Open Geospatial Consortium

Submission Date: <2024-mm-dd>

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 22-045r4

Version: 1.0

Category: OGC® Standard

Editors: Sisi Zlatanova, Abdoulaye Diakité, Taehoon Kim, and Ki-Joune Li

OGC IndoorGML Part I – Conceptual Model

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.

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.

Document type:	OGC® Standard
Document subtype:	if applicable
Document stage:	Draft
Document language:	English

License Agreement

Contents

II. KEYWORDS 7

III. PREFACE 7

IV. SECURITY CONSIDERATIONS 8

V. SUBMITTING ORGANIZATIONS 8

VI. SUBMISSION CONTACT POINTS 8

1 Scope 9

2 Conformance 9

3 References 9

4 Terms and Definitions 10

4.1 Indoor Space 10

4.2 Cellular Space 11

4.3 Graph 11

4.4 Adjacency Graph 11

4.5 Connectivity Graph 11

4.6 Logical Network 11

4.7 Geometric Network 11

4.8 Multi-Layered Space Model 11

5 Conventions 11

5.2. UML Notation 12

5.4 Identifiers 13

6 OVERVIEW OF INDOORGML 13

6.1 Motivation for developing the IndoorGML Standard 14

6.2 Modularization 15

7 GENERAL INDOORGML CONCEPTS 15
7.1 Space 16
7.2 Cellular space 16
7.2.1 Geometry 18
7.2.2 Topology 19
7.2.3 Subdivision, aggregation, and selection 19
7.3 Poincaré Duality 20
7.4 Structured space model 22
7.5 Semantics 23
7.6 Thematic layers 23
7.6.1 Multiple-Layered Space representation 25
7.6.2 Inter-Layer Relations 25
8 Data model 27
8.1 IndoorGML Core Module 27
8.1.1 CellSpace 30
8.1.2 CellBoundary 31
8.1.3 PrimalSpaceLayer 33
8.1.4 Node 33
8.1.5 Edge 34
8.1.6 DualSpaceLayer 35
8.1.7 InterLayerConnection 36
8.1.8 ThematicLayer 37
8.2 Navigation extension module 39
8.2.1 NavigableSpace 41
8.2.2 GeneralSpace 42
8.2.3 TransferSpace 43
8.2.4 NavigableBoundary 43
8.2.5 NonNavigableSpace 44
8.2.6 ObjectSpace 44
8.2.7 NonNavigableBoundary 45

8.2.8 Route 45

8.3 Requirements 46

9 Data dictionary and requirements 48

9.1 Feature Types in Core Module 48

9.1.1 IndoorFeatures 48

9.1.2 ThematicLayer 48

9.1.3 PrimalSpaceLayer 48

9.1.4 CellSpace 49

9.1.5 CellBoundary 49

9.1.6 DualSpaceLayer 50

9.1.7 Node 51

9.1.8 Edge 51

9.1.9 InterLayerConnection 52

9.2 Feature Types in Navigation Module 52

9.2.1 NavigableSpace 52

9.2.2 NonNavigableSpace 53

9.2.3 GeneralSpace 53

9.2.4 TransferSpace 53

9.2.5 ObjectSpace 53

9.2.6 NavigableBoundary 54

9.2.7 NonNavigableBoundary 54

9.2.8 Route 54

A Annex A (Normative) Abstract Test Suite 55

A.1 Introduction 55

A.2 General Tests 55

A.3 UML Data Model 55

A.4 Class ThematicLayer 55

A.5 Class CellSpace 55

A.6 Class CellBoundary 56

A.7 Class Node 56

A.8 Class Edge 56

A.9 Class InterLayerConnection 56

A.10 Class ObjectSpace 57

B Annex B Bibliography 58

I. ABSTRACT

The OGC IndoorGML Part I – Conceptual Model Standard (this Standard) specifies an UML model for indoor information. Part 2 of the standard specifies technical implementation schemas in GML, SQL and JSON. While there are several standards supporting 3D modelling concepts such as CityGML, KML, IFC, LADM and IMDF that deal with interiors of buildings from geometric, cartographic, and semantic viewpoints, IndoorGML focuses on modeling indoor spaces and their neighborhood relationships to support indoor location-based services. This document describes the conceptual model (Part 1) of IndoorGML that 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 the OGC IndoorGML Standard is to represent and support interoperable 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), and IFC (ISO,2018) have been published to describe 3D geometry and semantics of building features. However, these standards are not readily appropriate to derive spaces and their topological relationships. The OGC IMDF Community Standard (OGC, 2021) provides a comprehensive model to compute path(s) between features located on a map, but the derived network is application specific. The IndoorGML Standard aims to provide a unified, standardized and flexible approach for indoor spatial information required for space-graph based applications such as indoor navigation.

Version 2.0 of the IndoorGML Part 1 - Conceptual Model 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. While the core data model, as its name suggests, is the main conceptual model that IndoorGML is built upon, the navigation model extends it to support navigation applications. This allows illustrating the practical strength of the core model as well as how other application extensions could be built around it.

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, IFC, LADM, and IMDF. In this respect, IndoorGML is a complementary standard to CityGML, 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
- CityGeometrix
- National Institute of Advanced Industrial Science and Technology (AIST)

The initial concepts defined in this Standard were 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	CityGeometrix	abdou at citygeometrix.com

Jeremy Morley	Ordnance Survey	Jeremy.Morley at os.uk
Taehoon Kim	National Institute of Advanced Industrial Science and Technology	kim.taehoon at aist.jp

1 Scope

The OGC IndoorGML Standard is for the representation and interoperable exchange of indoor navigation network models. IndoorGML Part 1 is a conceptual schema documented using UML and Part 2 is an implementation schema of the Geography Markup Language version 3.2.1.

IndoorGML Part 1 establishes a common schema for indoor space model and indoor navigation applications. Part 1 models topology and semantics of indoor spaces, which are needed for the components of navigation networks. The IndoorGML Standard defines 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.

2 Conformance

Conformance targets of IndoorGML Part 1 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 requirements and schema document implementations SHALL:

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 (ATS) 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 site[1].

3 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 8601-1:2019, Date and time – Representation for information interchange – Part 1: Basic rules, , 2019. ISO (2019).

ISO: ISO 19103:2015, Geographic Information – Conceptual Schema Language, 2015. ISO (2015).

ISO: ISO 19105:2022, Geographic information – Conformance and testing, 2022. ISO (2022).

ISO: ISO 19107:2019, Geographic Information – Spatial Schema, 2019. ISO (2019).

ISO: ISO 19109:2015, Geographic Information – Rules for Application Schemas, 2015. ISO (2015).

ISO: ISO 19111:2019, Geographic information – Referencing by coordinates, 2019. ISO (2019).

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

ISO/TS: ISO/TS 19139:2019, Geographic Information –XML schema implementation – Part 1: Encoding rules 2019. ISO and TS (2019).

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

4 Terms and Definitions

This Standard used the terms defined in Policy Directive 49[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.

4.1 Indoor Space

A space within one or multiple buildings.

[1] https://portal.ogc.org/public_ogc/directives/directives.php

4.2 Cellular Space

A cellular space S is a set of identifiable cells c_i grouped according to theme T and is noted S^T is $S^T = \{c_1, c_2, \ldots, c_n\}$

4.3 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.

4.4 Adjacency Graph

A graph G_{adj} (V, E_{adj}) where V is a set of nodes representing cells and E_{adj} is the set of edges indicating the adjacency relationship.

4.5 Connectivity Graph

A graph $\mathrm{G}_{\mathrm{con}}(V, E_{\mathrm{con}})$ where V is a set of nodes representing cells in indoor space and E_{con} is the set of edges indicating the connectivity relationship.

4.6 Logical Network

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

4.7 Geometric Network

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

4.8 Multi-Layered Space Model

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

5 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.

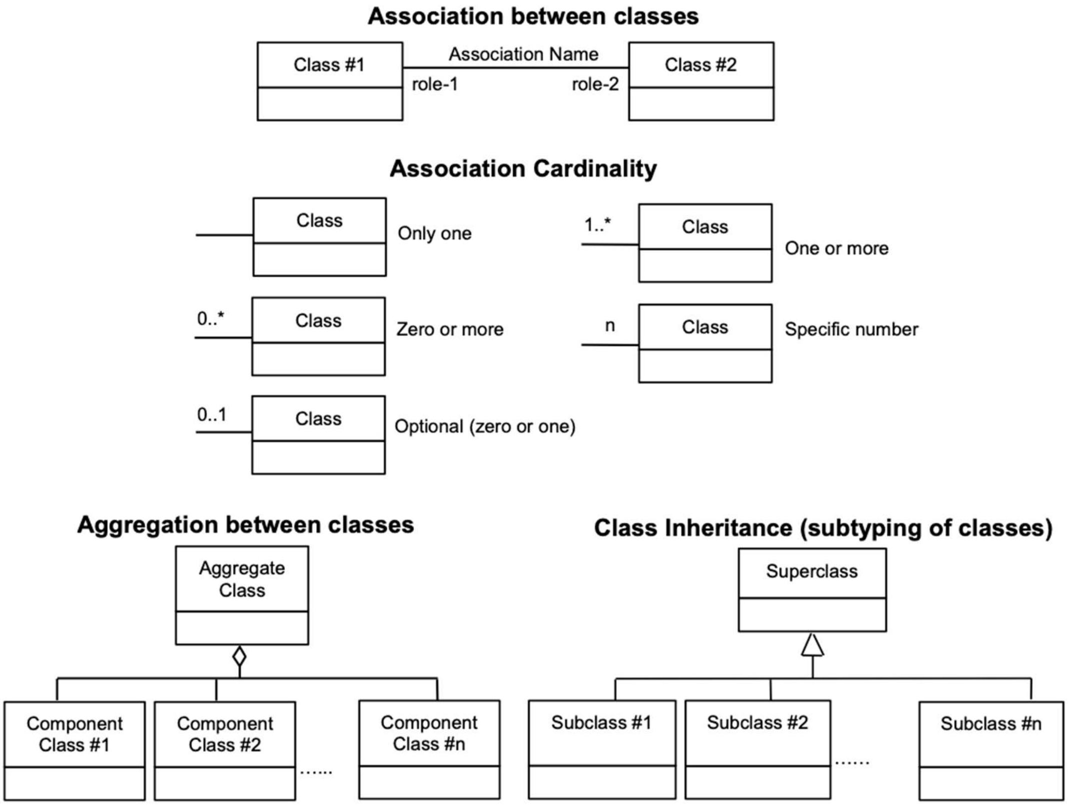

Figure 1: UML Notations

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.

5.4 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.

6 OVERVIEW OF INDOORGML

The IndoorGML Conceptual Model is designed to support applications developers in providing Location-based services applications. Figure 2 illustrates the place of IndoorGML in an ecosystem of standards, models, and files formats and end-user applications. IndoorGML provides simplified yet standardized 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 support protecting some sensitive building information.

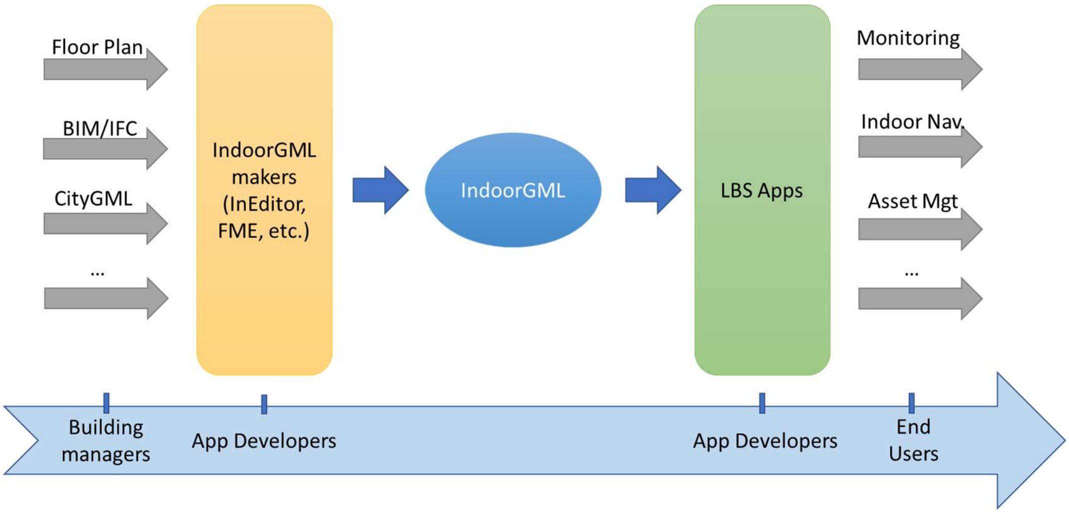

Figure 2: IndoorGML in the overall application development ecosystem

6.1 Motivation for developing the IndoorGML Standard

Indoor environments differ from outdoor in many aspects. Indoor spaces have fewer 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. Indoor 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 an extent that some spaces are not accessible for certain type of users. This has led to the existence of a variety of approaches for modelling indoor environments and providing services. Therefore, well-known concepts, data models, and standards need to 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 applications 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.

The IndoorGML Standard is intended to provide a unified modelling approach that is necessary to support indoor applications using information from those two categories. The leading concepts in the IndoorGML Conceptual Model 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 properties, 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 organized in a UML class diagram (Section 8), from which the implementation schemas for GML are provided.

6.2 Modularization

Following the OGC ModSpec guidance (OGC, 2009)￼, IndoorGML Part 1 – Conceptual Model is organised into a Core module and Extension modules that have one or more mandatory dependencies on the Core (see Figure 3). The IndoorGML Core module defines generic aspects of Primal-Dual spaces (see below) and each extension module introduces thematic semantics for a specific application. IndoorGML 2.0 Part 1 contains one extension named *Navigation*. IndoorGML 2.0 Part 2 provides the implementation of IndoorGML 2.0 Part 1 Conceptual model in GML, JSON, and SQL.

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.

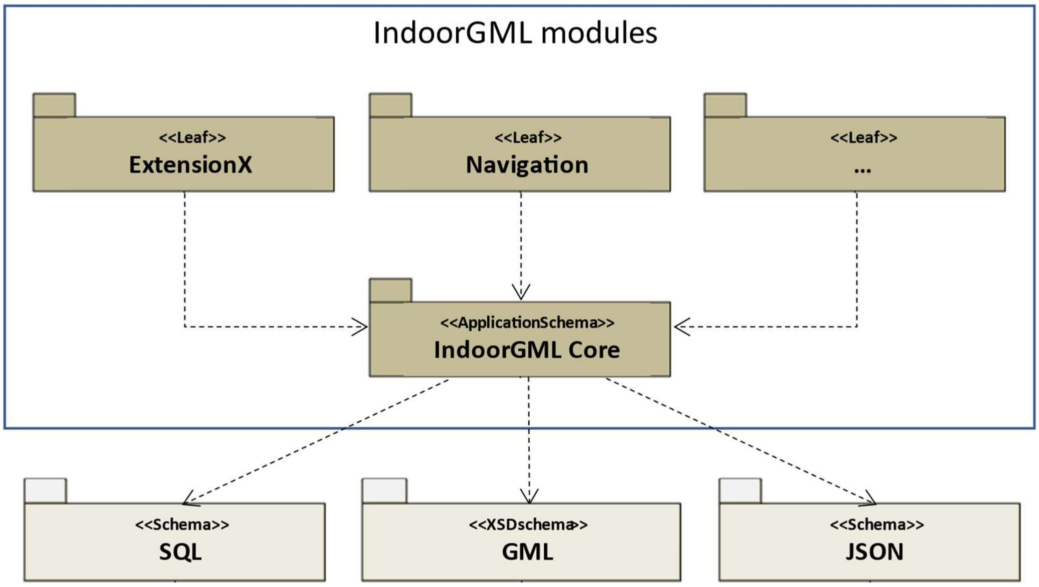

Figure 3: Modular organisation of IndoorGML

7 GENERAL INDOORGML CONCEPTS

IndoorGML is a space-centred standard. As such, it focuses on the three main types of information of spaces (2D or 3D): geometry, topology and semantic. 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 the Primal-Dual model that defines the core part of the IndoorGML Conceptual Model. The semantic extension mechanism, as the name suggests, supports adding more details to the basic semantics of the core module. Thematic layering mechanism supports organizing an IndoorGML model as a collection of layers with different themes. Those concepts are elaborated in the following subsections.

7.1 Space

The notion of space is widely explored in spatial science and urban applications in general (Zlatanova, et al., 2020). Among the 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 that an 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.

Indoor space is commonly perceived as a space within a building. It incorporates architectural components such as walls, slabs, and doors, and furniture such as chairs, tables, desks, and the remaining empty spaces as in rooms, corridors, halls, etc. IndoorGML 2.0 Conceptual Model 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 categorized 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, the IndoorGML Conceptual Model can account for all types of space within the built environment, if they can be represented with the IndoorGML Cellular space concept (below).

7.2 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 S^T is defined as follows:

$S^T = \{c_1, c_2, \ldots, c_n\}$

where c_i 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.

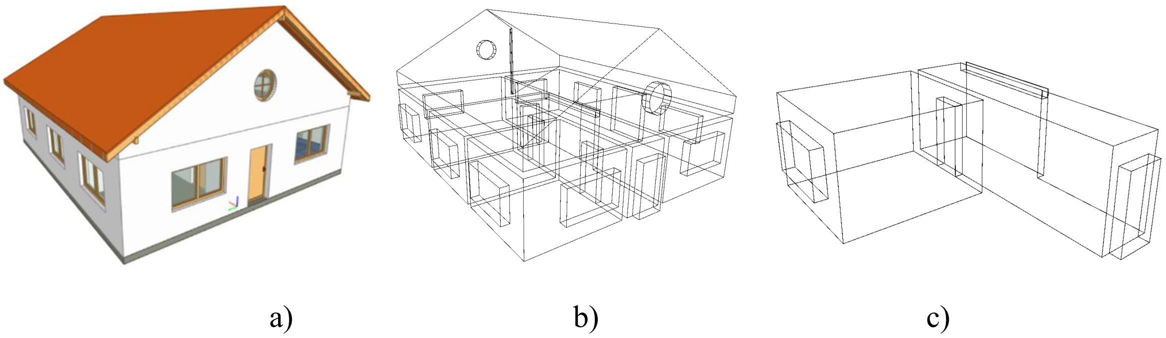

Figure 4: A building (a) and the corresponding cellural space, containing all empty cells (b) and corresponding cells of a room, a corridor and their openings (c).

Within a cellular space, only the adjacency relationship is allowed between cells: No overlap may occur. Overlapping cells must be organized in a new cellular space. A cellular space may be incomplete coverage: There can be gaps between cells (Figure 5).

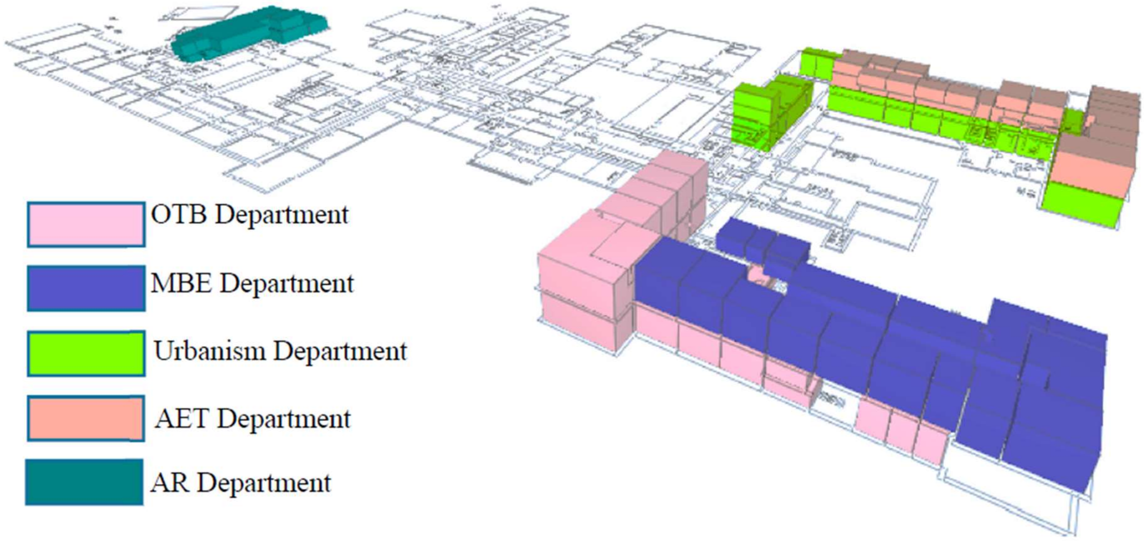

Figure 5: Cellular space containg disconected cells, i.e. all offices in an university building (Allatas et al 2017)

In the IndoorGML Conceptual Model, 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.

7.2.1 Geometry

Every cell defining an item in indoor space owns a shape, size, and location that can be collected and modelled. Cell 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 polygon or rectangle. Such an approach can be applied when considering highly irregular shapes such as furniture. Geometric information can be included in an IndoorGML implementation instance either directly or via an external link. Geometry of cells can be omitted as well.

Cell geometry is defined in 2 dimensional or 3-dimensional Euclidean space and provides the means for the quantitative description of the spatial characteristics of cell. Metrics are defined as in (Morris, 2019). In the IndoorGML Conceptual Model, cells are modelled as features and conform with requirements as specified in ISO 19107 (Spatial Schema) (ISO, 2019). ISO 19107 provides conceptual schemas to describe and model real world objects as features.

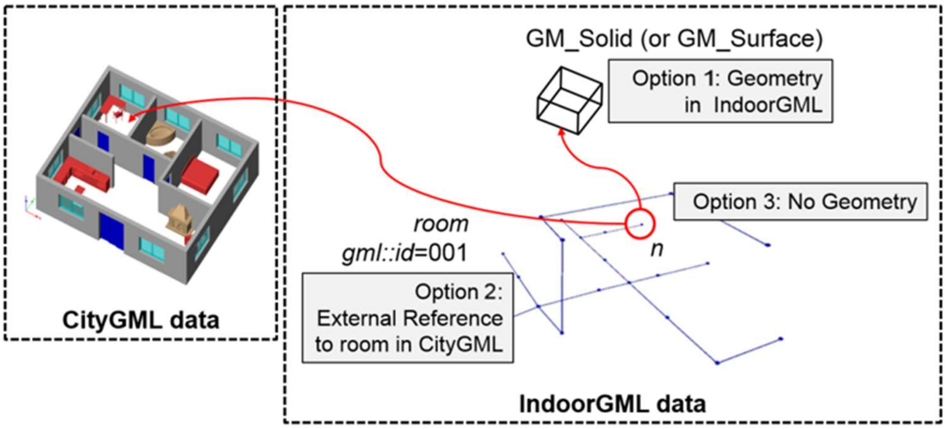

Figure 6: Three options to represent geometry in IndoorGML

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 a cell may be included within an IndoorGML document. As defined in ISO 19107 this is a GM_Solid in 3D space and GM_Surface in 2D space. 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 data sets is some encoding/format 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 these corresponding objects in other datasets.
3. No Geometry (Option 3): No geometric information is included in IndoorGML. This means that the shape, extent 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.

7.2.2 Topology

Beside the geometry of a cellular space, cells can be represented in a topological model, which specifies the relationships between points, lines, polygons and solids constructing the geometry (Gröger and George, 2012). Such topological models are dedicated to representing spatial relationships. As such, their shape and size are not described (Egenhofer 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 space, which are defined 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 8.1.3

7.2.3 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 be applied.

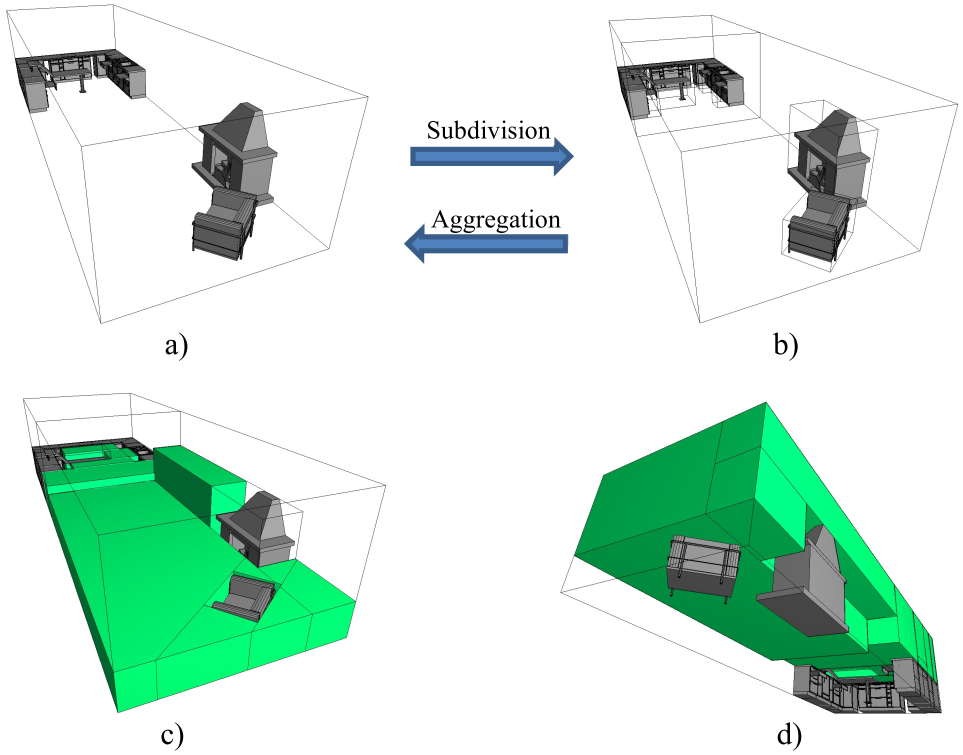

Figure 7: (a) A furnished indoor space. (b) Subdivision of the indoor space into two separate rooms with exclusion of furnishing elements' spaces. (c) Selection of specific CellSpaces (green) suitable for walking and rolling. (d) CellSpaces (green) suitable for flying.

As illustrated in Figure 7, the subdivision is performed by splitting the original cells into several subspaces, according to the function of cell. For example, in Figure 7(b), the indoor space is subdivided into several cells according to their functions (e.g. as in Figure 7(a)) into 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 navigation applications, subdivisions may be required due to:

- Geometry simplification such as working with spaces that have only convex shapes.
- Increase of granularity such as in for improving the localisation of people and items.
- Need to identify specific functional/perception spaces such as waiting or smoking areas.
- Defining free spaces such as spaces free of obstacles.

The aggregation process is the reverse of the subdivision process. An aggregation process results in subspaces being merged instead of being split. Therefore, the merging of all subspaces shown in Figure 7(a) allows retrieving the original cell spaces. 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, such as individual toilets or service areas in a building.
- There are CellSpaces, which are not accessible for specific users, such as restricted areas at hospitals and airports.

Finally, it is possible to discriminate CellSpaces of interest from the rest. Figure 7(c) and (d) illustrate a scenario where only CellSpaces that can support certain type of locomotion modes 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 such as selecting 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 such as. selecting only common spaces for someone visiting a public building.
- Eliminating spaces of danger such as in emergency cases, select only spaces which are still safe for users to be in.

7.3 Poincaré Duality

Topological relations between cells are crucial in the IndoorGML model. They allow establishing links between cells in the same or different thematic layers. This is critical information for several applications such as navigation, which relies on connectivity networks. 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. The Poincaré duality 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 in a 3D topological model (Lee, 2004).

The Poincaré duality refers to two spaces: *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, such as rooms within a building, are mapped to nodes (0D object) in dual space. A 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 in Dual space represent abstractions of cells and their adjacency relationships in Primal space.

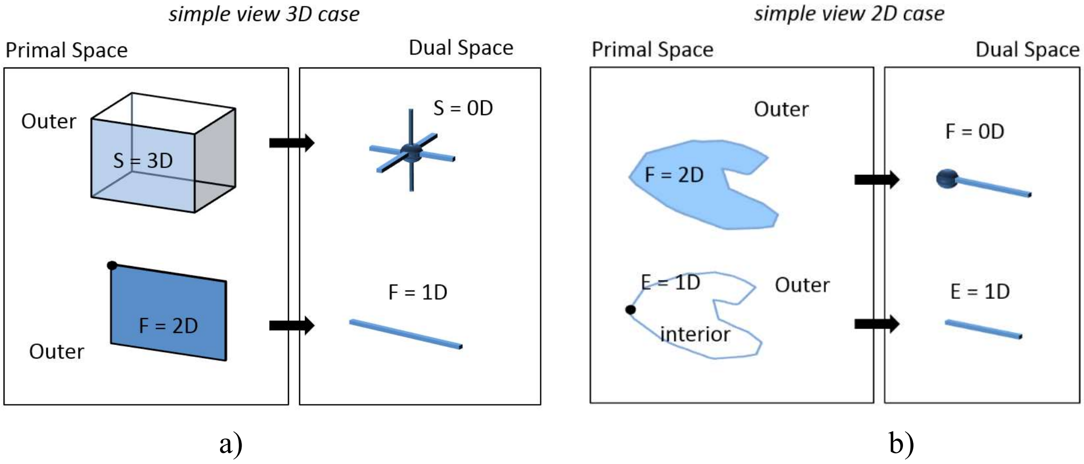

Figure 8: Principles of Poincaré duality. 3D Primal space case (a) and 2D case (b). (Mathematical definition of Poincaré duality in (Munkres, 2018))

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 the IndoorGML model since they are not considered as cells in most applications. However, 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 G_{adj} is defined as follows:

$$G_{adj} = (V, E_{adj})$$

where V and E_{adj} are sets of nodes and edges in dual space mapped from cells and surfaces in

3D primal space, respectively. The connectivity graph G_{con} is a subset of the adjacency graph and represents 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. Connectivity graph is defined as:

$$G_{con} = (V, E_{con})$$

where V and E_{con} 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.

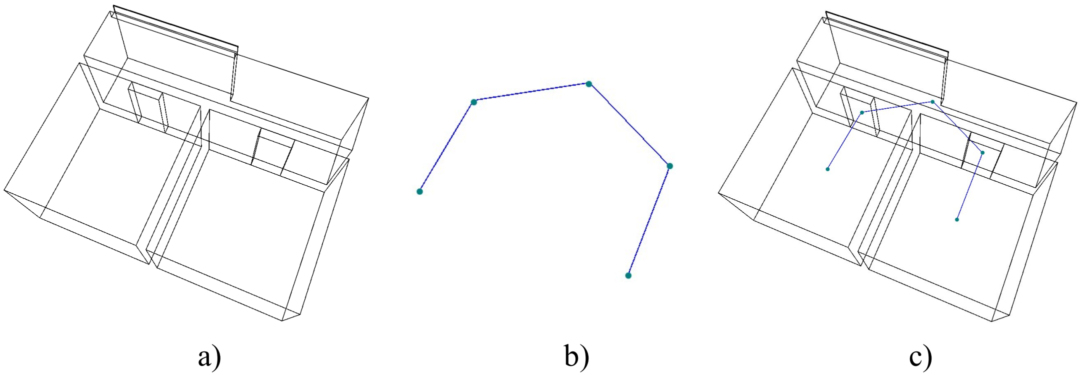

Figure 9: Poincaré duality on 3D cells of a building (a); Corresponding adjacency graph in the dual space (b); Combined primal and dual space view (c).

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 network holds geometry for nodes and edges.

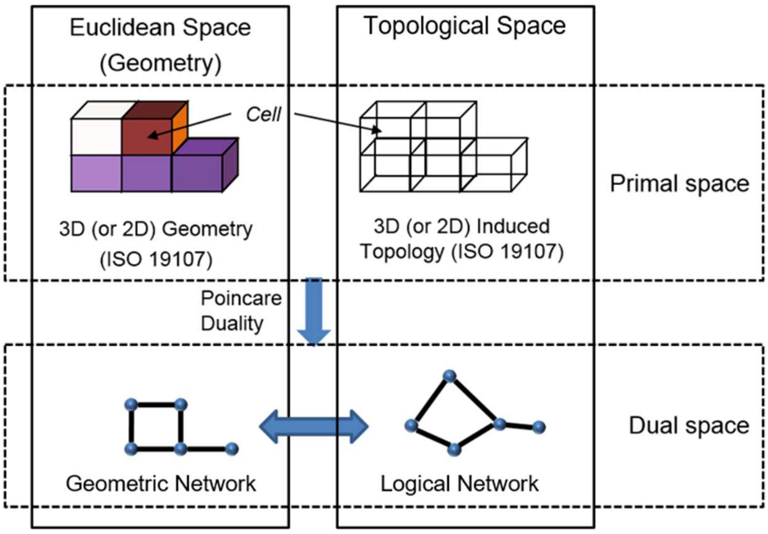
Euclidean Space (Geometry)
Topological Space
Cell
3D (or 2D) Geometry (ISO 19107)
3D (or 2D) Induced Topology (ISO 19107)
Primal space
Poincare Duality
Geometric Network
Logical Network
Dual space

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

7.4 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 the Geometric network or the 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. An IndoorGML encoding 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. An IndoorGML encoding may then contain 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 an IndoorGML encoding containing Geometry and Logical Network or Topology and Logical Network.

7.5 Semantics

The IndoorGML model contains the semantic for the Primal and Dual spaces of the core module. The semantics of the core model are generic for all applications. It specifies only some characteristics such as name, level, and Point of Interest (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 explained in Section 8.2. Every cell is further classified according to the semantics introduced by the extension module. The IndoorGML 2.0 Model defines semantics for Indoor navigation that are provided within the Navigation extension module. The semantics, defined in the Navigation extension module, are intended for two purposes to: 1) provide a classification of a cell, and 2) determine adjacency relationships that ensure connectivity between cells. Semantics thus supports defining 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 8.2). 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. These latter applications require 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.

The semantic extension mechanism enables adding more semantic on primal or dual spaces, as long as they follow the modularization principle. Cells can be organized in a hierarchical structure according to their semantics, corresponding properties, and semantic interrelations (specialization and generalization). 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 8.2

7.6 Thematic layers

A single indoor environment can be organized 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 organized 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, such as subdivision, aggregation and selection. Examples of such thematic layers are subdivision according to Wi-Fi or RFID coverage (see example below)[2]. The Navigation extension module provides additional notions for navigability and connectivity. Therefore, thematic layers that rely of these properties should include the 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 organized 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 facility (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 organized into thematic layers.

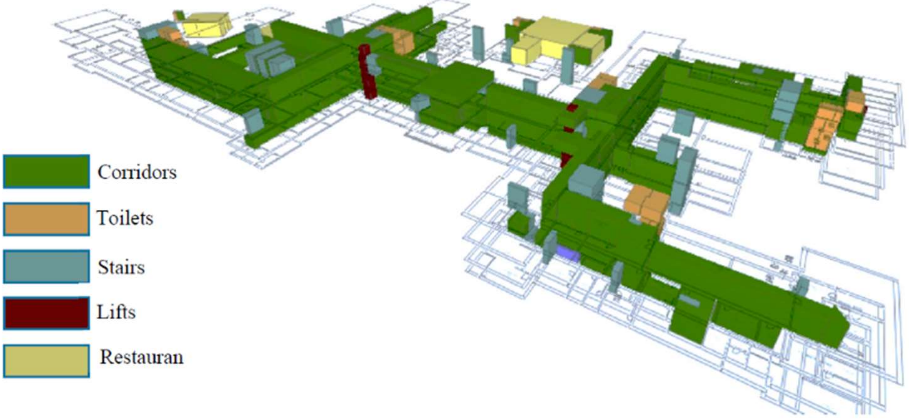

Figure 11: Cellular space for visitors (Alattas et al 2017)

In Figure 12, a physical indoor space named Topographic layer is organized according to the Navigation extension module. In addition, two thematic layers 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,

[2] While the reference to Wi-Fi and RFID as Indoor Positioning Systems (IPS) is essentially indicative, a wider family of IPS technologies known as Real-Time Locating Systems (RTLS) are increasingly being adopted. Future versions of IndoorGML will consider a better integration to such technologies through, for example, integration with existing open standards in that space such as omlox (https://omlox.com/).

although related to subdivision approaches, each form 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: Geometry, Topology, Geometric network, or Logical network.

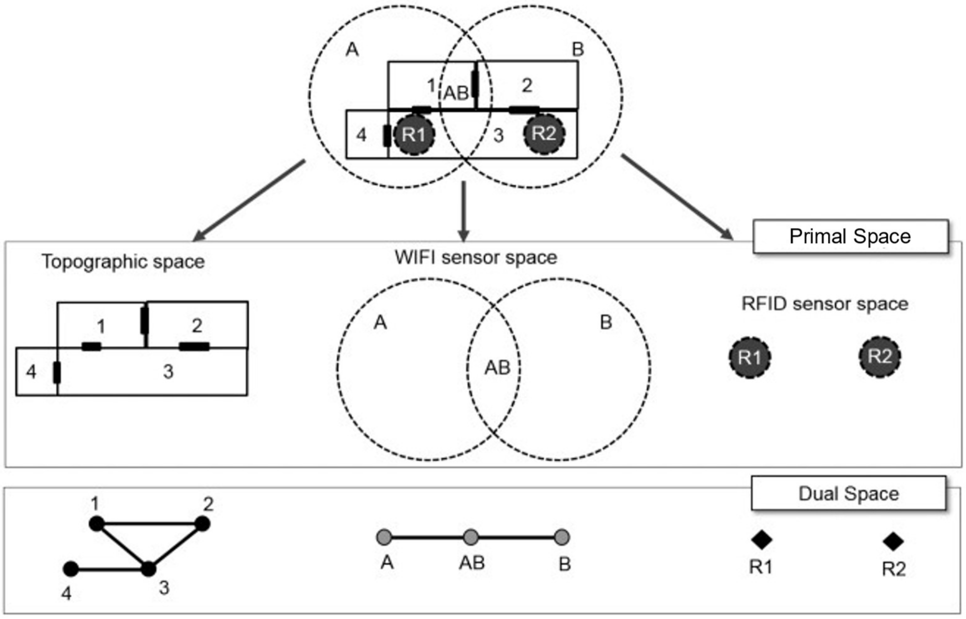

Figure 12: Three different cellural spaces for the same physical space

7.6.1 Multiple-Layered Space representation

The IndoorGML model provides mechanisms for maintaining and linking multiple Thematic layers for the 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, representing the hierarchical structure of indoor space, where each floor level is a single space layer. Another application example is indoor tracking using presence sensors, such as RFID, as shown in Figure 12. Given an indoor space represented as a Topographic layer and RFID sensor coverage layer respectively, the movement of a mobile object can be deduced using a RFID tag by the sequence of RFID coverage cells and corresponding inter-layer space edges.

7.6.2 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: *contains, within, covers, coveredBy*, *overlaps and equals* between cells in the Primal space and nodes in Dual space.

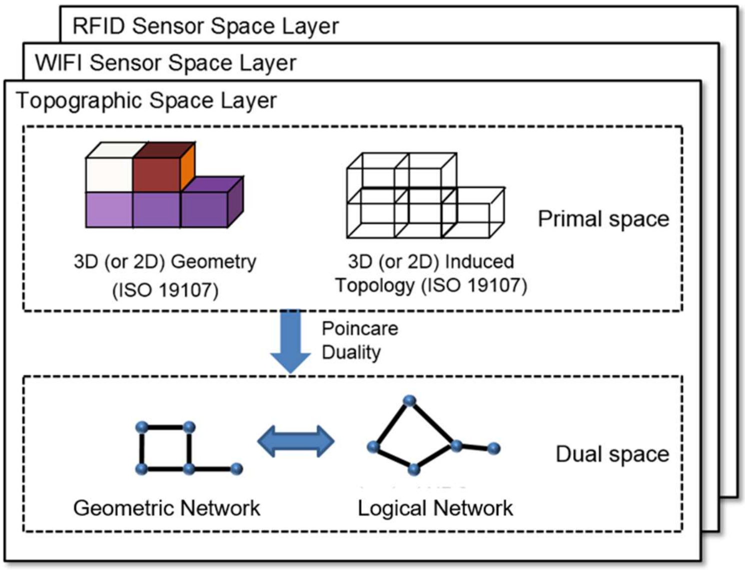

Figure 13: Corresponding Primal and Dual spaces of different thematic layers.

As illustrated in Figure 13, there are three space layers. 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). 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., adjacent rooms connected by a door). In the sensor space, the graph has a slightly different structure. The nodes represent again the cells (e.g., the entire coverage space of a Wi-Fi transmitter); 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 illustrates 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 the IndoorGML 2.0 model is the possibility of representing an inter-layer connection between two primal spaces. This is illustrated in Figure 15 where the inter-layer mechanism is used to represent a furnished room with a combination of two layers: One describing solely the cells of the room and openings (Figure 15(b)) and one describing the furniture CellSpaces (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 supports describing complex scenes while respecting the non-overlapping constraint of Poincare duality.

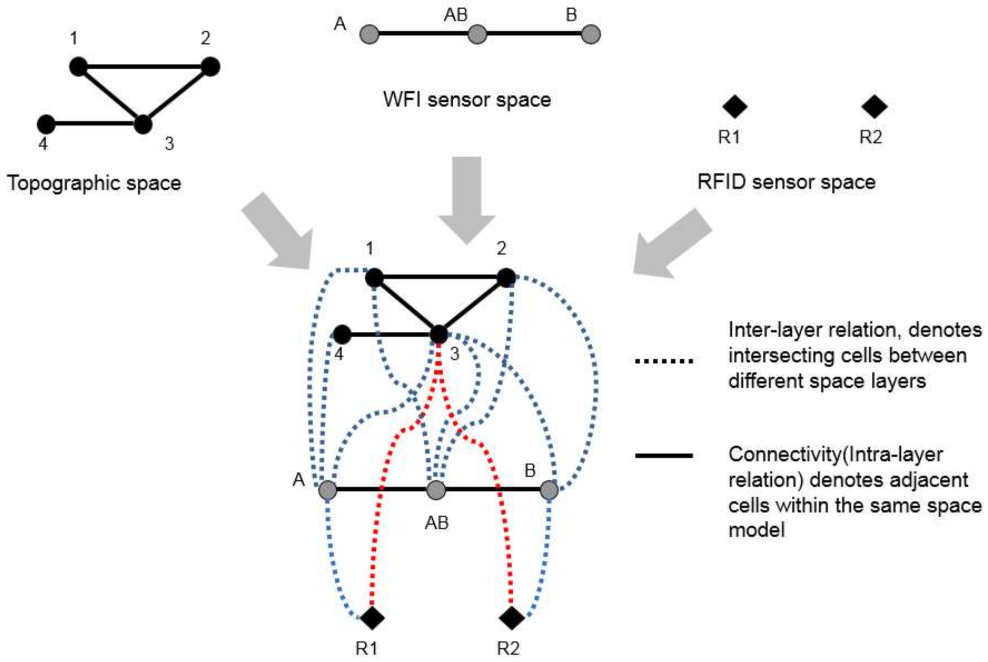
A
AB
B
WFI sensor space
1
2
4
3
Topographic space
R1
R2
RFID sensor space
1
2
4
3
A
AB
B
R1
R2
Inter-layer relation, denotes intersecting cells between different space layers
Connectivity(Intra-layer relation) denotes adjacent cells within the same space model

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

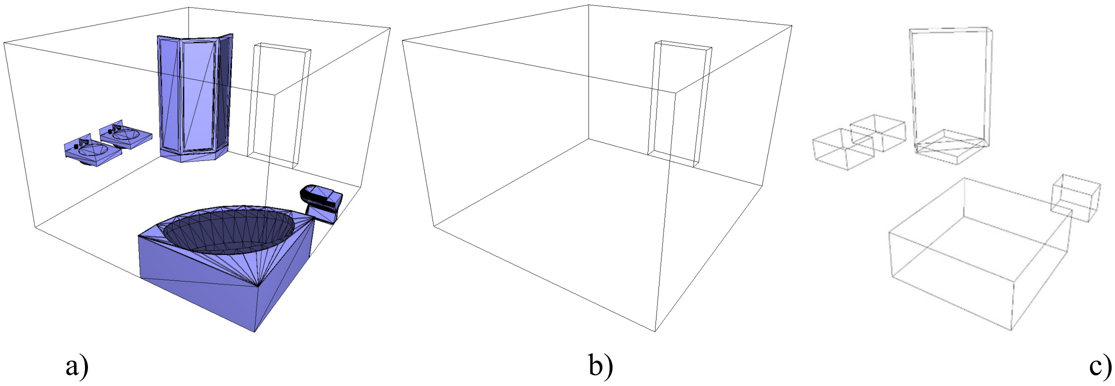

Figure 15: Inter-layer connection between two primal spaces. (a) furnished room. (b) cells of the room and door only. (c) cells of furnishing elements only represented by minmax boxes.

8 Data model

This section presents the IndoorGML conceptual data model as UML class diagrams.

8.1 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).

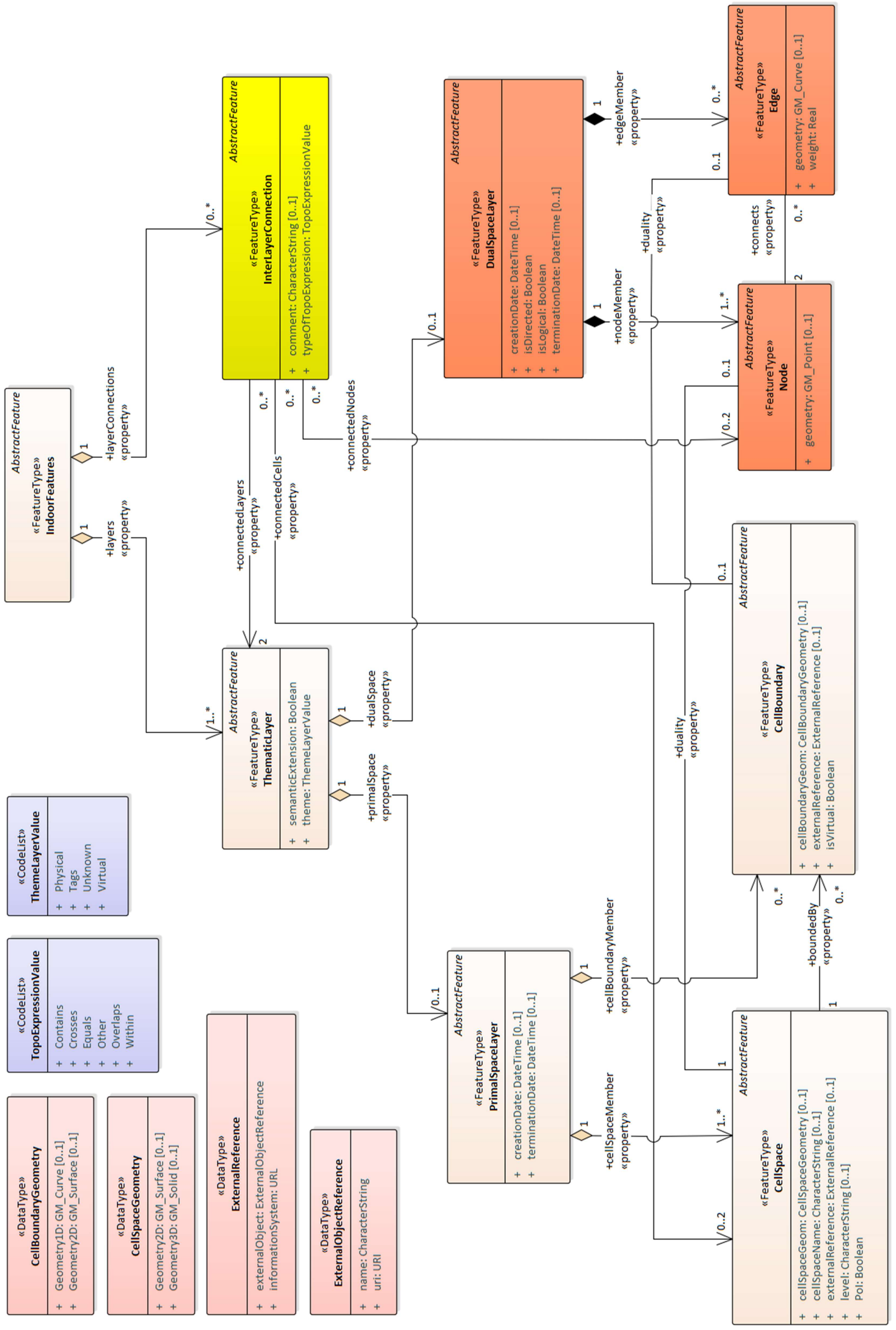
«DataType»
CellBoundaryGeometry
+ Geometry1D: GM_Curve [0..1]
+ Geometry2D: GM_Surface [0..1]
«DataType»
CellSpaceGeometry
+ Geometry2D: GM_Surface [0..1]
+ Geometry3D: GM_Solid [0..1]
«DataType»
ExternalReference
+ externalObject: ExternalObjectReference
+ informationSystem: URL
«DataType»
ExternalObjectReference
+ name: CharacterString
+ uri: URI
«CodeList»
TopoExpressionValue
+ Contains
+ Crosses
+ Equals
+ Other
+ Overlaps
+ Within
«CodeList»
ThemeLayerValue
+ Physical
+ Tags
+ Unknown
+ Virtual
AbstractFeature
«FeatureType»
IndoorFeatures
AbstractFeature
«FeatureType»
InterLayerConnection
+ comment: CharacterString [0..1]
+ typeOfTopoExpression: TopoExpressionValue
AbstractFeature
«FeatureType»
ThematicLayer
+ semanticExtension: Boolean
+ theme: ThemeLayerValue
AbstractFeature
«FeatureType»
DualSpaceLayer
+ creationDate: DateTime [0..1]
+ isDirected: Boolean
+ isLogical: Boolean
+ terminationDate: DateTime [0..1]
AbstractFeature
«FeatureType»
PrimalSpaceLayer
+ creationDate: DateTime [0..1]
+ terminationDate: DateTime [0..1]
AbstractFeature
«FeatureType»
Edge
+ geometry: GM_Curve [0..1]
+ weight: Real
AbstractFeature
«FeatureType»
Node
+ geometry: GM_Point [0..1]
AbstractFeature
«FeatureType»
CellBoundary
+ cellBoundaryGeom: CellBoundaryGeometry [0..1]
+ externalReference: ExternalReference [0..1]
+ isVirtual: Boolean
AbstractFeature
«FeatureType»
CellSpace
+ cellSpaceGeom: CellSpaceGeometry [0..1]
+ cellSpaceName: CharacterString [0..1]
+ externalReference: ExternalReference [0..1]
+ level: CharacterString [0..1]
+ PoI: Boolean
+layerConnections
«property»
1
0..*
+layers
«property»
1
1..*
+connectedLayers
«property»
0..*
2
+connectedCells
«property»
0..*
0..2
+connectedNodes
«property»
0..*
0..2
+dualSpace
«property»
1
0..1
+primalSpace
«property»
1
0..1
+edgeMember
«property»
1
0..*
+nodeMember
«property»
1
1..*
+connects
«property»
0..*
2
+duality
«property»
0..1
0..1
+duality
«property»
1
0..1
+cellBoundaryMember
«property»
1
0..*
+cellSpaceMember
«property»
1
1..*
+boundedBy
«property»
1
0..*

Figure 16: UML diagram of the Core module.

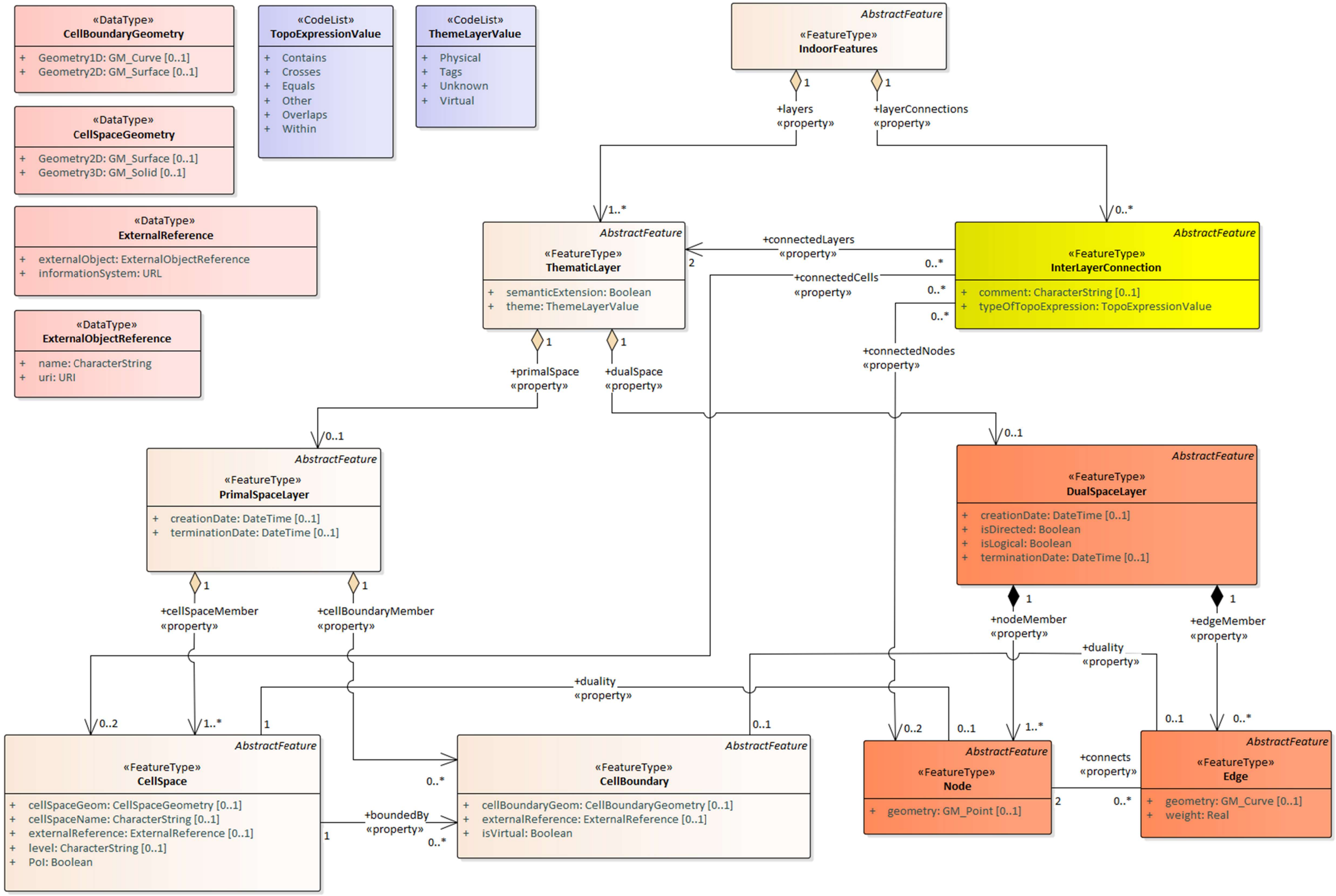

Figure 16, the UML diagram illustrates all the classes associated with those three parts. In the following, the classes are introduced and the data types that they invoke in their properties are detailed.

8.1.1 CellSpace

CellSpace is a core module class for representing the indoor environment in terms of cellular space. CellSpace is a mandatory class to have a valid IndoorGML2.0 encoding. It contains the following properties :

- *cellSpaceGeom* (*CellSpaceGeometryType*)
- *externalReference (url)*
- *level (string)*
- *name (string)*
- *PoI (Boolean)*

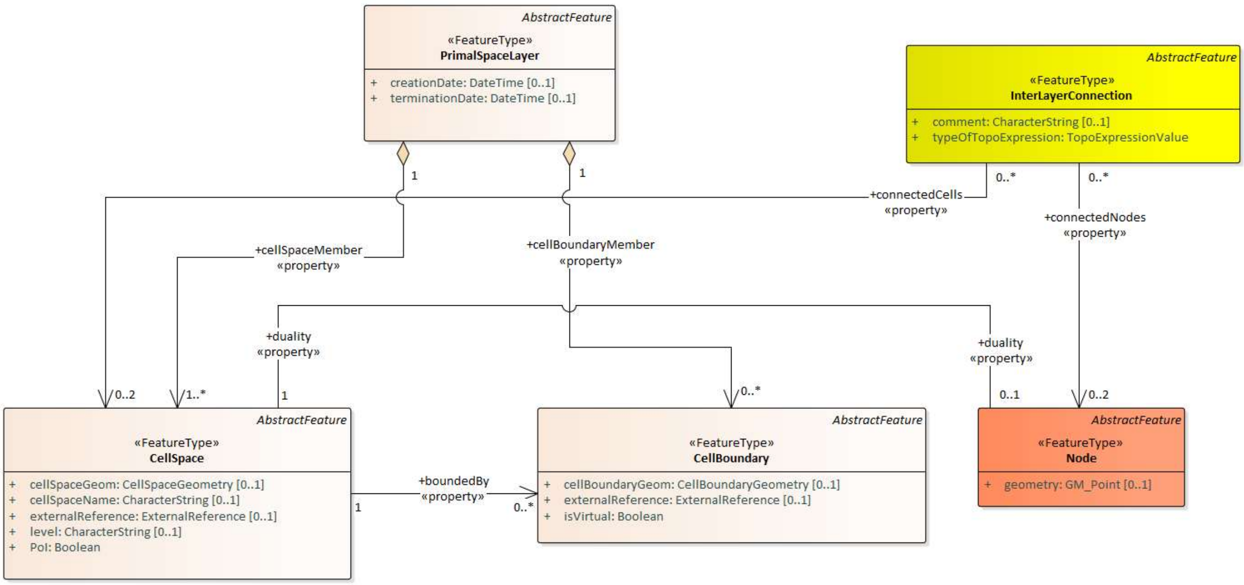

Figure 17: CellSpace and its related classes: PrimalSpaceLayer, CellBoundary, Node and InterLayerConnection

The *cellSpaceGeom* property carries an instance of type *CellSpaceGeometryType* for the description of geometric representations of space. A *CellSpaceGeometryType* is a geometry class type with two possible properties: *Geometry3D* or *Geometry2D*. They provide the 3D and 2D description of a CellSpace instance. The *Geometry3D* property 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* properties describes a representation of type surface, similar to the *GM_Surface* type. This property describes a CellSpace in 2D as one single surface (in the case of a 2D IndoorGML 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 property *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. For example, this is the case 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 property in IndoorGML 2 is *name*. The purpose is 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 property helps simplify space queries for applications. Another new property *PoI* is introduced to allow CellSpace elements to be flagged as Point of Interest for LBS applications. The property is a simple Boolean allowing the implementation of special considerations for flagged cells.

Note that apart from the *PoI* property, all applicable properties 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 property) 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. For example, a node can be identified as a *PoI* or associated with a *name* thanks to the property 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* association. For example, the 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 visualization). Finally, with the *duality* association, 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. For the case of multiple PrimalSpaceLayers, the class InterLayerConnection establishes the link between the depended CellSpace instances.

8.1.2 CellBoundary

CellBoundary is a core module class that describes the boundary of each cell in a cellular space. Unlike CellSpace, CellBoundary is not a compulsory class. It is only required when Edge instances exist in the model. CellBoundary contains the following properties:

- *cellBoundaryGeom (CellBoundaryGeometryType)*
- *externalReference (url)*
- *isVirtual (Boolean)*

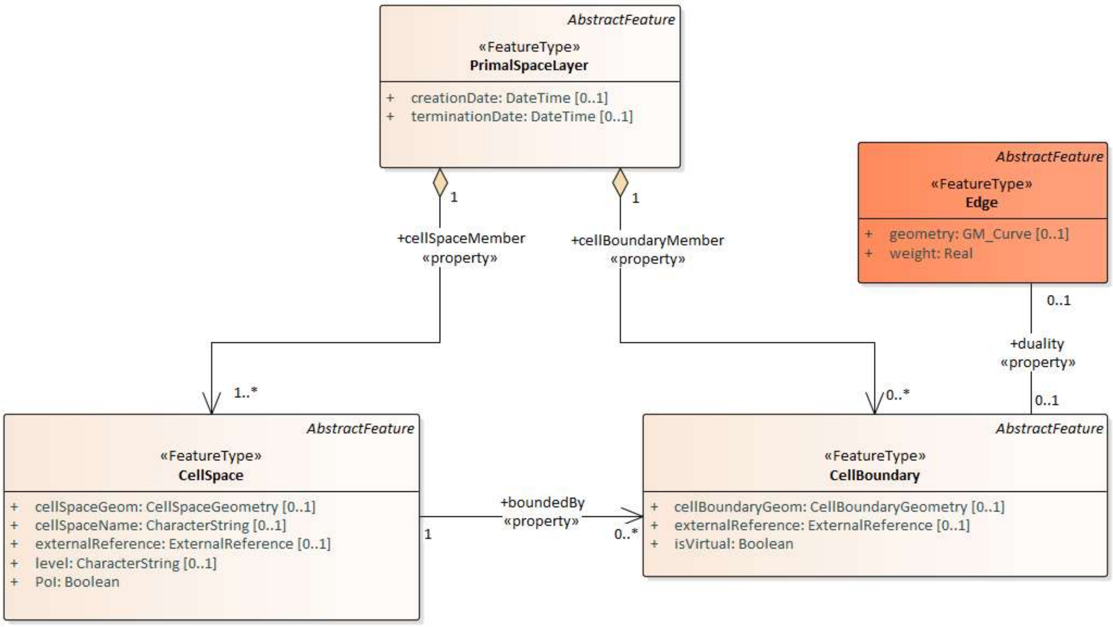
AbstractFeature
«FeatureType»
PrimalSpaceLayer
+ creationDate: DateTime [0..1]
+ terminationDate: DateTime [0..1]
1
1
+cellSpaceMember
«property»
+cellBoundaryMember
«property»
AbstractFeature
«FeatureType»
Edge
+ geometry: GM_Curve [0..1]
+ weight: Real
0..1
+duality
«property»
0..1
1..*
0..*
AbstractFeature
«FeatureType»
CellSpace
+ cellSpaceGeom: CellSpaceGeometry [0..1]
+ cellSpaceName: CharacterString [0..1]
+ externalReference: ExternalReference [0..1]
+ level: CharacterString [0..1]
+ PoI: Boolean
+boundedBy
«property»
1
0..*
AbstractFeature
«FeatureType»
CellBoundary
+ cellBoundaryGeom: CellBoundaryGeometry [0..1]
+ externalReference: ExternalReference [0..1]
+ isVirtual: Boolean

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

The *cellBoundaryGeom* geometry property 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 properties: *Geometry2D* and *Geometry1D*. The *Geometry2D* property is the same as that of *CellSpaceGeometryType*. Note, in this context, *cellBoundaryGeom* is embedded in 3D, i.e. has 3D coordinates and represents a part of the boundary of a CellSpace. The *Geometry1D* property describes a representation of type curve, GM_Curve type. Note, this property is intended for describing a CellBoundary in 2D as one single line/curve and has 2D coordinates. This is an adequate representation of *cellBoundaryGeom* based on 2D floor plans. CellBoundaryGeom is not required. In this case CellBoundaryGeom indicates only if a specific cell boundary is virtual.

The property *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* property 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* association, which corresponds to its dual representation. Unlike CellSpace, CellBoundary is not a mandatory instance in an IndoorGML data. In the case where there are CellSpace entities but no CellBoundary, the network may be derived from the cells using geometric operations.

In the case where there are CellBoundary entities provided without geometric properties in the model, only logical networks can be safely derived between two CellSpace entities sharing any of those CellBoundary. Therefore, providing geometric networks involves similar issues described previously. A final scenario may see an IndoorGML data 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.

8.1.3 PrimalSpaceLayer

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 properties) to represent spatial objects in primal space. The PrimalSpaceLayer class has the following properties:

- *creationDate (DateTime)*,
- *terminationDate (DateTime)*.

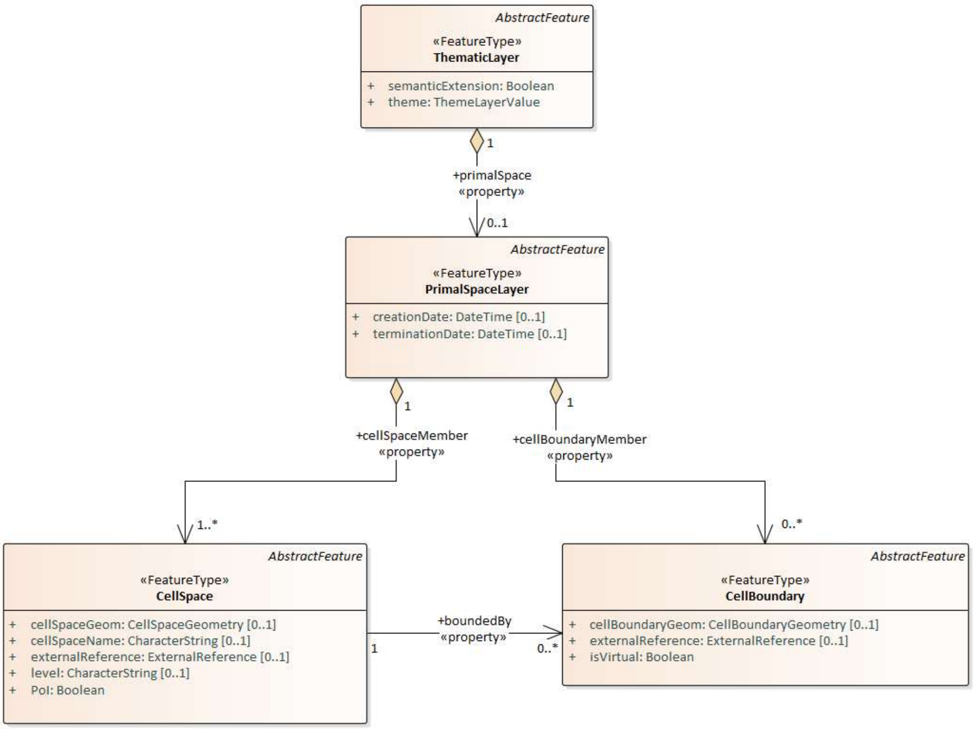
AbstractFeature
«FeatureType»
ThematicLayer
+ semanticExtension: Boolean
+ theme: ThemeLayerValue
1
+primalSpace
«property»
0..1
AbstractFeature
«FeatureType»
PrimalSpaceLayer
+ creationDate: DateTime [0..1]
+ terminationDate: DateTime [0..1]
1
+cellSpaceMember
«property»
1
+cellBoundaryMember
«property»
1..*
0..*
AbstractFeature
«FeatureType»
CellSpace
+ cellSpaceGeom: CellSpaceGeometry [0..1]
+ cellSpaceName: CharacterString [0..1]
+ externalReference: ExternalReference [0..1]
+ level: CharacterString [0..1]
+ PoI: Boolean
+boundedBy
«property»
1
0..*
AbstractFeature
«FeatureType»
CellBoundary
+ cellBoundaryGeom: CellBoundaryGeometry [0..1]
+ externalReference: ExternalReference [0..1]
+ isVirtual: Boolean

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

Both *creationDate* and *terminationDate* properties can be used to describe the chronology of the layer. A PrimalSpaceLayer instance also provides references to its CellSpace and CellBoundary entities through the *cellSpaceMember* and *cellBoundaryMember* elements.

8.1.4 Node

Node is a core module class for representing a node in dual space (Figure 20). It has one property:

- *geometry (GM_Point)*.

The value of *geometry* corresponds to a 2D or 3D Point in the IndoorGML nodel, 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, a Node always has an association with its corresponding CellSpace (e.g. room, door, sensor coverage, etc.) through the *duality* association. This way, a Node can always access 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* association.

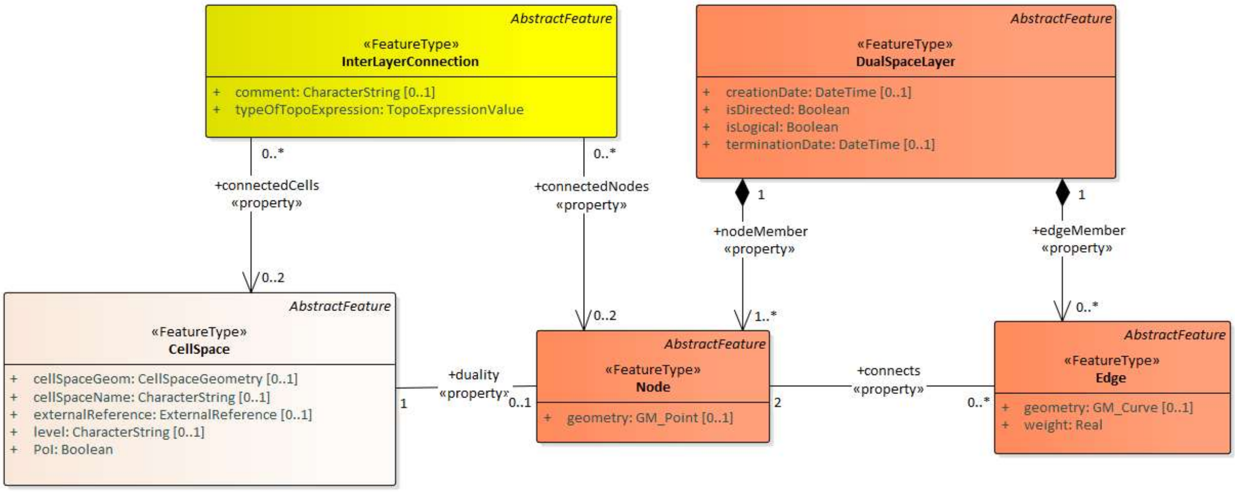

Figure 20: Node and its related classes: CellSpace, Edge, DualSpaceLayer and InterLayerConnection

8.1.5 Edge

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

- *geometry (GM_Curve),*
- *weight (real).*

The property *geometry* provides the description of a 2D or 3D curve, but similar to Node entities its cardinality can be 0 or 1 as well. The property *weight* can be used for graph-based applications such as dealing 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* association. This association can be skipped in situations where a CellBoundary is not necessary to represent the link between two CellSpace entities. For example, logical networks or visibility graphs where two CellSpaces connected by visibility may not share a CellBoundary. Finally, an Edge always connects two Nodes.

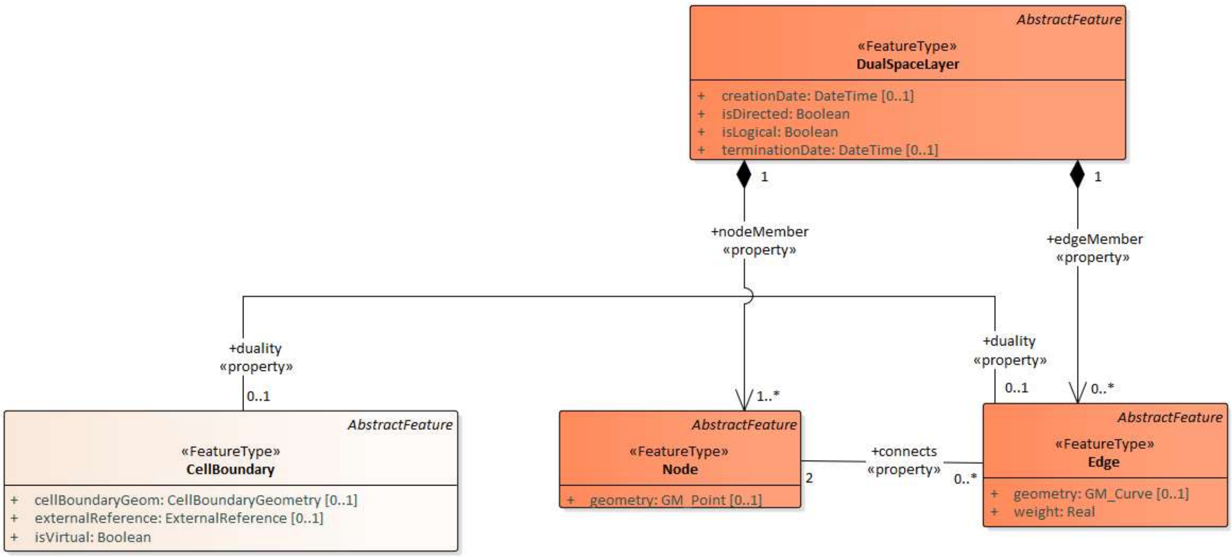

Figure 21: Edge and its related classes: CellBoundary, Node and DualSpaceLayer

8.1.6 DualSpaceLayer

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

- *isLogical* (Boolean)
- *isDirected* (Boolean)
- *creationDate (DateTime)*
- *terminationDate (DateTime)*

While *creationDate* and *terminationDate* are similar to those of PrimalSpaceLayer, the *isLogical* property allows differentiating 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* property 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* compositions.

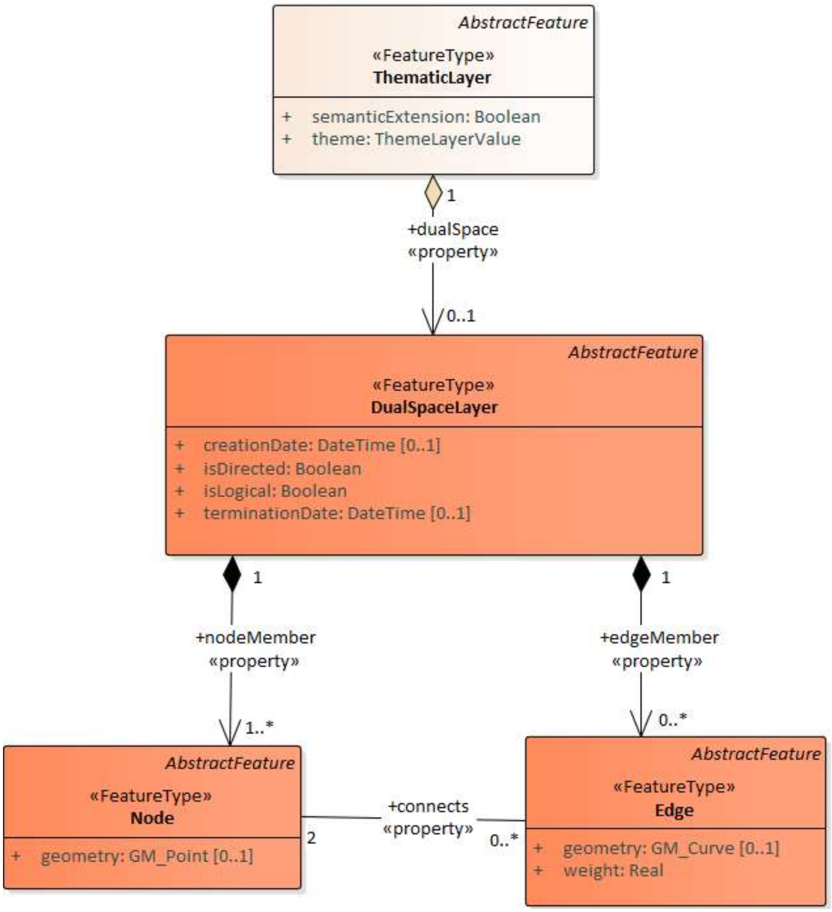

Figure 22: DualSpaceLayer and its related classes: Node, Edge and Thematic Layer.

8.1.7 InterLayerConnection

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

- *typeOfTopoExpression (TopoExpressionValue)*
- *comment* (string)

The *typeOfTopoExpression* property represents the topological relationship between two layers. *typeOfTopoExpression* is 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* property. In other words, for two layers successively described by the *connectedLayers* association, such as 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* assocations. The former is used when the connection is between two primal spaces and the latter is used otherwise. Finally, the *comment* property can contain an additional description for the InterLayerConnection.

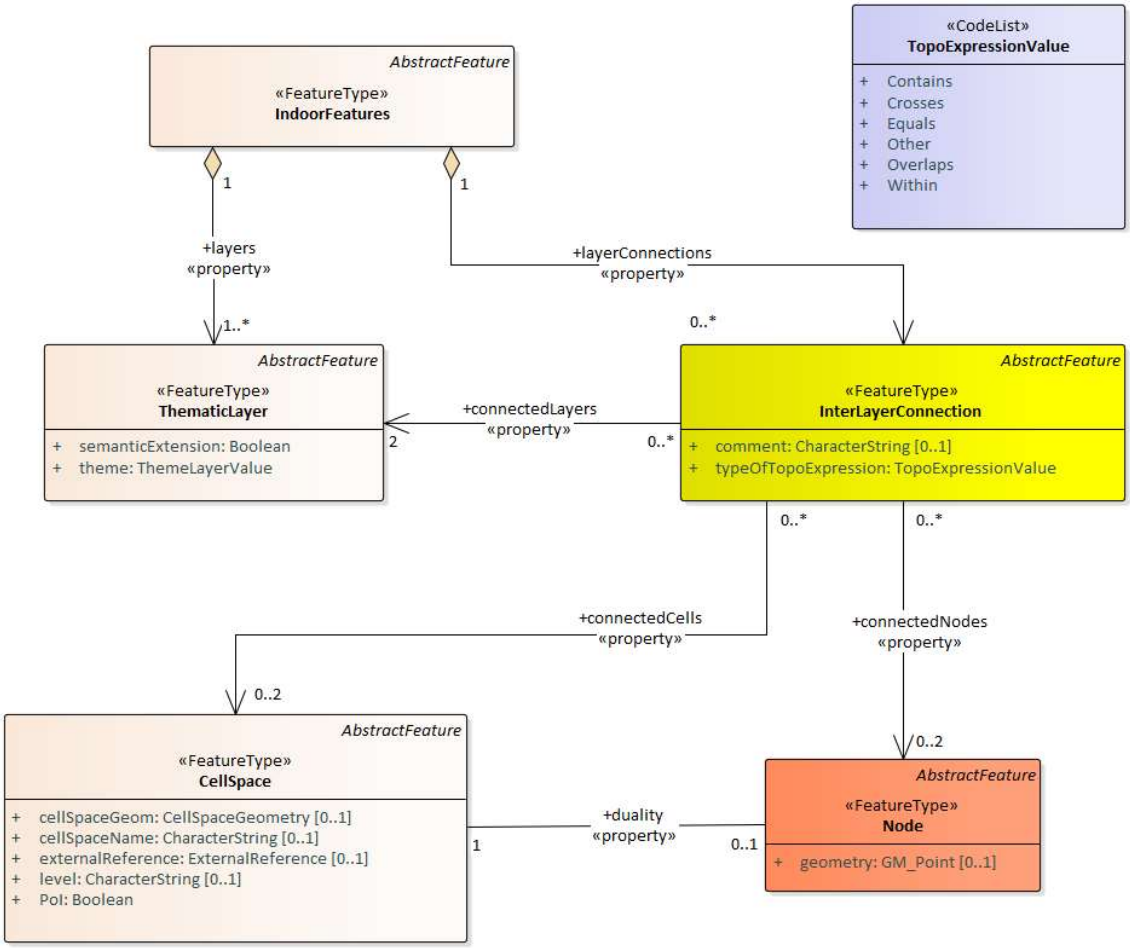

Figure 23: InterLayerConnection and its related classes: CellSpace, Node, ThematicLayer and IndoorFeatures

8.1.8 ThematicLayer

The ThematicLayer is a core module class introduced in IndoorGML2.0. ThematicLayer is 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 ThematicLayer class is composed of the following properties:

- *semanticExtension* (Boolean)
- *theme (ThemeLayerValue)primalSpace* (ref. to PrimalSpaceLayer)

The *semanticExtension* property is a Boolean that indicates whether there is Extension module with additional semantic information associated to the PrimalSpaceLayer. The IndoorGML 2.0 Standard only specifies the Navigation extension module (see Section 8.2). The use of a Boolean is considered enough to indicate the presence of the Navigation extension. However, this is expected to evolve in the future (into a codeList). The *theme* property determines what type of representation of the model can be expected in the corresponding layer (e.g topographic). The theme property 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) (Nagel et al. 2010). It is the most common type of layer for applications such as indoor navigation, where the physical elements highly constrain the use of the space. Similarly, a layer is qualified as *Virtual* when its description of the space relies exclusively on virtual, or a combination of physical and virtual extents. For example, the case for functional spaces that can represent spaces necessary for some indoor objects to operate or to be used properly (Diakité, 2018). This 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. The Tags type 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.

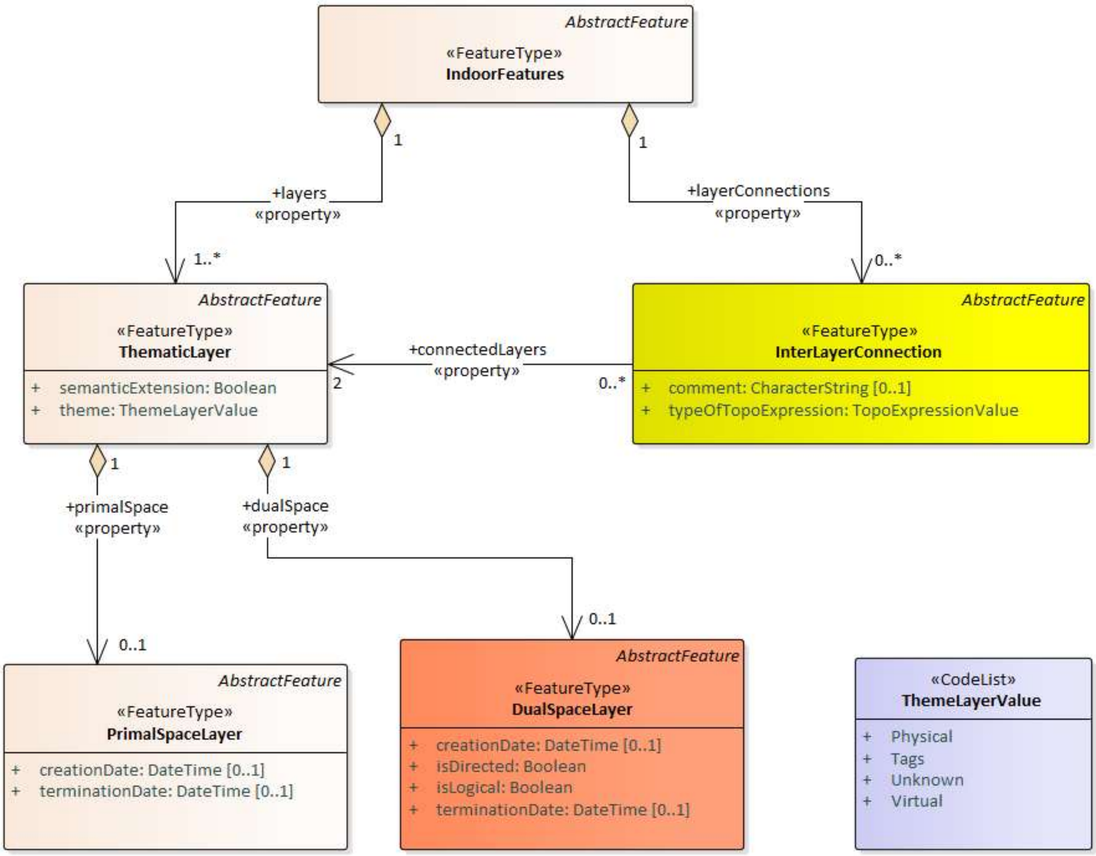

Figure 24: ThematicLayer and its related classes: PrimalSpaceLayer, DualSpaceLayer, InterLayerConnection and IndoorFeatures

8.2 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 properties 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.

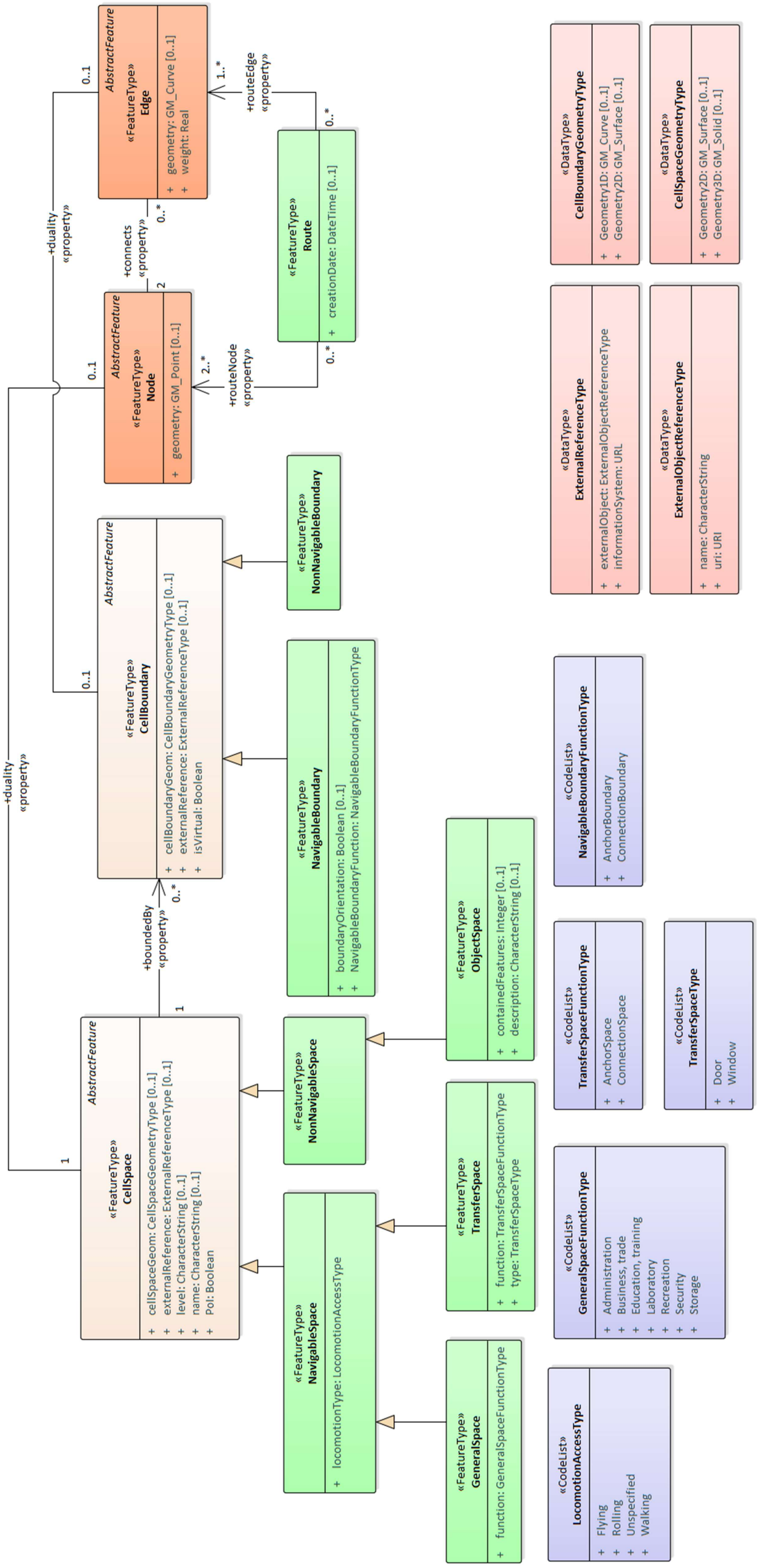

Figure 25: UML diagram of the Navigation Extension Module (classes in green)

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 (such as openings. NonNavigableSpace represents all indoor spaces that are not navigable, either because they are physically occupied by indoor features such as furniture or 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 are for describing 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.

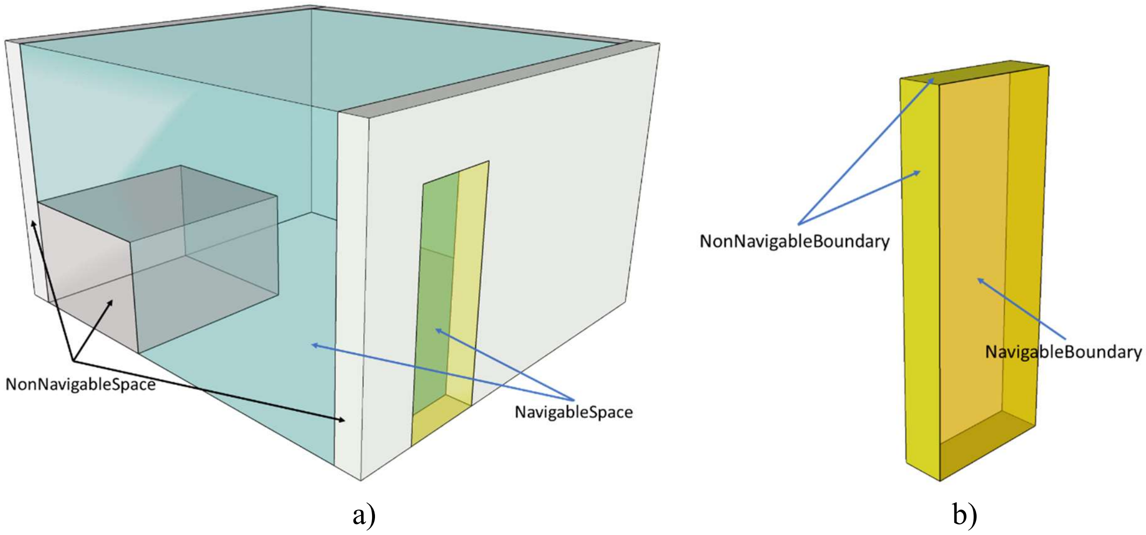

Figure 26: Navigable and Non-navigable spaces (a) and boundaries (b) on a 3D model with walls and furniture (grey), indoor space (blue) and a door space (yellow).

8.2.1 NavigableSpace

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 the 3D case. In 2D, doors are commonly represented as boundaries of rooms and must be considered NavigableBoundaries (see Section 8.2.4)

NavigableSpace entities can carry information about the type of locomotion, which is the allowed transportation mode in indoor space. The *locomotionType* property has 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.

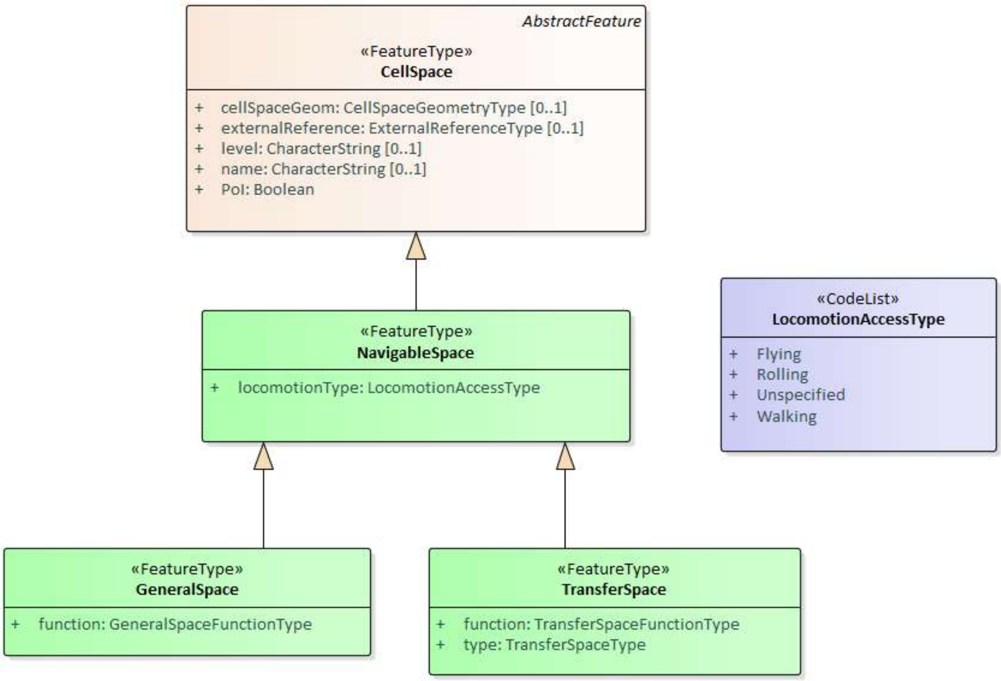
AbstractFeature
«FeatureType»
CellSpace
+ cellSpaceGeom: CellSpaceGeometryType [0..1]
+ externalReference: ExternalReferenceType [0..1]
+ level: CharacterString [0..1]
+ name: CharacterString [0..1]
+ PoI: Boolean
«CodeList»
LocomotionAccessType
+ Flying
+ Rolling
+ Unspecified
+ Walking
«FeatureType»
NavigableSpace
+ locomotionType: LocomotionAccessType
«FeatureType»
GeneralSpace
+ function: GeneralSpaceFunctionType
«FeatureType»
TransferSpace
+ function: TransferSpaceFunctionType
+ type: TransferSpaceType

Figure 27: NavigableSpace and its related class: CellSpace

8.2.2 GeneralSpace

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 property *function* which specifies details about the function of the cell. In IndoorGML, those functions are described in a code list derived from OmniClass Table 13 (OmniClass, 2021)

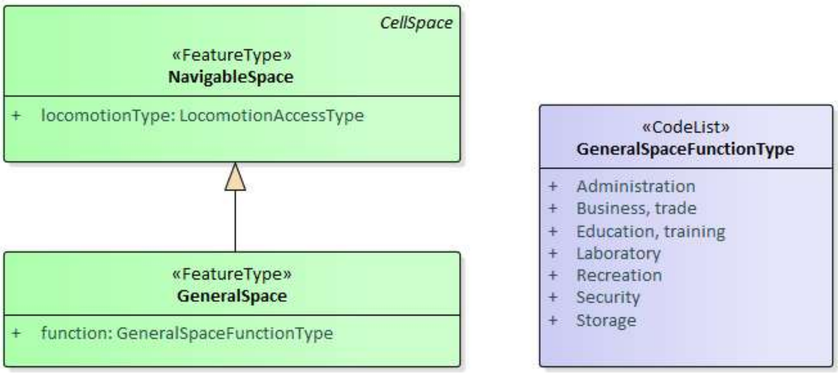

Figure 28: General Space and its related class: NavigableSpace

8.2.3 TransferSpace

The class TransferSpace is specialization of NavigableSpace. It is used to model a space that provides 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, TransferSpace carries a *function* property 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 property is *type* which specified through a codeList the TransferSpaceType (Door or Window).

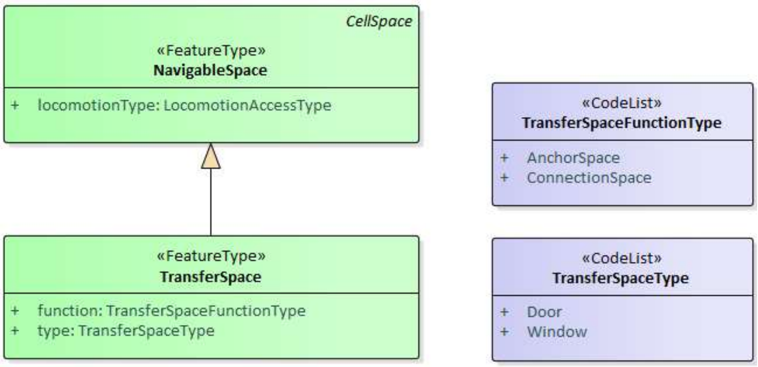

Figure 29: Transfer Space and its related class: NavigableSpace

8.2.4 NavigableBoundary

The NavigableBoundary class is a specialization 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. The NavigableBoundary class is therefore mainly found between GeneralSpace and TransferSpace cells but can happen between two GeneralSpace cells as well such as 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 serves for transition between connected spaces.

The class carries a *boundaryOrientation* property and a *navigableBoundaryFunction* property specifying if the boundary is an *AnchorBoundary* or a *ConnectionBoundary* (see subsection 8.2.3 for more details).

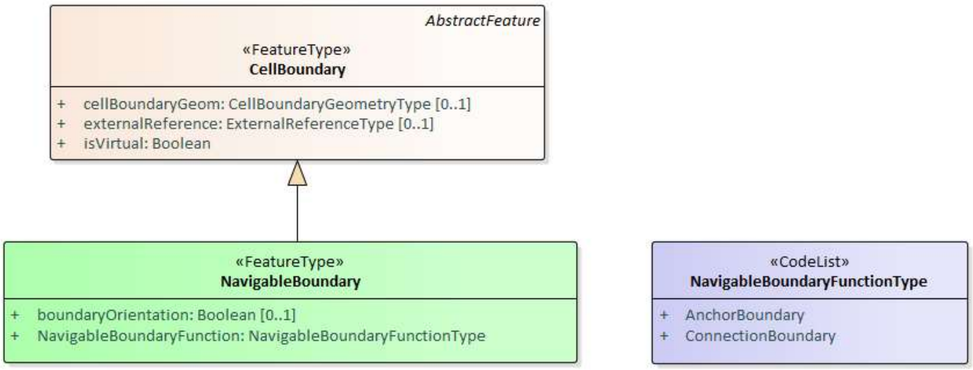
AbstractFeature
«FeatureType»
CellBoundary
+ cellBoundaryGeom: CellBoundaryGeometryType [0..1]
+ externalReference: ExternalReferenceType [0..1]
+ isVirtual: Boolean
«FeatureType»
NavigableBoundary
+ boundaryOrientation: Boolean [0..1]
+ NavigableBoundaryFunction: NavigableBoundaryFunctionType
«CodeList»
NavigableBoundaryFunctionType
+ AnchorBoundary
+ ConnectionBoundary

Figure 30: Navigable Boundary and its related class: CellBoundary

8.2.5 NonNavigableSpace

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.). NonNavigableSpace is a class without properties, but allows options to further classify non-navigable cells.

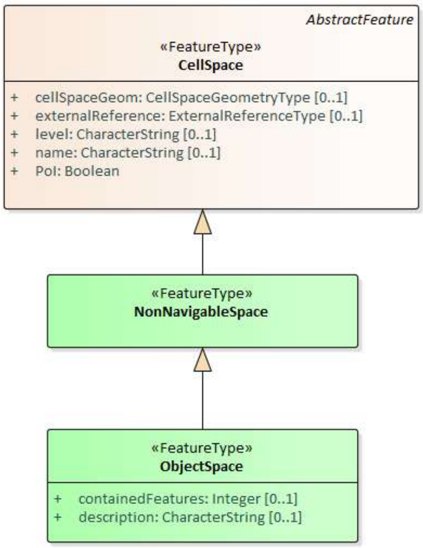
AbstractFeature
«FeatureType»
CellSpace
+ cellSpaceGeom: CellSpaceGeometryType [0..1]
+ externalReference: ExternalReferenceType [0..1]
+ level: CharacterString [0..1]
+ name: CharacterString [0..1]
+ PoI: Boolean
«FeatureType»
NonNavigableSpace
«FeatureType»
ObjectSpace
+ containedFeatures: Integer [0..1]
+ description: CharacterString [0..1]

Figure 31: NonNavigableSpace its related class: CellSpace

8.2.6 ObjectSpace

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 properties: *containedFeatures (Integer),* and *description (String).*

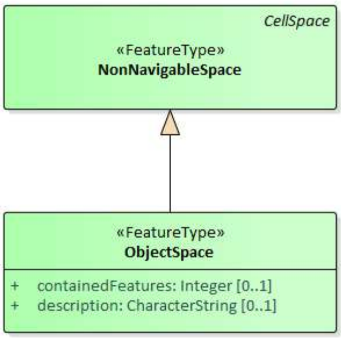
CellSpace
«FeatureType»
NonNavigableSpace
«FeatureType»
ObjectSpace
+ containedFeatures: Integer [0..1]
+ description: CharacterString [0..1]

Figure 32: ObjectSpace and its related class: NonNavigableSpace.

The *containedFeatures* property 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* property is meant to provide any relevant information regarding the objects contained within the space in plain text.

8.2.7 NonNavigableBoundary

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.

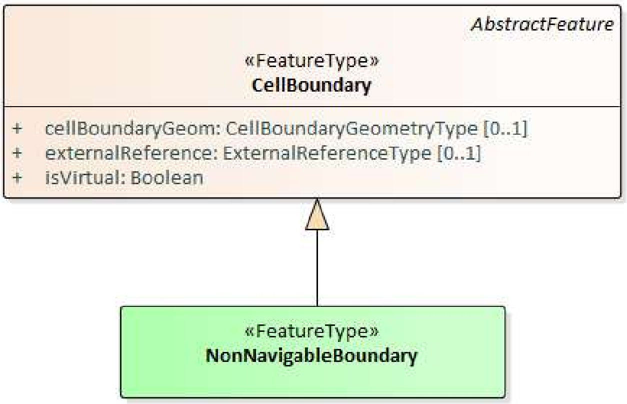
AbstractFeature
«FeatureType»
CellBoundary
+ cellBoundaryGeom: CellBoundaryGeometryType [0..1]
+ externalReference: ExternalReferenceType [0..1]
+ isVirtual: Boolean
«FeatureType»
NonNavigableBoundary

Figure 33: NonNavigableBoundary and its related classes: CellBoundary

8.2.8 Route

The Route class is a specialization 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.

The Route class has one property: *creationDate.* Because dynamic indoor environments may imply change in space availability and accessibility, a suiteable path at a given time may not be suitable at another time. For this reason, the *creationDate* property helps indicating at which time a given route was created. The *routeNode* and *routeEdge* properties 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.

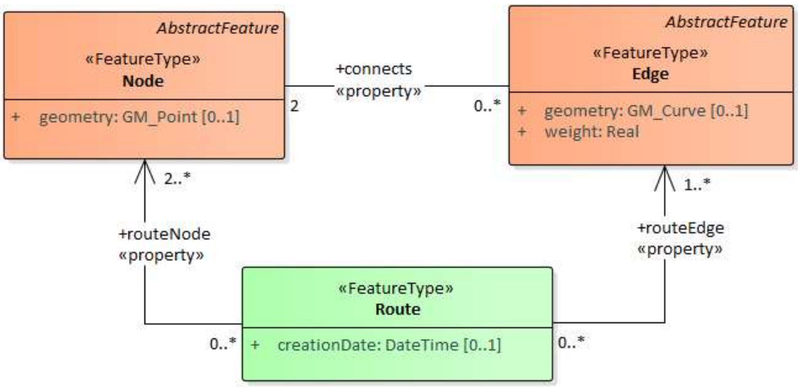

Figure 34: Route and its related classes: Node and Edge

8.3 Requirements

In this subsection, the requirements for implementing IndoorGML Part 1 – Conceptual Model are defined. The implementation of IndorrGML Part I SHALL be in conformance with the requirements given in the list below.

Requirement 1 – UML class diagram
/req/umlclassdiagram
For the UML class diagram defined in Core Module and Navigation Module
The Implementation Specification SHALL contain or represents the same concepts as defined in the UML class diagram – - Contain elements which represent the same concept as that defined for the UML class, - Represent associations of the UML classes and their superclasses with the same source, target, direction, roles, and multiplicities, and - Contain the attributes of the classes and their superclasses with the same name, definition, type, code list, and multiplicity,

Requirement 2 – Thematic layer
/req/thematiclayer
Any feature of a thematic layer SHALL belong to the same theme.

Requirement 3 – Cell Space	
/req/cellspace	
A	Any feature of a thematic layer SHALL belong to the same theme.
B	Cell belonging to the same primal space layer SHALL not intersect.

Requirement 4 – Cell Boundary	
/req/cellboundary	
A	Cell boundaries belonging to the same primal space layer SHALL not intersect.
B	The geometry of cell boundary SHALL not exceed the extent of the corresponding cell space

Requirement 4 – Node
/req/node
When the isLogical property of a DualSpaceLayer is set to FALSE, the geometries of its Node instances SHALL be spatially located inside of their corresponding CellSpaces.

Requirement 5 – Edge	
/req/edge	
A	No self-intersection is allowed when its geometry is given.
B	If dualspaceLayer.directed=true, then the order of nodes represents the direction.

Requirement 6 – Interlayer Connection	
/req/interlayerconnection	
A	Two target cell spaces (or nodes) SHALL not belong to a same primal space layer (or dual space layer)
B	Connected nodes or connected cells SHALL be consistent with connected layers. This means that the target cell spaces (or nodes) SHALL belong to primal space layer (or dual space layer) of the connected layer
C	The cardinalities of Node (connectedNodes property) and CellSpace (connectedCells property) SHALL either be 0 or 2 but can never be 1.
D	Two connectedNodes are not commutative. For example, "node *A* contains node B” does not mean “node *B* contains node A”.

Requirement 7 – ObjectSpace
/req/objectspace
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).

9 Data dictionary and requirements

In this section, the data dictionary of the feature types defined in IndoorGML 2.0 UML class diagram are defined. The aim is to clarify the concepts of each feature type and help the implementation of this standard. The data dictionary is defined based on ISO TC 211 Standards, particularly ISO 19109 for the rules for application schema, ISO 19107 for spatial schema, and ISO 19136 for GML. As IndoorGML 2.0 is a model based on these base standards, the data dictionary for the feature types defined by these standards is not included in this 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 provided in the following subsections.

9.1 Feature Types in Core Module

9.1.1 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)	

9.1.2 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	Any feature of a thematic layer SHALL belong to the same theme (Requirement ID: /req/thematiclayer)

9.1.3 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**
	none	

9.1.4 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	Cell spaces belonging to the same primal space layer SHALL not overlap. (Requirement ID: /req/cellspace)

9.1.5 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. . (Requirement ID: /req/ cellboundary-A)

	Indoorgml2/constraints/ cellboundary-2	The geometry of cell boundary SHALL not exceed the extent of the corresponding cell space. (Requirement ID**:** /req/ cellboundary-B)

9.1.6 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 [0..1]
Constraints	**Constraint ID**	**Constraint**
	none	

9.1.7 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	When the isLogical property of a DualSpaceLayer is set to FALSE, the geometries of its Node instances SHALL be spatially located inside of their corresponding CellSpaces. (Requirement ID: /req/node)

9.1.8 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. (Requirement ID: /req/edge-A)
	Indoorgml2/constraints/edge-2	If dualspaceLayer.directed=true, then the order of nodes represents the direction. (Requirement ID: /req/edge-B)

9.1.9 InterLayerConnection

Name	**InterLayerConnection**	
Definition	Relationship between cell spaces and nodes in two different thematic layers	
Super classes	GML AbstractFeature	
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) (Requirement ID: /req/interlayerconnection-A)
	Indoorgml2/constraints/ InterLayerConnection-2	Connected nodes or connected cells SHALL be consistent with connected layers. This means that the target cell spaces (or nodes) SHALL belong to primal space layer (or dual space layer) of the connected layer. (Requirement ID: /req/interlayerconnection-B)
	Indoorgml2/constraints/ InterLayerConnection-3	The cardinalities of Node and CellSpace SHALL either be 0 or 2 but can never be 1. (Requirement ID: /req/interlayerconnection-C)
	Indoorgml2/constraints/ InterLayerConnection-4	Two connectedNodes are not commutative. For example, "node *A* contains node B" does not mean "node *B* contains node A". (Requirement ID: /req/interlayerconnection-D)

9.2 Feature Types in Navigation Module

9.2.1 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	

9.2.2 NonNavigableSpace

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

9.2.3 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	

9.2.4 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	

9.2.5 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]
	description	string [0..1]
	Requirement ID	**Constraint**
Constraints	Indoorgml2/constraints/ objectspace	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). (Requirement ID: /req/objectspace)

9.2.6 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	

9.2.7 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	

9.2.8 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**
	none	

A Annex A (Normative) Abstract Test Suite

A.1 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.

A.2 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.

A.3 UML Data Model

ID	Indoorgml2/ate/umldatamodel
Requirement	UML class diagram defined in section 8 SHALL be correctly applied to any implementation of IndoorGML Part 2 – Conceptual Model Requirement ID: req/umlclassdiagram
Test Method	Manual or automated inspection

A.4 Class ThematicLayer

ID	Indoorgml2/ate/thematiclayer
Requirement	Any feature of a thematic layer SHALL belong to the same theme. Requirement ID: req/thematiclayer
Test Method	Manual or automated inspection

A.5 Class CellSpace

ID	Indoorgml2/ate/cellspace
Requirement	Any feature of a thematic layer SHALL belong to the same theme. Requirement ID: req/cellspace
Test Method	Automated inspection by geometric computation

A.6 Class CellBoundary

ID	Indoorgml2/ate/cellboundary
Requirements	Cell boundaries belonging to the same primal space layer SHALL not intersect. Requirement ID: req/ cellboundary-A
	The geometry of cell boundary SHALL not exceed the extent of the corresponding cell space Requirement ID: req/ cellboundary-A
Test Method	Automated inspection by geometric computation

A.7 Class Node

ID	Indoorgml2/ate/node
Requirement	When the isLogical property of a DualSpaceLayer is set to TRUE, the geometries of its Node instances SHALL be spatially located inside of their corresponding CellSpaces. Requirement ID: req/node
Test Method	Automated inspection by geometric computation

A.8 Class Edge

ID	Indoorgml2/ate/edge
Requirement	No self-intersection is allowed when its geometry is given. Requirement ID: req/edge-A
	If dualspaceLayer.directed=true, then the order of nodes represents the direction. Requirement ID: req/ edge-B
Test Method	Automated inspection by geometric computation

A.9 Class InterLayerConnection

ID	Indoorgml2/ate/interlayerconnection
Requirement	Two target cell spaces (or nodes) SHALL not belong to a same primal space layer (or dual space layer) Requirement ID: req/interlayerconnection-A
	Connected nodes or connected cells SHALL be consistent with connected layers. This means that the target cell spaces (or nodes) SHALL belong to primal space layer (or dual space layer) of the connected layer Requirement ID: req/interlayerconnection-B
	The cardinalities of Node and CellSpace SHALL either be 0 or 2 but can never be 1. Requirement ID: req/interlayerconnection-C
	Two connectedNodes are not commutative. For example, "node *A* contains node B" does not mean "node *B* contains node A". Requirement ID: req/interlayerconnection-D
Test Method	Automated inspection by geometric computation

A.10 Class ObjectSpace

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

B 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
- Becker, T., Nagel, C., and Kolbe, T. H., (2009) A Multilayered Space-Event Model for Navigation in Indoor Spaces, Proc. 3rd International Workshop on 3D Geoinformation
- 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
- Lee, J. (2004). A spatial access-oriented implementation of a 3D GIS topological data model for urban entities. GeoInformatica, 237-264.
- 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.
- 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.