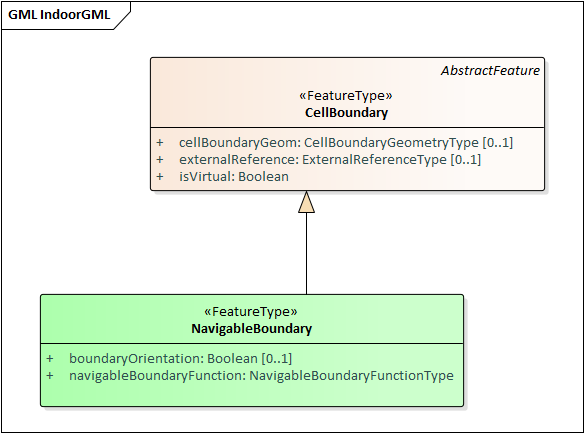


Figure 30: Navigable Boundary and its related class: CellBoundary

The NavigableBoundary class is a specialisation of a CellBoundary and provides further information related to NavigableSpace (Figure 30). As illustrated in Figure 26, it typically represents the space boundaries that correspond to entrances or exits through which agents navigate from one cell to another. It is therefore mainly found between GeneralSpace and TransferSpace cells but can happen between two GeneralSpace cells as well (e.g., in the case of a room subdivided to distinguish areas of different purposes).

A NavigableSpace is necessarily bound by at least one NavigableBoundary. In the specific case of a TransferSpace, it is expected to have at least two NavigableBoundary instances bound to it, as a TransferSpace serve for transition between connected spaces.

The class carries a *boundaryOrientation* attribute and a *navigableBoundaryFunction* attribute specifying if the boundary is an *AnchorBoundary* or a *ConnectionBoundary* (see subsection 8.5.2 for more details).

### NonNavigableSpace

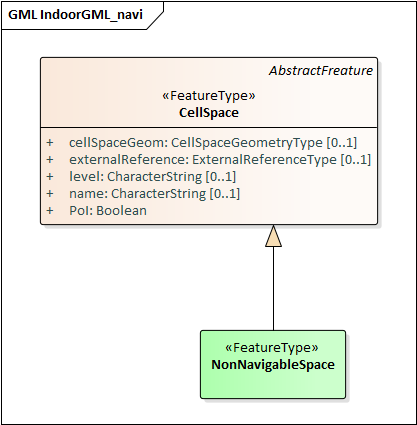


Figure 31: NonNavigableSpace its related class: CellSpace

The NonNavigableSpace class represents cells that are occupied by obstacles (Figure 31). It can correspond to the structural elements of a building (walls, slabs, etc) or other indoor features populating the space (furniture, appliances etc.). It is a class without attributes, but opens options to classify further the non-navigable cells.

### ObjectSpace

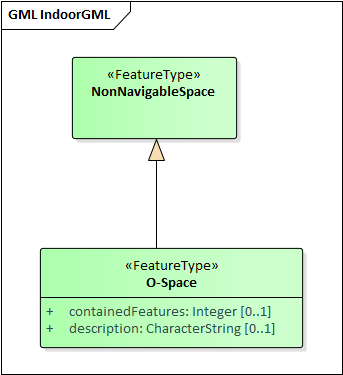


Figure 32: ObjectSpace and its related class: NonNavigableSpace.

The ObjectSpace (Figure 32) class is meant to bring additional details to a NonNavigableSpace when it contains some objects that makes it non-navigable. The class has two attributes: *containedFeatures (Integer),* and *description (String).*

The *containedFeatures* attribute is an integer that describes the number of objects encapsulated within the ObjectSpace and thus, by extension within the parent NonNavigableSpace. The objects in question can be represented in a different layer of the model and the link to the corresponding ObjectSpace can be made through an InterLayerConnection instance with a *within* or *contains* relationship. The *description* attribute is meant to provide any relevant information regarding the objects contained within the space in plain text.

### NonNavigableBoundary

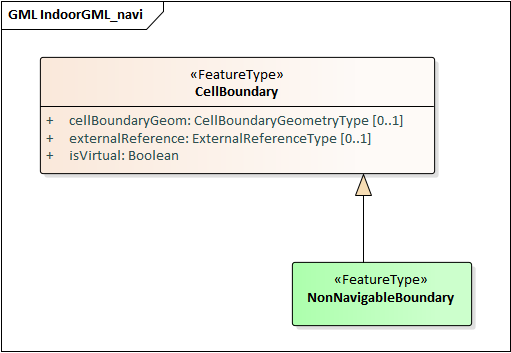


Figure 33: NonNavigableBoundary and its related classes: CellBoundary

NonNavigableBoundary entities represent the boundaries between two NonNavigableSpace cells or between a NavigableSpace and a NonNavigableSpace cells (Figure 31). As such, it is the type of boundary that can be found typically at the lateral sides of a TransferSpace (see Figure 26 b), corresponding for example to the walls surrounding a door.

### Route

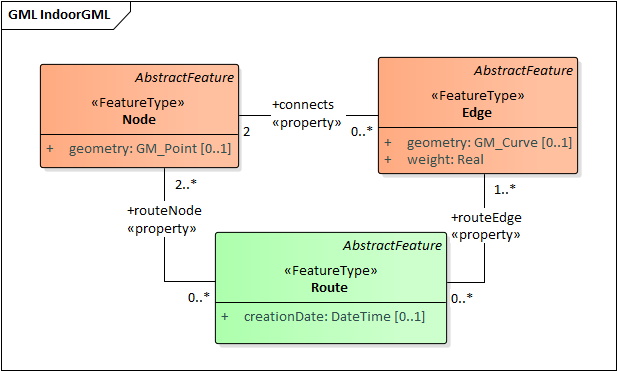


Figure 34: Route and its related classes: Node and Edge

The Route class is a specialisation of a Dual space that represents a subset of Network (logical or physical), which includes a path to navigate through indoor space. It is usually defined as the result of a path finding query.

It has one attribute which is *creationDate.* Because dynamic indoor environment may imply change in space availability and accessibility, a path at a given time may not be suitable anymore at another time. For this reason, the *creationDate* attribute helps indicating at which time a given route was created. The *routeNode* and *routeEdge* attributes are both ordered sequences of Node and Edge references to describe the different parts of the route path. Therefore, the first and last *routeNode* elements correspond respectively to the starting and destination points of the route.

# Data dictionary and requirements

In this section, we present the data dictionary of the feature types defined in IndoorGML 2.0 UML class diagram. It aims to clarify the concepts of each feature type and help the implementation of this standard. The data dictionary is defined based on ISO standards from TC 211, particularly ISO 19109 for the rules for application schema, ISO 19107 for spatial schema, and ISO 19136 for GML. As IndoorGML 2.0 is an application schema from these base standards, we will not include the data dictionary for the feature types defined by these standards in section. For example, the properties of GML AbstractFeature such as gmlID, and name are not described in the data dictionary. The data dictionary of the feature types defined in section 8 is given in the following subsections.

## Feature Types in Core Module

### IndoorFeatures

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **IndoorFeatures** | | |
| **Definition** | Set of all features and their relationships to describe a given indoor space. | | |
| **Super classes** | GML AbstractFeature | | |
| **Composition** | **Role name** | **Type and Cardinality** | |
| layers | ThematicLayer [1..\*] | |
| layerConnections | InterLayerConnection [0..\*] | |
| **Properties** | **Property name** | **Type and Cardinality** | |
| (none) |  | |
| **Constraints** | **Constraint ID** | | **Constraint** |
| (none) | |  |

### ThematicLayer

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **ThematicLayer** | | |
| **Definition** | Aggregation of features for a specific theme consisting of primal space layer and dual space layer. | | |
| **Super classes** | GML AbstractFeature | | |
| **Association** | **Role name** | **Type and Cardinality** | |
| connectedLayers | InterLayerConnection [1..1] | |
| **Properties** | **Property name** | **Type and Cardinality** | |
| semanticExtension | Boolean [1..1] | |
| theme | ThematicLayerValue [1..1] | |
| **Constraints** | **Constraint ID** | | **Constraint** |
| Indoorgml2/constraints/ thematiclayer-1 | | Any feature of a thematic layer shall belong to the same theme. |

### PrimalSpaceLayer

|  |  |  |
| --- | --- | --- |
| **Name** | **PrimalSpaceLayer** | |
| **Definition** | Aggregation of cell spaces and cell boundaries describing the topography of a given theme in indoor space. | |
| **Super classes** | GML AbstractFeature | |
| **Aggregation** | **Members** | **Class and Cardinality** |
| cellSpaceMember | CellSpace [1..\*] |
| cellBoundaryMember | CellBoundary [0..\*] |
| **Properties** | **Property name** | **Type and Cardinality** |
| function | GenericName [0..1] |
| creationDate | DateTime [0..1] |
| terminationDate | DateTime [0..1] |
| **Constraints** | **Constraint ID** | **Constraint** |
| Indoorgml2/constraints/ primalspacelayer-1 | (?) |

### CellSpace

|  |  |  |
| --- | --- | --- |
| **Name** | **CellSpace** | |
| **Definition** | the basic unit of indoor space, such as room and corridor, the union of which makes the entire indoor space. | |
| **Super classes** | GML AbstractFeature | |
| **Association** | **Role name** | **Type and Cardinality** |
| boundedBy | CellBoundary [0..\*] |
| duality | Node [0..1] |
| **Properties** | **Property name** | **Type and Cardinality** |
| cellSpaceGeometry | CellSpaceGeometryType [0..1] |
| externalReference | ExternalReferenceType [0..1] |
| level | CharacterString [0..1] |
| name | CharacterString [0..1] |
| PoI | Boolean [1..1] |
| **Constraints** | **Constraint ID** | **Constraint** |
| Indoorgml2/constraints/cellspace-1 | Cell spaces belonging to the same primal space layer should not overlap. |

### CellBoundary

|  |  |  |
| --- | --- | --- |
| **Name** | **CellBoundary** | |
| **Definition** | explicit boundary of cell space, to which we may assign additional properties such as material, texture, etc. | |
| **Super classes** | GML AbstractFeature | |
| **Association** | **Role name** | **Type and Cardinality** |
| bounds | CellSpace [1..2] |
| duality | Edge [0..1] |
| **Properties** | **Property name** | **Type and Cardinality** |
| cellBoundaryGeometry | CellBoundaryGeometryType [0..1] |
| externalReference | ExternalReferenceType [0..1] |
| isVirtual | Boolean |
| **Constraints** | **Constraint ID** | **Constraint** |
| Indoorgml2/constraints/ cellboundary-1 | Cell boundaries belonging to the same primal space layer should not intersect. |
| Indoorgml2/constraints/ cellboundary-2 | The geometry of cell boundary shall not exceed the extent of the corresponding cell space |

### DualSpaceLayer

|  |  |  |
| --- | --- | --- |
| **Name** | **Node** | |
| **Definition** | Dual space layer corresponds to primal space layer and mainly describes adjacency or connectivity relationship between nodes, where node is an abstraction of cell space and edge is a relationship between two nodes. It is a graph composed of nodes and edges. | |
| **Super classes** | GML AbstractFeature | |
| **Aggregation** | Role Name | Aggregated Class and Cardinality |
| nodeMember | Node [1..\*] |
| edgeMember | Edge [0..\*] |
| **Property** | **Property name** | **Type and Cardinality** |
| isLogical | GM\_Curve [0..1] |
| creationDate | DateTime [0..1] |
| terminationDate | DateTime [0..1] |
| isDirected | Boolean [1..1] |
| **Constraints** | **Constraint ID** | **Constraint** |
| none |  |

### Node

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Node** | | |
| **Definition** | space abstraction of cell space in dual space to a point or virtual point, which is defined as 0-dimentional topological primitive in ISO 19107. | | |
| **Super classes** | GML AbstractFeature | | |
| **Association** | **Role name** | **Type and Cardinality** | |
| connectedNodes | InterLayerConnection [0..\*] | |
| duality | CellSpace [0..1] | |
| connects | Edge [0..\*] | |
| **Properties** | **Property name** | **Type and Cardinality** | |
| geometry | GM\_Point [0..1] | |
| **Constraints** | **Constraint ID** | | **Constraint** |
| Indoorgml2/constraints/node-1 | | When the isLogical attribute of a DualSpaceLayer is set to TRUE, the geometries of its Node instances shall be spatially located inside of their corresponding CellSpaces. |

### Edge

|  |  |  |
| --- | --- | --- |
| **Name** | **Edge** | |
| **Definition** | adjacency or connectivity relationship between nodes, which is defined as 1-dimensional topological primitive in ISO 19107. | |
| **Super classes** | GML AbstractFeature | |
| **Association** | **Role name** | **Type and Cardinality** |
| connects | Node [2..2] |
| duality | CellBoundary [0..1] |
| **Properties** | **Property name** | **Type and Cardinality** |
| geometry | GM\_Curve [0..1] |
| weight | Real [1..1] |
| **Constraints** | **Constraint ID** | **Constraint** |
| Indoorgml2/constraints/ edge-1 | No self-intersection is allowed when its geometry is given. |
| Indoorgml2/constraints/ edge-2 | If dualspaceLayer.directed=true, then the order of nodes represents the direction. |

### InterLayerConnection

|  |  |  |
| --- | --- | --- |
| **Name** | **InterLayerConnection** | |
| **Definition** | Relationship between cell spaces and nodes in two different thematic layers | |
| **Super classes** | None | |
| **Association** | **Role name** | **Type and Cardinality** |
| connectedLayers | ThematicLayer [2..2] |
| connectedNodes | Node [0..2] |
| connectedCells | CellSpace [0..2] |
| **Properties** | **Property name** | **Type and Cardinality** |
| comment | CharacterString [1..1] |
| typeOfTopoExpression | TopoExpressiveValue [1..1] |
| **Constraints** | **Constraint ID** | **Constraint** |
| Indoorgml2/constraints/  interlayerconnection-1 | Two target cell spaces (or nodes) shall not belong to a same primal space layer (or dual space layer) |
| Indoorgml2/constraints/  InterLayerConnection-2 | Connected nodes or connected cells shall be consistent with connected layers. It means that the target cell spaces (or nodes) shall belong to primal space layer (or dual space layer) of the connected layer |
| Indoorgml2/constraints/  InterLayerConnection-3 | The cardinalities of Node and CellSpace can either be 0 or 2, but can never be 1. |

## Feature Types in Navigation Module

### NavigableSpace

|  |  |  |
| --- | --- | --- |
| **Name** | **NavigableSpace** | |
| **Definition** | a cell space in which users can move freely | |
| **Super classes** | CellSpace | |
| **Properties** | **Property name** | **Type and Cardinality** |
| locomotionType | LocomotionAccessType [1..1] |
| **Constraints** | **Constraint ID** | **Constraint** |
| None |  |

### NonNavigableSpace

|  |  |  |
| --- | --- | --- |
| **Name** | **NavigableSpace** | |
| **Definition** | a cell space in which users cannot move | |
| **Super classes** | CellSpace | |
| **Constraints** | **Constraint ID** | **Constraint** |
| None |  |

### GeneralSpace

|  |  |  |
| --- | --- | --- |
| **Name** | **GeneralSpace** | |
| **Definition** | A type of NavigableSpace such as rooms, lobbies, kitchen, etc., where agents can stay or use  for a longer period of time and can serve as starting and target cell in navigation. | |
| **Super classes** | NavigableSpace | |
| **Properties** | **Property name** | **Type and Cardinality** |
| function | GeneralSpaceFunctionType [1..1] |
| **Constraints** | **Requirement ID** | **Constraint** |
| None |  |

### TransferSpace

|  |  |  |
| --- | --- | --- |
| **Name** | **TransferSpace** | |
| **Definition** | A type of NavigableSpace that provides passages between GeneralSpaces | |
| **Super classes** | NavigableSpace | |
| **Properties** | **Property name** | **Type and Cardinality** |
| function | TransferSpaceFunctionType [1..1] |
| **Constraints** | **Requirement ID** | **Constraint** |
| None |  |

### ObjectSpace

|  |  |  |
| --- | --- | --- |
| **Name** | **ObjectSpace** | |
| **Definition** | A type of NonNavigableSpace containing objects that make it non-navigable | |
| **Super classes** | NonNavigableSpace | |
| **Association** | **Role name** | **Associated Class** |
| (none | (none) |
| **Properties** | **Property name** | **Type and Cardinality** |
| containedFeature | integer [0..1] |
| ~~externalReference~~ | ~~ExternalReferenceType [0..1]~~ |
| description | string [0..1] |
| **Constraints** | **Requirement ID** | **Constraint** |
| Indoorgml2/constraints/  objectspace - 1 | ObjectSpace has to be separated from cell spaces to form another space layer as no cell space shall overlap with others. |
| Indoorgml2/constraints/  objectspace - 1 | ObjectSpace instances also fall under the non-overlapping constraint of CellSpaces. As such, they should not overlap with any other CellSpace or its specialized classes. Therefore, ObjectSpace can either be carved out of the space containing them or they can be defined in different layers (to avoid complex Boolean operations for example). |

### NavigableBoundary

|  |  |  |
| --- | --- | --- |
| **Name** | NavigableBoundary | |
| **Definition** | A type of CellBoundary, which agents can pass through. | |
| **Super classes** | CellBoundary | |
| **Properties** | **Property name** | **Type and Cardinality** |
| boundaryOrientation | Boolean [0..1] |
| navigableBoundaryFunction | NavigableBoundaryFunctuinType [1..1] |
| **Constraints** | **Requirement ID** | **Constraint** |
| None |  |

### NonNavigableBoundary

|  |  |  |
| --- | --- | --- |
| **Name** | NavigableBoundary | |
| **Definition** | A type of CellBoundary, which does not allow passage. | |
| **Super classes** | CellBoundary | |
| **Properties** | **Property name** | **Type and Cardinality** |
| None |  |
| **Constraints** | **Requirement ID** | **Constraint** |
| None |  |

### Route

|  |  |  |
| --- | --- | --- |
| **Name** | Route | |
| **Definition** | A path to navigate between two nodes  (look at any other OGC or ISO TC 204?) | |
| **Super classes** | GML AbstractFeature | |
| **Association** | **Role name** | **Associated Class and Cardinality** |
| routeNode | Node [2..\*] |
| routeEdge | Edge [1..\*] |
| **Properties** | **Property name** | **Type and Cardinality** |
| creationDate | DateTime [0..1] |
| **Constraints** | **Requirement ID** | **Constraint** |
| Indoorgml2/constraints/route-1 | No self-intersection is allowed. |

Annex A Abstract Test Suite

## Annex A (Normative) Abstract Test Suite

## Introduction

This normative annex specifies the test suite, which will be used for the conformance test of OGC IndoorGML 2.0 Part I. As OGC IndoorGML 2.0 Part I is a conceptual model, this test suite is an abstract one and its executable test suite shall depend on the implementation of OGC IndoorGML 2.0 Part I, more precisely the encoding system of OGC IndoorGML 2.0 such as XML, JSON, or SQL.

## General Tests

Since OGC IndoorGML 2 is based on ISO 19103:2015, ISO 19107:2019, ISO 19109:2015, ISO 19110:2016, and ISO 19111:2019, the abstract tests of these precedent standards shall be applied to OGC IndoorGML 2.

## Cardinalities

|  |  |
| --- | --- |
| **ID** | Indoorgml2/ate/cardinalities |
| **Requirement** | The cardinalities defined in the UML diagrams in the core and navigation modules shall be correctly applied to any implementation of IndoorGML 2. |
| **Test Method** | Manual or automated inspection |

## Properties

|  |  |
| --- | --- |
| **ID** | Indoorgml2/ate/properties |
| **Requirement** | The properties of classes defined in the UML diagrams in the core and navigation modules shall be correctly applied to any implementation of IndoorGML 2. |
| **Test Method** | Manual or automated inspection |

## Code List

|  |  |
| --- | --- |
| **ID** | Indoorgml2/ate/codelist |
| **Requirement** | The value of class properties shall be in the code list if the value type is defined as an enumeration in the UML diagrams in the core and navigation modules of IndoorGML 2. |
| **Test Method** | Manual or automated inspection |

## Class ThematicLayer

|  |  |
| --- | --- |
| **ID** | Indoorgml2/ate/thematiclayer |
| **Requirement** | Indoorgml2/constraints/thematiclayer-1 |
| **Test Method** | Manual or automated inspection |

## Class CellSpace

|  |  |
| --- | --- |
| **ID** | Indoorgml2/ate/cellspace |
| **Requirement** | Indoorgml2/constraints/cellspace-1 |
| **Test Method** | Automated inspection by geometric computation |

## Class CellBoundary

|  |  |
| --- | --- |
| **ID** | Indoorgml2/ate/cellboundary |
| **Requirements** | Indoorgml2/constraints/cellboundary-1,  Indoorgml2/constraints/cellboundary-2 |
| **Test Method** | Automated inspection by geometric computation |

## Class Node

|  |  |
| --- | --- |
| **ID** | Indoorgml2/ate/node |
| **Requirement** | Indoorgml2/constraints/node-1 |
| **Test Method** | Automated inspection by geometric computation |

## Class Edge

|  |  |
| --- | --- |
| **ID** | Indoorgml2/ate/edge |
| **Requirement** | Indoorgml2/constraints/edge-1,  Indoorgml2/constraints/edge-2 |
| **Test Method** | Automated inspection by geometric computation |

## Class ObjectSpace

|  |  |
| --- | --- |
| **ID** | Indoorgml2/ate/objectspace |
| **Requirement** | Indoorgml2/constraints/objectspace-1,  Indoorgml2/constraints/objectspace -2 |
| **Test Method** | Automated inspection by geometric computation |

## Class Route

|  |  |
| --- | --- |
| **ID** | Indoorgml2/ate/route |
| **Requirement** | Indoorgml2/constraints/route-1 |
| **Test Method** | Automated inspection by geometric computation |

## Annex B Bibliography

* Alattas A, S. Zlatanova, P. van Oosterom P, E. Chatzinikolaou, C. Lemmen and K-J. Li (2017) Supporting Indoor Navigation Using Access Rights to Spaces Based on Combined Use of IndoorGML and LADM Models, ISPRS International Journal of Geoinformation, vol. 6 (12), pp. 384
* Diakité, A. A. and S. Zlatanova (2018). Spatial subdivision of complex indoor environments for 3D indoor navigation. International Journal of Geographical Information Science, 213-235.
* Egenhofer M.J. (1989) A formal definition of binary topological relationships. In: Litwin W., Schek HJ. (eds) Foundations of Data Organization and Algorithms. FODO 1989. Lecture Notes in Computer Science, vol 367. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/3-540-51295-0_148>
* Gröger, G., B. George (2012) Geometry and Topology, in Kresse and Danko (eds) Handbook of Geographic Information, Springer 10.1007/978-3-540-72680-7, pp. 303-321
* ISO (2012) ISO 19152:2012, Geographic Information – Land Administration Domain Model (LADM)
* ISO (2018) ISO 16739-1:2018, Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema,
* ISO. (2019). ISO 19107:2019, Geographic information - Spatial Schema. Retrieved 02 10, 2020, from International Organization for Standardization: https://www.iso.org/standard/66175.html
* Morris, S. (2019). Topology Without Tears. Retrieved 02 10, 2020, from Topology Without Tears Online Book: http://www.topologywithouttears.net/topbook.pdf
* Munkres, J. R. (2018). Elements of algebraic topology. CRC Press.
* Lee, J. (2004). A spatial access-oriented implementation of a 3D GIS topological data model for urban entities. GeoInformatica, 237-264.
* OGC (2012) City Geography Markup Language (CityGML) Encoding Standard, Open Geospatial Consortium
* OGC (2015) OGC KML 2.3, Encoding Standard, Open Geospatial Consortium
* OGC (2021) Indoor Mapping Data Format, OGC community standard
* OGC. (2009). The Specification Model – A Standard for Modular specifications. Open Geospatial Consortium.
* OGC. (2015). IndoorGML. Retrieved 02 06, 2020, from https://www.opengeospatial.org/standards/indoorgml
* Princeton University. (2010). About Wordnet. (Princeton University) Retrieved 02 06, 2020, from https://wordnet.princeton.edu/
* Yan, J., Diakité, A. A., & Zlatanova, S. (2019). A generic space definition framework to support seamless indoor/outdoor navigation systems. Transactions in GIS, 23(6), 1273-1295.
* Zlatanova, S., Yan, J., Wang, Y., Diakité, A., Isikdag, U., Sithole, G., & Barton, J. (2020). Spaces in Spatial Science and Urban Applications—State of the Art Review. ISPRS International Journal of Geo-Information, 9(1), 58.

1. <https://portal.ogc.org/public_ogc/directives/directives.php> [↑](#footnote-ref-1)