OGC 14-005r4

Open Geospatial Consortium

Submission Date: 2015-08-24

Approval Date: 2015-09-07

Publication Date: 2016-08-23

External identifier of this OGC® document: http://www.opengis.net/doc/IS/indoorgml/1.0

Normative URL for this document: http://docs.opengeospatial.org/is/14-005r4/14-005r4.html

Internal Reference number of this OGC® project document: OGC 14-005r4

Version: 1.0.2

Category: OGC® Implementation

Editors: Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Thomas H. Kolbe, Claus Nagel, Thomas Becker

OGC® IndoorGML – with Corrigendum

Copyright notice

Warning

This document is an OGC Member approved international standard, however, this version is informative. The normative version is available at: **http://docs.opengeospatial.org/is/14-005r3/14-005r3.html**. This document is available on a royalty free, non-discriminatory basis. 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. This is a corridendum for 14-005r3

Document type:	OGC® Standard
Document subtype:	Encoding
Document stage:	Approved as official OGC standard
Document language:	English

Contents

Abstract

This OGC® IndoorGML standard specifies an open data model and XML schema of indoor spatial information. IndoorGML is an application schema of OGC® GML 3.2.1. While there are several 3D building modelling standards such as CityGML, KML, and IFC, which deal with interior space of buildings from geometric, cartographic, and semantic viewpoints, IndoorGML intentionally focuses on modelling indoor spaces for navigation purposes.

Keywords

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

Submitting organizations

The following organizations submitted this Implementation Standard to the Open Geospatial Consortium:

a) Pusan National University,

b) University of Seoul,

c) Technical University of Munich,

d) Technical University of Berlin,

e) Technical University of Delft,

f) University of Nottingham,

g) Hitachi Ltd.,

h) Electronic and Telecommunication Research Institute

Submission contact points

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

Name	Affiliation	Contact
Ki-Joune Li	Pusan National University	lik at pnu.edu
Jiyeong Lee	University of Seoul	jlee at uos.ac.kr
Thomas H. Kolbe	Technical University of Munich	thomas.kolbe at tum.de
Sisi Zlatanova	Technical University of Delft	S.Zlatanova at tudelft.nl
Jeremy Morley	University of Nottingham	Jeremy.Morley at notingham.ac.uk
Claus Nagel	Virtual City Systems	cnagel at virtualcitysystems.de
Thomas Becker	Technical University of Berlin	thomas.becker at tu-berlin.de

Participants in development

Name	Institute
Ki-Joune Li, Joonseok Kim	Pusan National University
Jiyeong Lee, Hye-Young Kang	University of Seoul
Thomas H. Kolbe,	Technical University of Munich
Thomas Becker, Robert Kaden	Technical University of Berlin
Claus Nagel	Virtual City Systems
Sisi Zlatanova	Technical University of Delft
Jeremy Morley	University of Nottingham
Nobuhiro Ishimaru Yaemi Teramoto	Hitachi Ltd.
Jae-Joon Yu	ETRI
Seok-Ho Lee, Dongkwon Seo	Hyundai MNSoft Co.
Cliff Behrens	Applied Communication Sciences

Changes to the OGC® Abstract Specification

The OGC® Abstract Specification does not require changes to accommodate this OGC® standard.

Foreword

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.

Introduction

The goal of IndoorGML is to represent and allow for exchange of geoinformation that is required to build and operate indoor navigation systems. Several standards such as CityGML[10,28], KML[30], and IFC[31] have been published to describe 3D geometry and semantics of buildings not only for outdoor space but also indoor space, but they lack important features that are required by indoor navigation applications. This standard aims to provide complementary and additional encoding features for indoor spatial information required for indoor navigation.

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

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

1 Scope

IndoorGML is an OGC® standard for the representation and exchange of indoor navigation network models. IndoorGML is implemented as an application schema of the Geography Markup Language version 3.2.1.

IndoorGML aims to establish a common schema for indoor navigation applications. It models topology and semantics of indoor spaces, which are needed for the components of navigation networks. Nevertheless, IndoorGML contains only a minimum set of geometric and semantic modelling of construction components to avoid duplicated efforts with other standards, such as CityGML and IFC.

IndoorGML defines the following information about indoor space;

- Navigation context and constraints
- Space subdivisions and types of connectivity between spaces
- Geometric and semantic properties of spaces and connectivity
- Navigation networks (logical and metric) and their relationships

2 Conformance

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

In order to conform to IndoorGML, and schema document should

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

b) pass all relevant test cases of the abstract test suite given in annex B

3 Normative references

The following normative documents contain provisions which, 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. However, parties to agreements based on this document are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

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

ISO/TS 19103:2005, *Geographic Information – Conceptual Schema Language*

ISO 19105:2000, *Geographic information – Conformance and testing*

ISO 19107:2003, *Geographic Information – Spatial Schema*

ISO 19109:2005, Geographic Information – *Rules for Application Schemas*

ISO 19111:2003, Geographic information – *Spatial referencing by coordinates*

ISO 19115:2003, Geographic Information – *Metadata*

ISO/TS 19139:2007, Geographic Information – *Metadata – XML schema implementation*

OpenGIS® Abstract Specification Topic 0, *Overview, OGC document 04-084*

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

OpenGIS® Abstract Specification Topic 8, *Relations between Features, OGC document 99-108r2*

OpenGIS® Abstract Specification Topic 10, *Feature Collections, OGC document 99-110*

OpenGIS® Geography Markup Language Implementation Specification, *Version 3.2.1, OGC document 07-036*

4 Terms and definitions

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

4.1 Indoor Space

A space within one or multiple buildings consisting of architectural components.

4.2 Coordinates Space

A space where location is identified by (*x,* y) or (*x, y, z*) coordinates where *x, y,* and *z* are real values.

4.3 Cellular Space and Symbolic Space

A space where location is identified by a cell identifier (or symbolic code).

4.4 NR (Node-Relation) Graph

A graph (*V*, *E*) where *V* is a set of nodes representing cells in indoor space and *E* is the set of edges indicating the topological relationship between two cells, which may be connectivity or adjacency.

4.5 Connectivity NRG

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

4.6 Adjacency NRG

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

4.7 Accessibility NRG

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

4.8 Logical NRG

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

4.9 Geometric NRG

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

4.10 Multi-Layered Space Model

A space represented by multiple layers of connectivity graphs and inter-layer connections between two nodes from different layers.

5 Conventions

5.1 Symbols (and abbreviated terms)

The following symbols and abbreviated are used in this standard;

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
MVD	Model View Definition

NRG	Node-Relation Graph
OGC	Open Geospatial Consortium
RFID	Radio Frequncy IDentifier
SensorML	Sensor Model Language
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.

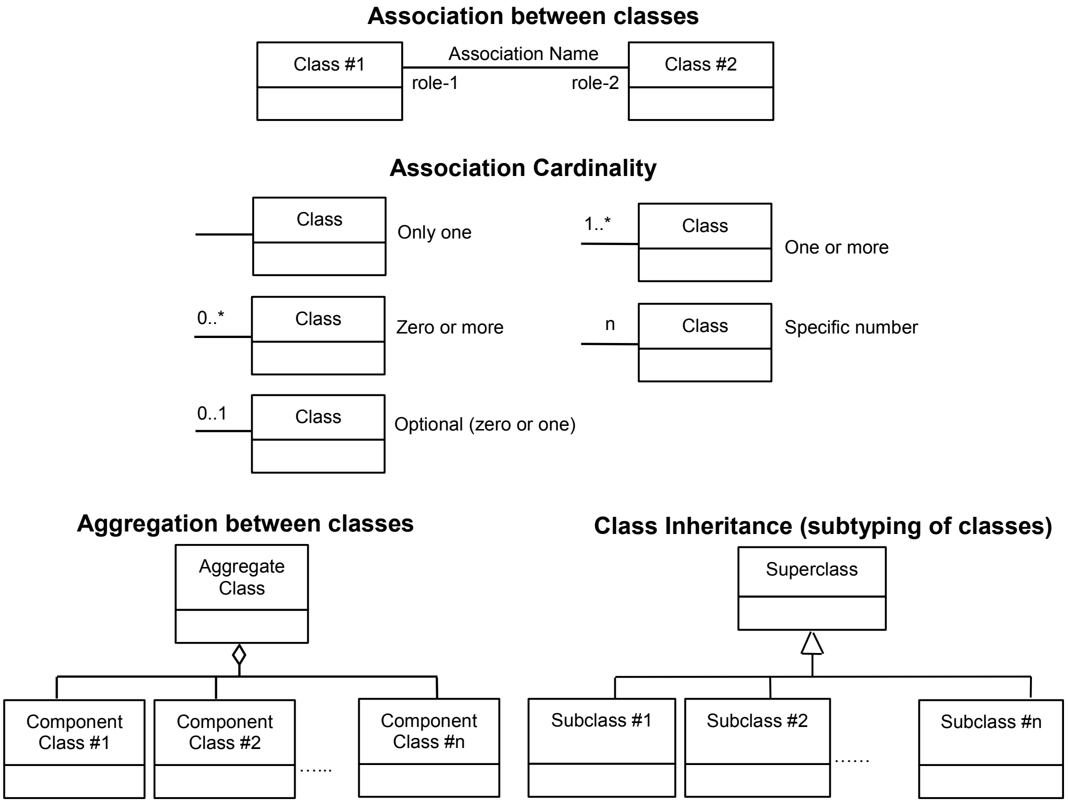


Figure 1: UML notation

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

a) <<Interface>> A definition of a set of operations that is supported by objects having this interface. An Interface class cannot contain any attributes.

b) <<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.

c) <<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:

a) CharacterString – A sequence of characters

b) Integer – An integer number

c) Double – A double precision floating point number

d) Float – A single precision floating point number

6 Overview of IndoorGML

6.1 Motivation for defining IndoorGML

Indoor space differs from outdoor space in many aspects. Therefore, basic concepts, data models, and standards need to be redefined to meet the requirements of indoor spatial applications. The requirements of indoor spatial information are specified according to the types of applications. In general, the applications of indoor spatial information are classified into two categories as follows;

- Management of building components and indoor facilities, and
- Usage of indoor space.

Building construction and management and facility management belong to the first category. While the main focus of the first category are on building components such as roofs and walls, the second category is focused on usage and localization of features (stationary or mobile) in indoor space. The indoor spatial information of the second category includes requirements for representing spatial components such as rooms and corridors, and constraints such as doors. Indoor location-based services, indoor route analysis or indoor geo-tagging services belong to the second category.

Instead of representing building architectural components, the goal of the IndoorGML standard is to define a framework of indoor spatial information to locate stationary or mobile features in indoor space and to provide spatial information services utilizing their positions in indoor space. IndoorGML is intended to provide the following functions;

- Representing the properties of indoor space, and
- Providing spatial reference of features in indoor space.

In summary, IndoorGML version 1 is based on the requirements from indoor navigation due to strong and urgent standardization demands, such as indoor LBS, routing services, and emergency control in indoor space. We expect that other requirements such as facilities management will be handled by future versions of IndoorGML.

7 General characteristics of IndoorGML

7.1 Representation of Indoor Objects in IndoorGML

An important difference of indoor space from outdoor space is that an indoor space is composed of complicated constraints such as corridors, doors, stairs, elevators, etc., like a road network space is composed of road constraints. This means that proper representations of indoor constraints are key issues for indoor spatial information modelling and standards. In IndoorGML, indoor constraints are considered from the following aspects;

- Cellular space (defined below)
- Semantic representation
- Geometric representation
- Topological representation
- Multi-Layered Representation

7.1.1 Definition of Indoor Space

Indoor space is defined as space within one or multiple buildings consisting of architectural components such as entrances, corridors, rooms, doors, and stairs. In IndoorGML, we are not concerned about architectural components themselves (e.g. roofs, ceilings, walls) but instead the spaces (e.g. rooms, corridors, stairs) defined by architectural components, where objects can be located and navigate. IndoorGML is also concerned with the relationships between spaces. Components irrelevant to describe the spaces, such as furniture, are not within the scope of IndoorGML.

While indoor space models may be restricted to a single building, they may be comprised of multiple buildings or a complex of connected buildings. All the buildings are not necessarily covered by a roof. For example, an inner court or veranda can belong to an indoor space.

7.1.2 Cellular notion of space and cells

One of the differences of IndoorGML from standards dealing with building interior space such as IFC is that they provide standards for representing architectural components but standards for indoor space. For this reason, we consider an indoor space as a set of *cells*, which are defined as the smallest organizational or structural unit of indoor space [28]. A cellular space S is defined as follows;

$S = \{c_1, c_2, \dots , c_n \}$, where c_i is i^{th} cell.

Cellular space has important properties. First, every cell has an identifier (namely *c*.ID), such as a room number. Second, each cell may have a common boundary with other cells but does not overlap with any other cell. Third, postion in cellular space can be specified by cell identifier, although we may employ (*x*, *y*, *z*) coordinates to specify a position for more precise location.

While a set of cells is the minimum information to determine a cellular space, additional information can be also included in cellular space as follows;

- semantics: e.g. classification and interpretation of cells
- geometry: e.g. solids in 3D or surfaces in 2D
- topology: e.g. adjacency or connectivity

7.1.3 Semantic Representation of Indoor Space

Semantics are an important characteristic of cells. The indoor space can be decomposed into different cells if differet criteria are considered. The cell subdivision can represent the topography (construction) of a building, available WiFi coverages, indicate security areas, or public/office areas, etc. Every cell is then given semantics with respect to the semantics used to the space subdivision. For example, in a topographic space it is possible to have *'room', 'door', 'window',* in WiFi space - '*WiFi point A', '*WiFi *point B', etc.* and in a security space - '*check-in area'*, '*boarding area', 'crew areas'.*

In IndoorGML, semantics is used for two purposes: to provide classification and to identify a cell and determine the connectivity between cells. Semantics thus allows us to define cells that are important for navigation. For example, the most commonly used classification of cells in topolographic space is into navigable (rooms, corridors, doors) and non-navigable (walls, obstacles) cells.

Cells can be organised in a hierarchical structure according to their semantics, corresponding properties and semantic interrelations (specialisation and generalisation). For example 'room' is a specialisation of 'navigable cell' and 'non-navigable cell' is a generalisation of 'walls' and 'obstacles'. Cells created for one space represenation may be aggregated or subdivided for the purpose of another one. For example, in security space 'check-in area' cell can be an aggregation of several 'room' cells, which have been created for the topography space.

Connectivity, in terms of possibility to navigate through cells, is largely derived from the semantics of cells. For example to be able to go from one room to another, we need to know that at least one common opening (door, window) cell exists.

The properties of a semantically identified cell have impact on connectivity and can act as a navigation constraint. For example, certain doors might provide access in one direction only (emergency exits), or forbid access to a specific group of users (security areas) or allow access according to specific time intervals (e.g. shops).

IndoorGML allows different space representation to be integrated via the concept of Multi-Layered Space Representation (see 7.1.6).

7.1.4 Geometric Representation of Indoor Space

Every cell defining indoor space, such as a room or a corridor, owns a form, extension and position that can be collected and modelled. Geometric information can be included in IndoorGML in several ways. In order to represent geometry of cell, we assume 3D or 2D Euclidean spaces. Using the concepts of Euclidean space, the geometry provides the means for the quantitative description of the spatial characteristics of cell, where a metric space is defined as [18].

ISO19107 (Spatial Schema) [1] provides conceptual schemas to describe and model real world objects as features, where cells in indoor space are a type of feature. The geometry package contains various classes for coordinate geometry used in IndoorGML. The mathematical functions which are used for describing the geometry of a cell depend on the type of coordinate reference system (CRS) which is used to define a spatial position.

The geometric representation of 2D or 3D features in indoor space is not a major focus of IndoorGML, since they are clearly defined by ISO 19107, CityGML, and IFC. However, for the sake of self-completeness, the geometry of 2D or 3D object may be optionally defined within IndoorGML according to the data model defined by ISO 19107. As illustrated in Figure 2 there are three options for representing geometry of a cell in indoor space;

1. External Reference (Option 1): Instead of explicit representation of geometry in IndoorGML, an IndoorGML document only contains external links (namely *c*.xlink, where *c* is a cell in IndoorGML) to objects defined in other data sets such as CityGML, where the referenced objects in external data set include geometric information. Then there must be 1:1 or *n*:1 mappings from cells in IndoorGML to corresponding objects in other dataset.
2. Geometry in IndoorGML (Option 2): Geometric representation of cell (namely *c*.geom, where *c* is a cell in IndoorGML) may be included within an IndoorGML document. It is GM_Solid in 3D space and GM_Surface in 2D space as defined in ISO 19107. Note that solid with holes or surface with holes are allowed in this standard.
3. No Geometry (Option 3): No geometric information is included in IndoorGML document.

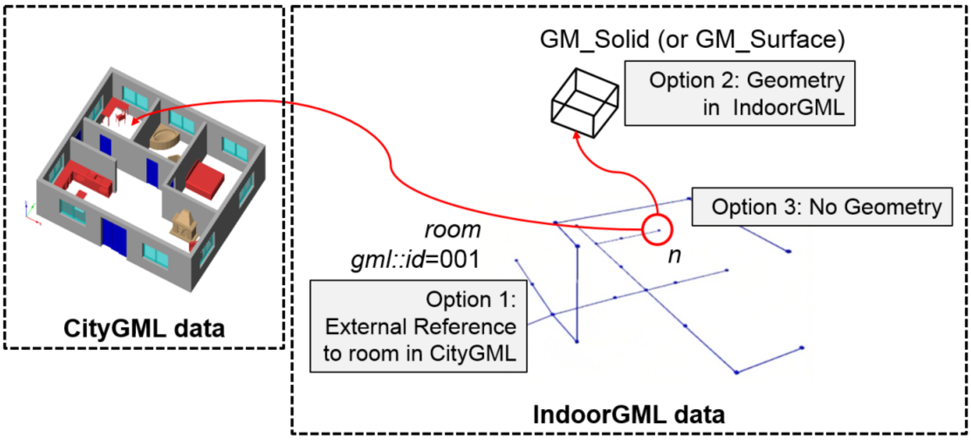


Figure 2: Three options to represent geometry in IndoorGML

When IndoorGML is independently used without referencing CityGML or IFC, it may contain the full 3D geometry of feature as defined in ISO 19107 but can also include only a 2D footprint. When IndoorGML is used with CityGML or IFC, we recommend making reference to the geometry defined in CityGML or IFC. Note that these options are not exclusive. For example, while IndoorGML document contain external references to the corresponding objects in CityGML, it also contains geometries of features by either the second or the third option. However, the second and third options are apparently exclusive.

7.1.5 Network Representation of Cellular Space

Topology is an essential component of cellular space and IndoorGML. A natural topology such as "neighbourhood, interior, disjoint and boundary" may be induced from geometry in Euclidean space. However, topological properties are not implicitly included in cellular space. Therefore, we need to explicitly describe the topological relationship in IndoorGML.

The Node-Relation Graph (NRG) [25] represents topological relationships, e.g., adjacency and connectivity, among indoor objects. The NRG allows abstracting, simplifying, and representing topological relationships among 3D spaces in indoor environments, such as rooms within a building. It can be implemented as a graph representing the adjacency, connectivity relationships without geometrical properties. It enables the efficient implementation of complex computational problems within indoor navigation and routing systems.

The Poincaré duality [8] provides a theoretical background for mapping indoor space to NRG representing topological relationships. A given indoor space can be transformed into a NRG in topology space using the Poincaré duality. This approach simplifies the complex spatial relationships between 3D by a combinatorial (or logical) topological network model [25]. According to Poincaré duality, a *k*-dimensional object in *N*-dimensional primal space is mapped to (*N*-k) dimensional object in dual space. Thus solid 3D objects in 3D primal space, e.g., rooms within a building, are mapped to nodes (0D object) in dual space. 2D surfaces shared by two solid objects is transformed into an edge (1D) linking two nodes in dual space. The nodes and edges in dual space form an adjacency graph, where the nodes and the edges of dual space represent cells and *adjacency relationships* between cells in primal space, respectively. Figure 3-a and Figure 3-b illustrate this duality transformation for the case where the primal space is 3D and 2D respectively. Note that the transformations from 1D object (curve) or 0D object (point) in 3D primal space are not included in IndoorGML since they are not considered as cells in most applications. But the transformation may be applied to 1D or 0D objects of 3D primal space in a similar way if it is required.

Then the adjacency graph 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.

Once adjacency relationships between cells are determined by Poincaré duality, other topological relationships can be defined from adjacency relationships based semantic information. An example of adjacency relationships in dual space is depicted by Figure 4. Figure 4-a shows a primal space with three cells including exterior cell (EXT), and boundaries between cells and the corresponding adjacency graph in dual space is given in Figure 4-b. Adjacency graph of dual space serves as a basic topological graph, since other topological graphs can be derived from the adjacency graph.

While no semantic information is used to generate adjacency graph in Figure 4, a different graph can be derived from adjacency graph by using semantic information. In Figure 5, boundaries are classified into walls and doors, then the graph in Figure 4-b becomes a different graph, called *connectivity graph*, which represents connectivity between cells as shown in Figure 5. Among adjacency relationships between cells in Figure 4-b, the edge of doors are included in the graph, while walls are removed from the graph since walls are not navigable. In a similar way, we may derive *accessibility graph* from adjacency graph by using constraint information as shown in Figure 6. If there is a constraint that the width of door D1 is 1.2 meters, then cell R1 is not accessible to tables bigger than 1.2 meters via door D1 and the accessibility graph becomes as Figure 8.

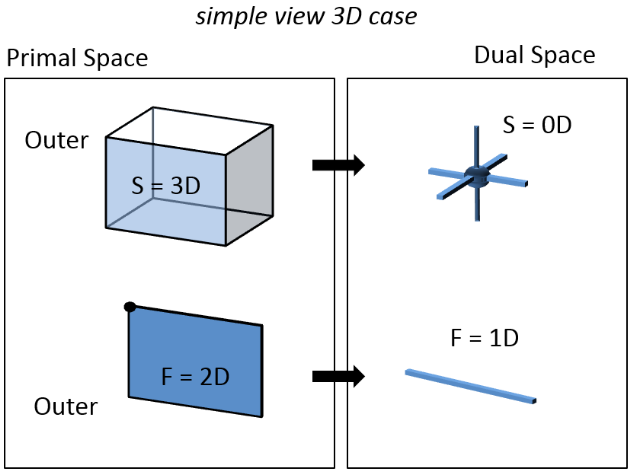


Figure 3-a: 3D Primal space case

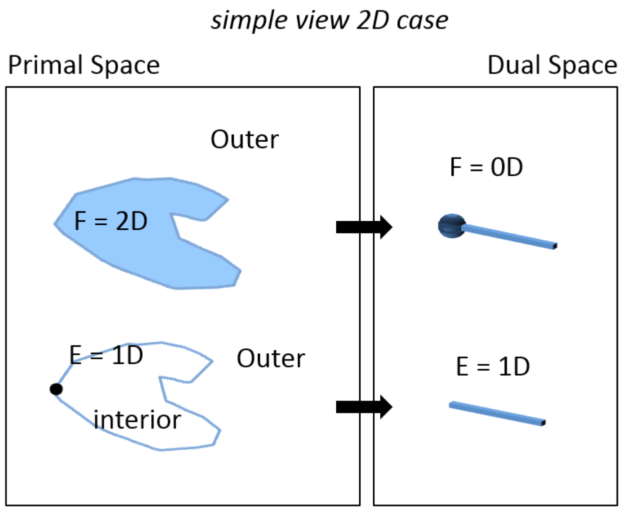


Figure 3-b: 2D Primal space case

Figure 3: Principles of Poincaré duality as shown by Lee [21]
(mathematical definition of Poincaré duality in [8])

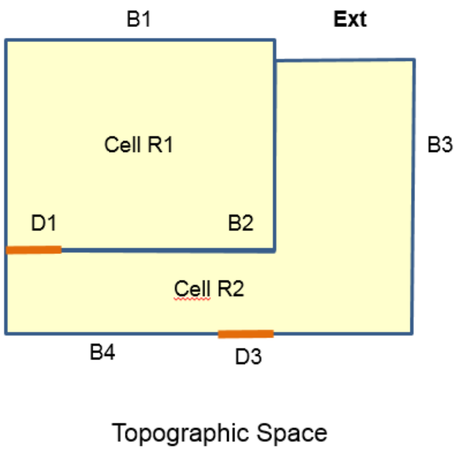


Figure 4-a. Primal space

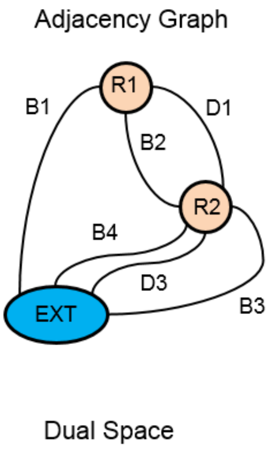
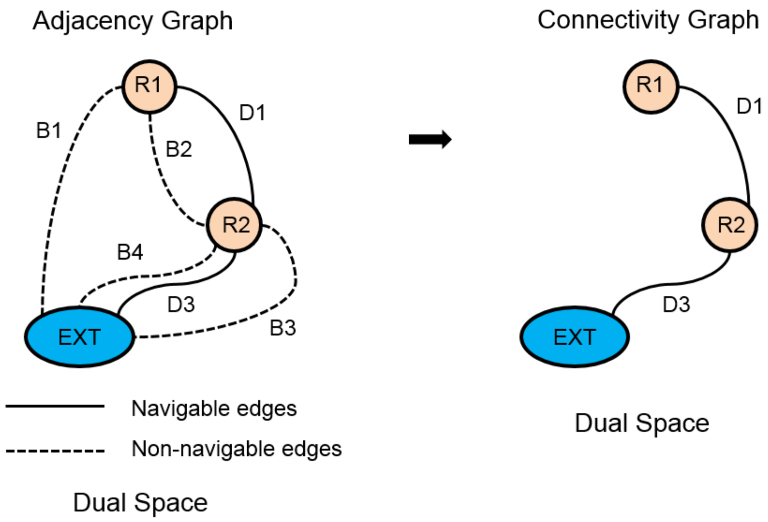


Figure 4-b. Adjacency graph in dual



space

Figure 5: Derivation of connectivity graph from adjacency graph

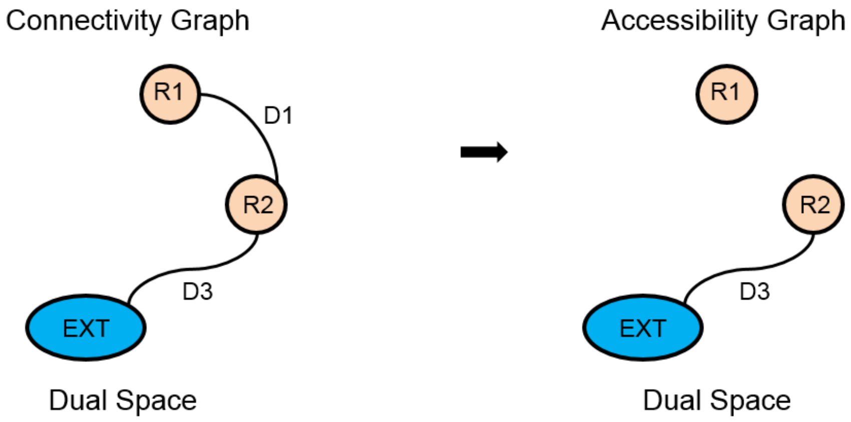


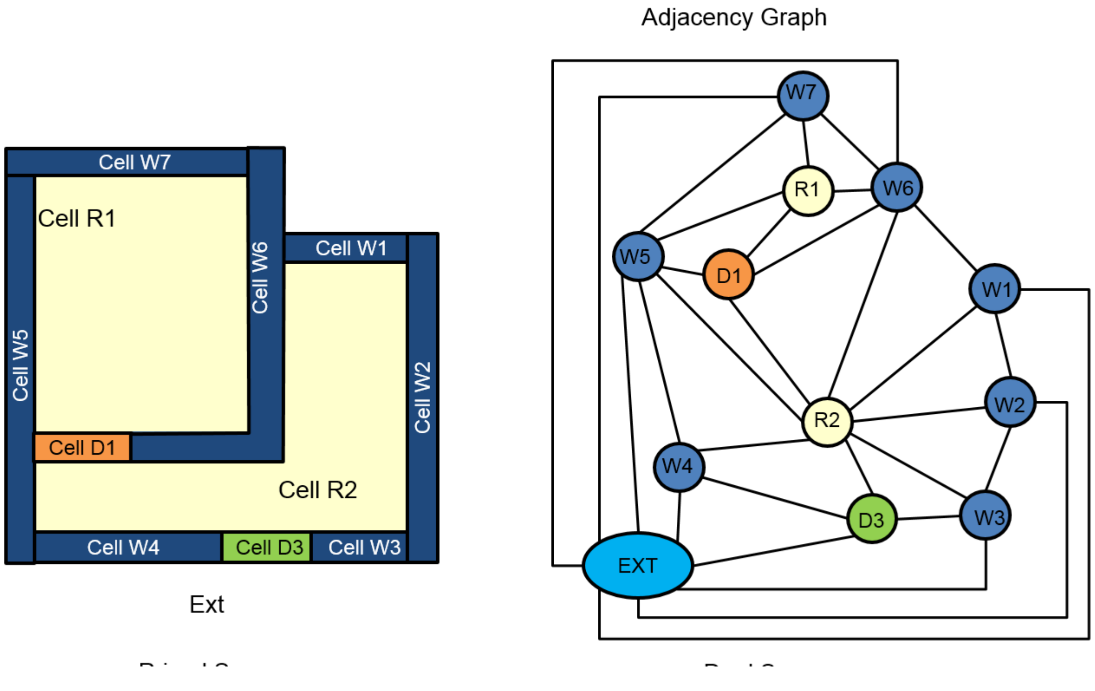
Figure 6: Derivation of accessibility graph

Connectivity graph G_{con} and accessibility graph G_{acc} are defined in similar ways as follows;

$G_{con} = (V, E_{con})$, where V is a set of nodes in dual space mapped from cells in 3D primal space and E_{con} is a set of edges representing connectivity between cells in primal space. Note that E_{con} □ E_{adj}.

$G_{acc} = (V, E_{acc})$, where V is a set of nodes in dual space mapped from cells in 3D primal space and E_{acc} is a set of edges representing accessibility between cells in primal space. Note that E_{acc} □ E_{adj}.

The walls and doors in the primal space are represented as boundaries in Figure 4-a, and they are accordingly mapped to edges in dual space as depicted in Figure 4-b. However, walls and doors may be also represented as cells with certain thickness depending on applications as shown in Figure 7. We call this representation *thick wall model* and the representation in Figure 4 is called *thin* (or *paper*) *wall model*. Then the NRG in dual space should be differently constructed as shown in Figure 7, where



walls and doors are mapped to nodes of dual space.

Figure 7: Adjacency graph for thick wall model

While the nodes and edges in NRG for the previous examples have no geometric properties, we may embed basic geometric data with nodes and edges such that each node has point coordinates and each edge has the coordinates of the starting, ending, and intermediate vertices. We call this geometrically embedded graph *geometric NRG*, while NRG without any geometric properties is called *logical NRG*. In geometric NRG, the geometries of node and edges are defined as GM_Point and GM_Curve of ISO 19107.

7.1.6 Multi-Layered Space Representation

A single indoor space is often semantically interpreted into different cellular spaces. For example, an indoor space is represented as a topographic cellular space composed of rooms, corridors, and stairs, while also being represented as different cellular spaces such as WiFi coverage cells and RFID sensor coverage cells respectively as shown in Figure 8. For this reason, IndoorGML supports multiple representation layers with different cellular spaces for an indoor space. Each semantic interpretation layer results in a different decomposition of the same indoor space where eachdecomposition forms a separate layer of cellular space as shown in Figure 8.

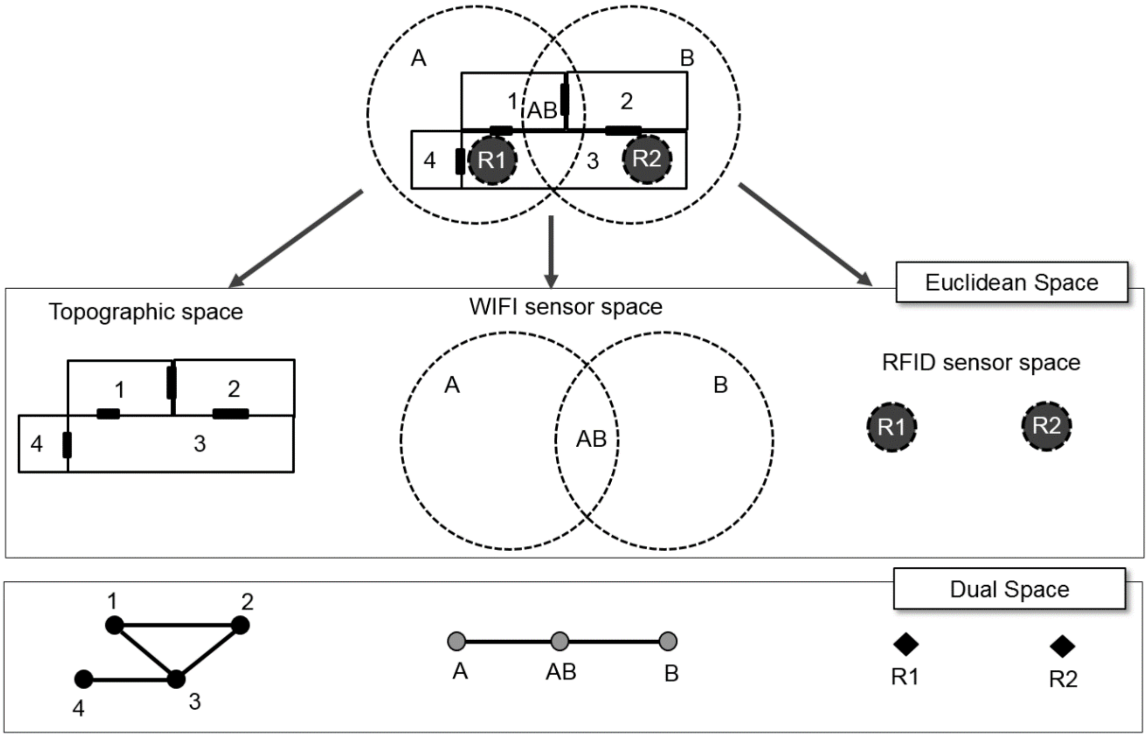


Figure 8: Example - Multiple Layered Space Representation

As shown in Figure 8, an indoor space is interpreted by three semantic representations – Topographic space layer, WiFi sensor space layer, and RFID sensor space. Since they are semantically different in terms of wheelchair movement, each layer forms a different cellular space and derived NRG for dual space.

This representation method with multiple cellular space layers is called *Multiple Layered Space Representation* (MLS Representation). The MLS representation is useful for many purposes. For example, we can represent the hierarchical structure of indoor space by MLS representation, where each level is represented as a single space layer and the relationships between two hierarchical levels are represented by inter-layer edges. Interlayer edges are explained in section 7.3. Another application example of MLS representation is indoor tracking with presence sensors such as RFID. Given an indoor space represented as topographic cellular space layer and RFID sensor coverage layer respectively, we can deduce the movement of a mobile object with a RFID tag by the sequence of RFID coverage cells and corresponding inter-layer space edges.

7.2 Structured Space Model

IndoorGML is based on two conceptual frameworks: the *Structured Space Model* and thye *Multi-Layered Space Model* (MLSM). The Structured Space Model defines the general layout of each space layer independent from the specific space model which it represents. Each layer is systematically subdivided into four segments as shown in see Figure 9.

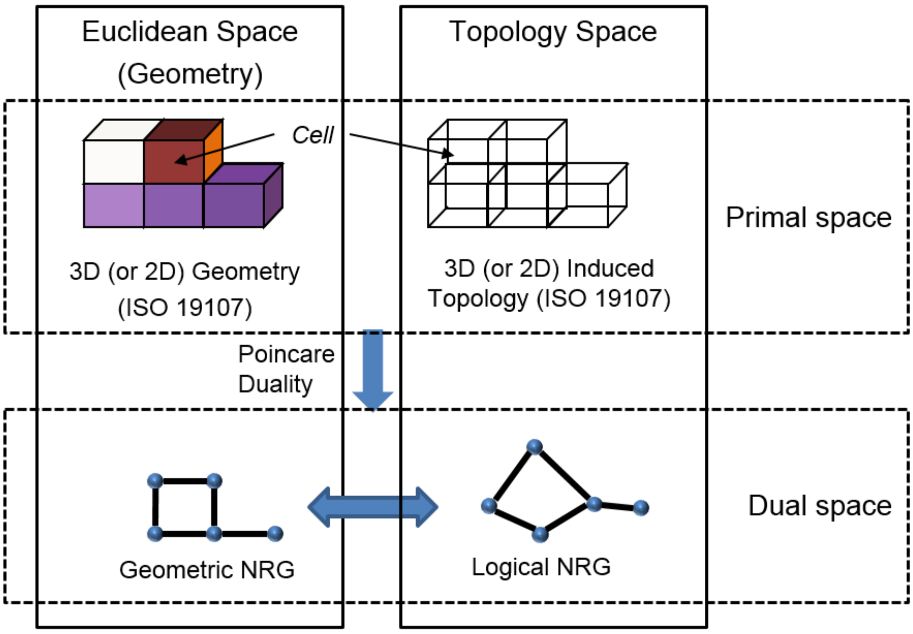


Figure 9 — Structure Space Model

Figure 9 illustrates the structured space model that allows for the distinct separation of primal space from dual space on the one hand, and geometry and pure topology on the other hand. This structure forms the basis for the framework proposed indoor space model.

The upper and the lower parts of Figure 9 are consistent with the rules of ISO 19107 for modelling geometrical features of real world phenomena. However, the transition from primal to dual space cannot be modelled or described via the ISO standard. Further, the topological relationships in IndoorGML such as adjacency and connectivity are not defined by means of the topology in ISO 19107 but by explicit associations within the IndoorGML data model.

In the Structured Space Model, topological relationships between 3D (or 2D) spatial objects are represented within topology space (i.e., the right part of Figure 9). By applying a duality transformation, the 3D cells in primal space are mapped to nodes (0D) in dual space. The topological adjacency relationships between 3D cells are transformed to edges (1D) linking pairs of nodes in dual space. Furthermore, the node of NRG is called *state* and the edge of NRG is called *transition*. The active state is represented by a node within the NRG and denotes the spatial area where the guided object is currently located. Once the object moves into a topologically connected area, another node within the NRG and thus a new active state is reached. The edge connecting both nodes represents the event of this state transition. The NRG representing topological relationships among 3D spatial objects in topological space is a logical *NRG*, while the NRG embedded to Euclidean IR^3 space is a geometric *NRG* as seen Figure 9.

The UML diagram depicted in Figure 10 shows the data model for the Structured Space Model perspective. A SpaceLayer represents a separate interpretation and a decomposition layer explained in section 7.1.6 and it is composed of State and Transition which represent nodes and edges of NRG for dual space, respectively. The NRG and state-transition diagram for each layer are realized by SpaceLayer. Note that the current version of IndoorGML supports logical NRG and geometric NRG for dual space.

As mentioned above, NRG as a part of the Structured Space Model is implemented in IndoorGML model. In dual space, the logical NRG in the lower right part of structured space model as seen in Figure 9 represents topological relationships among spaces in topological space, which is described as the cardinality of State and Transition to *Geometry* classes is 0 in Figure 10. When the cardinality is 1 in Figure 10, the topological model is implemented by coordinate space embedding of NRG (Geometric NRG), which is in the lower left part of structured space model as seen in Figure 9.

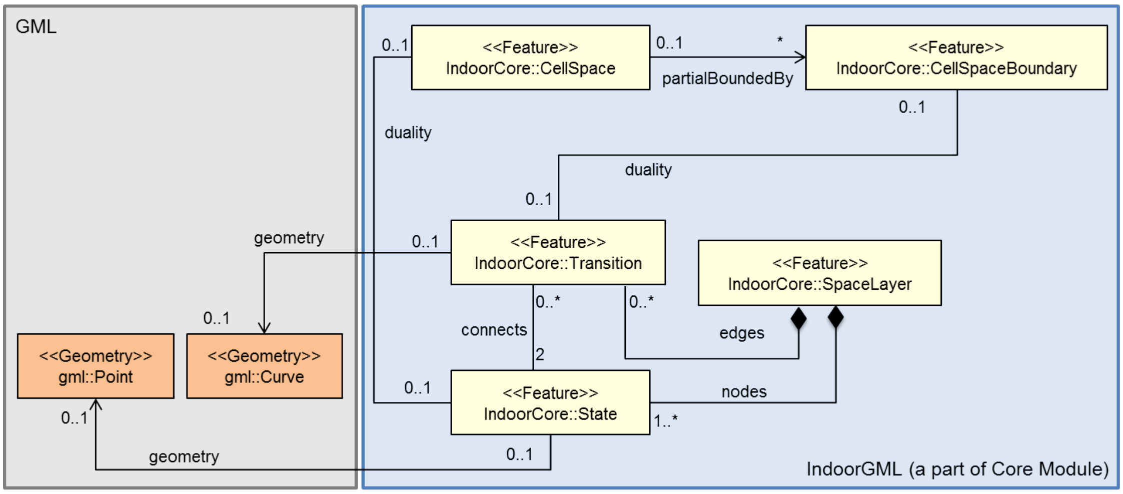


Figure 10: Implementation of Structure Space Model in IndoorGML

7.3 Multi-Layered Space Model

The concept of Structure Space model is further extended to *Multi-Layered Space Model* (MLSM). Multi-Layered Space Model provides an approach for combining multiple space structures for different interpretations and decomposition layers to support full indoor information services.

7.3.7 Multi-Layered Space Model – Key Concepts

A same indoor space is often differently interpreted depending on the application requirements as discussed in section 7.1.6. This results in different decompositions of a same indoor space, and each decomposition results in a specific NRG. For example, the layers for topographic space layer, WiFi sensor space layer, and RFID sensor space in Figure 8 form independent structured spaces and each layer results in three separate NRGs as depicted in Figure 11.

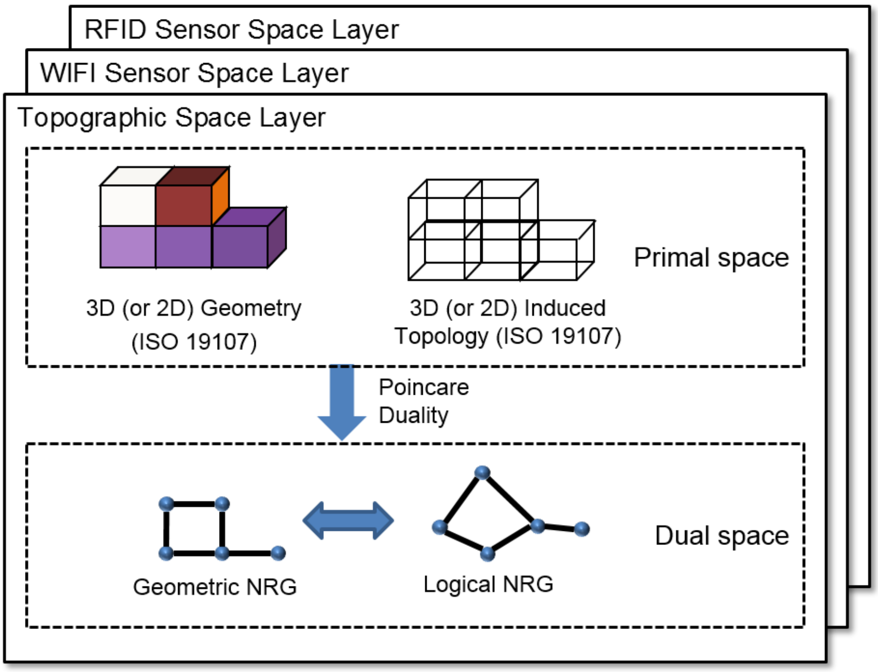


Figure 11: Multi-Layer combination of alternative space concepts

There may be several interpretations of a same indoor space. In most cases, the topographic space layer composed of geospatial features in indoor space such as rooms, corridors, staircases, and lift shafts are of the most important layer. For indoor positioning purposes, sensor space layer is also a fundamental one, where the notion of sensor space substantially differs from topographic space. The sensor space is rather decomposed according to signal characteristics such as propagation and signal coverage areas depending on different localization techniques such as WiFi or RFID which differ in signal propagation and signal coverage. There are other possible interpretations, and the number of layers is generally unbounded and any definition for space (e.g., security space, movement space, activity space, visual space etc.) can be given for a semantic modelling of indoor space, where each of them is defined in its own layer.

7.3.8 Inter-Layer Relations

Layers of multi-layered space model can be connected by inter-layer relations. As illustrated in Figure 11, there are three space layers, where each layer constitutes a NRG. In a topographic layer, the nodes represent the possible states of a navigating object and correspond to cells with volumetric extent in primal space (e.g. rooms) while the edges represent state transitions, i.e., the movement of an object from one space to another. They correspond to connectivity relations between the cells in primal space (e.g., neighboured rooms connected with a door). In the sensor space, NRG has a slightly different structure. The nodes represent again the cells with volumetric extend (e.g. the entire coverage space of a WiFi transmitter), while the edges represent the transition from one space to another based on the neighbouring WiFi coverage spaces. Since the layers cover the same real world space, the separated dual graphs can be combined into a multi-layered graph. Figure 12 illustrates overlaid space layers.

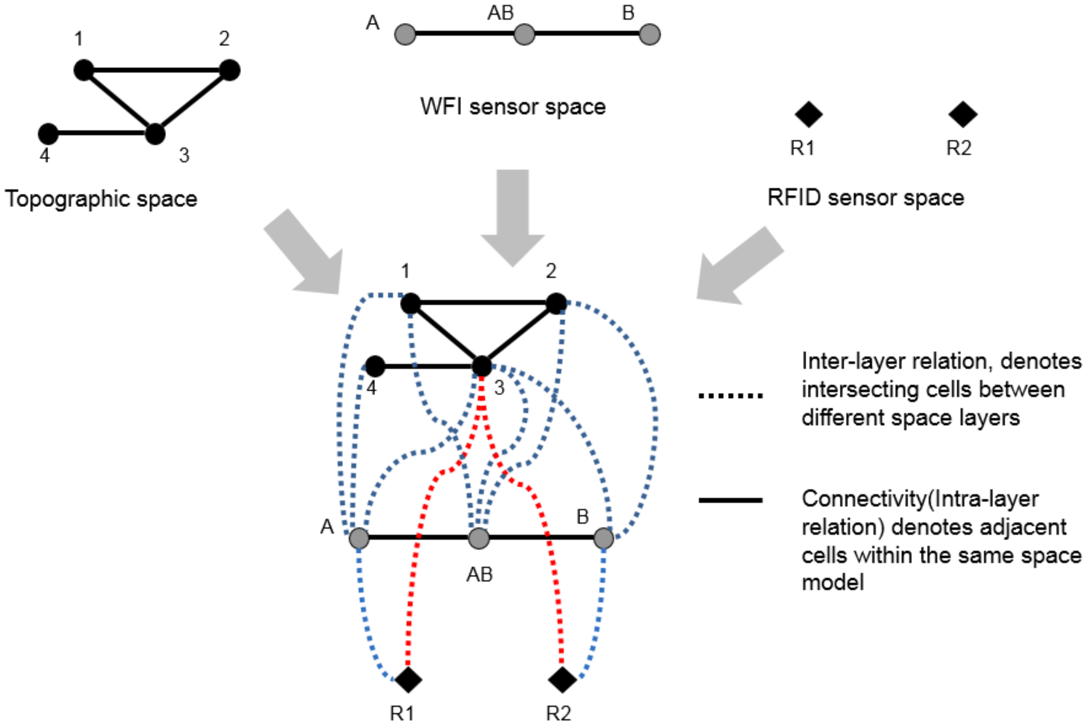

1
2
4
3
Topographic space
A
AB
B
WFI sensor space
R1
R2
RFID sensor space
1
2
4
3
A
B
AB
R1
R2
Inter-layer relation, denotes intersecting cells between different space layers
Connectivity(Intra-layer relation) denotes adjacent cells within the same space model

Figure 12: Overlaid space layers in dual space

If we assume that each space model, whether it is for topographic or sensor space, is based upon a non-overlapping partitioning of space, a navigating object can only belong to one cell at a time and thus always only one state may be active. Therefore, an object is at any given time exactly in one cell (named *state*) in each layer simultaneously. This overall state is thereby denoted by the combination of active states from all space layers.

However, only specific combinations of states from different layers are valid and can be active at the same time. The combinations are expressed by additional edges linking the nodes between different layers. These so called *joint edges* are derived by pairwise intersecting the cell geometries from different layers. A joint edge between two such nodes is inserted if the intersection of the interior of the two corresponding cells is non-empty. Therefore, the joint edges represent all relationships according to the eight relation model except “disjoint” and “touch” between two cells from different space layers defined in [14] and thus denote *inter-layer relationships*. Figure 13 illustrates the dual graphs of three space layers together with their inter-layer relationships.

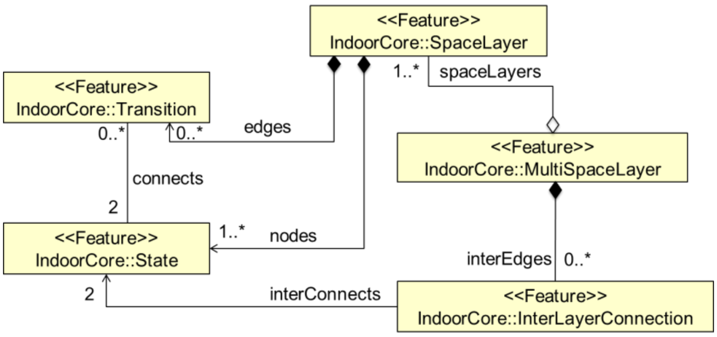


Figure 13: Implementation of Multi-Layer Space Model in IndoorGML

In IndoorGML, the space model for multi-layered space representation, called *multi-layered space model*, is implemented using the MultiSpaceLayer class. MultiLayeredGraph consists of SpaceLayers and InterLayerConnections as shown in Figure 13, while SpaceLayer represents each space layer (e.g. topographic space layer, sensor space layer, etc.) and it forms a NRG composed of objects from State and Transition. The inter-layer relationships are implemented by InterLayerConnection class. In Figure 12, {(1,A,Within), (4,A,Within), (3,A,Overlaps), (3,AB,Overlaps), (3,B,Overlaps), (2,B,Within), (1,AB,Overlaps), (3,AB,Overlaps), (A,R1,Contains), (B,R2,Contains), (3,R1,Contains), (3,R2,Contains)} are the set of instances from InterLayerConnection class, where each instance represents the relationship between two cells of different space layers of Figure 8. The MultiSpaceLayer is an aggregation of SpaceLayer and InterLayerConnection.

7.4 External reference

Since a main focus of IndoorGML is on the notion of cellular space and topological representation, an IndoorGML encoding may not contain geometries and detailed semantic information for indoor features. Instead, IndoorGML provides a method to reference an object in external dataset such as CityGML or an IFC. Depending on application areas, indoor features may have different geometric and semantic representation models. For example, indoor spaces are often represented by a grid model in the robotic navigation domain. By separating domain specific representation models from IndoorGML and providing external reference, a high level of flexibility can be achieved.

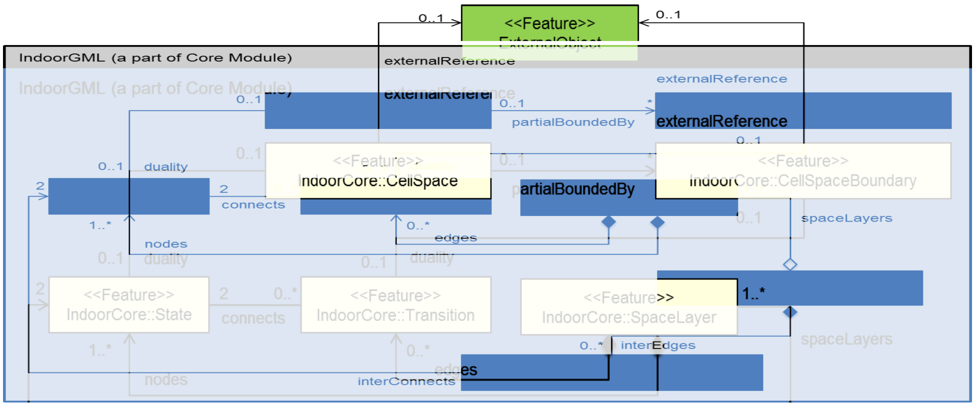


Figure 14: External References in IndoorGML

CellSpace and CellSpaceBoundary of IndoorGML core module depicted in Figure 18 may have external references to corresponding objects in external data sets. Note that the external reference is optional and can have at most one target object as shown by the cardinality in Figure 14.

Figure 15-(a) and Figure 15-(b) provide examples of external references to CityGML LoD 4 and SensorML respectively. For example, regarding the topography space layer, the subclasses of NavigableSpace can have an external reference to CityGML objects. The GeneralSpace has an external reference to bldg::Room of CityGML and the AnchorSpace and ConnectionSpace refer to bldg::Door in CityGML. bldg::BoundarySurface in CityGML is also referred by NavigableBoundary. Regarding the sensor space layer, all of subclasses of the NavigableSpace class also can have an external reference to sml::Component as shown in Figure 15-(b) which includes all the location and interface properties of any physical process. Note that NavigableSpace, AnchorSpace, and ConnectionSpace belong to IndoorGML navigation module, which will be explained in section 8.

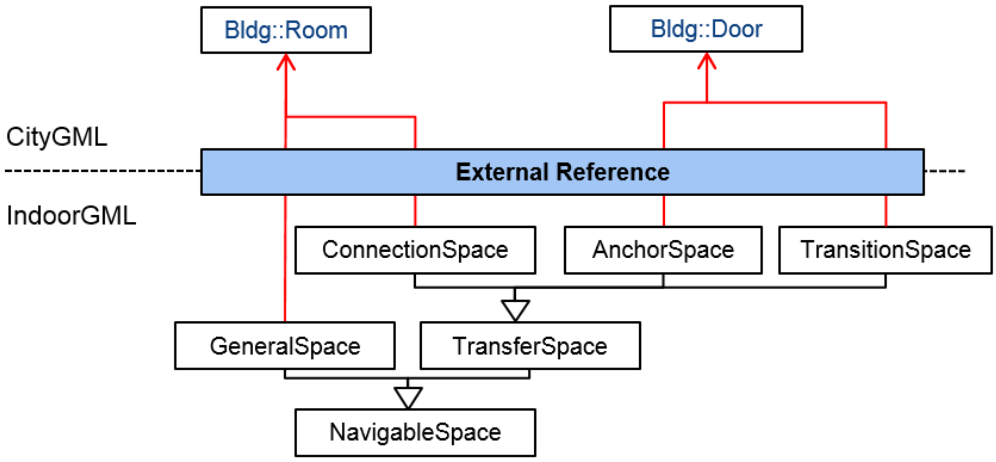


Figure 15-(a): External references to CityGML

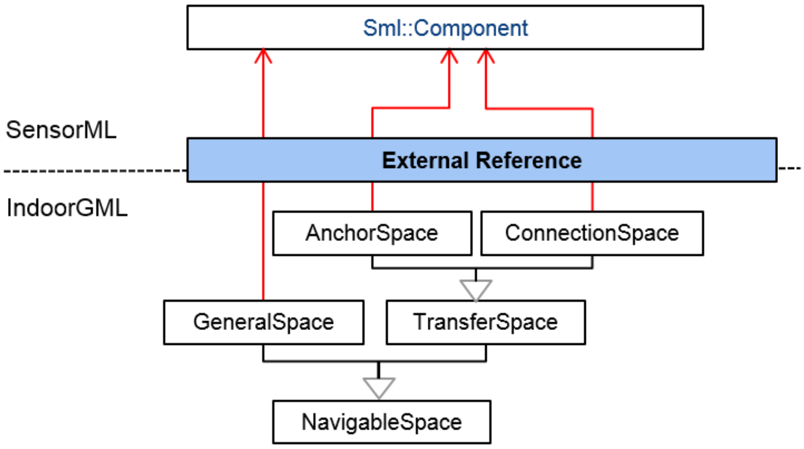


Figure 15-(b): External references to SensorML

7.5 Connection between Indoor and Outdoor Spaces

Connecting indoor and outdoor spaces is an important requirement for navigation and other applications. IndoorGML provides a concept to connect indoor and outdoor spaces by defining additional topology elements between indoor and outdoor spaces. Every indoor space contains at least one entrance, and it can be used to connect indoor and outdoor spaces. In IndoorGML, “entrance” is represented as a special node of the topological graph in indoor space, connecting indoor and outdoor as shown in Figure 16. We call this element *anchor node*, which differs from other node in the topological graph, since it may include additional information for converting a relative indoor CRS to an outdoorCRS.

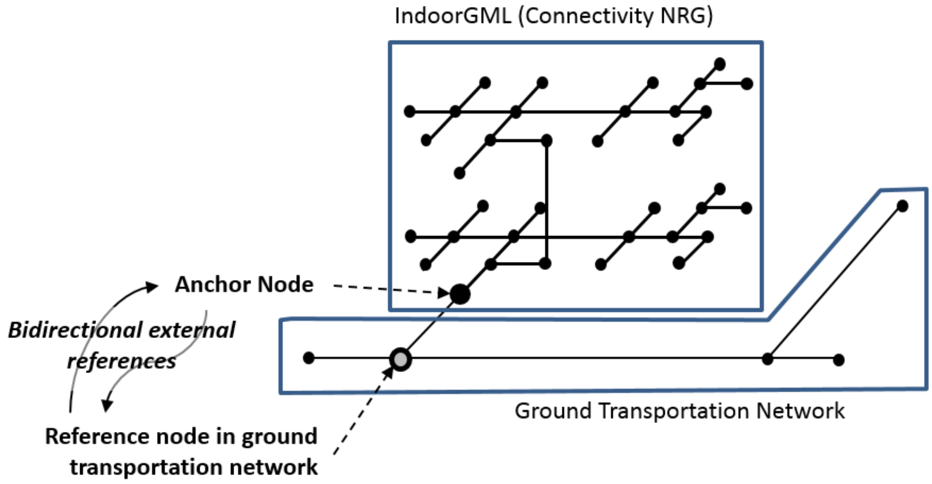

IndoorGML (Connectivity NRG)
Anchor Node
Bidirectional external references
Reference node in ground transportation network
Ground Transportation Network

Figure 16: Anchor Node Connecting Indoor and Outdoor Networks [27]

The anchor node element contains attributes to support the seamless conversion between indoor and outdoor spaces.

1. External reference to outdoor transportation network: The anchor node includes an external reference to a node in a ground transportation network, that is connected to the anchor node as shown in Figure 16. Note that the relationship betwteen anchor node and nodes in outdoor ground transportation is bidirectional. The anchor nodes are not only defined within IndoorGML document but also accessible from external data set such as the outdoor ground transportation network. For example, when a vehicle is entering to a building, we can get the IndoorGML document of the building via the external reference from the node in the ground transportation network.
2. Conversion parameters: In many cases, a relative CRS is applied to an indoor space and it is necessary to convert the coordinates of each point of indoor geometry according to the outdoor CRS. Anchor node therefore contains the parameters for transformation;

   - rotation origin point (x_0, y_0, z_0)
   - rotation angles (α, β, γ, along x, y, and z-axis),
   - rescaling factor (s_x, s_y, s_z), and
   - translation vector (t_x, t_y, t_z).

In cases where an absolute CRS is used for indoor space, the conversion parameters are not necessary. However anchor nodes are still useful not only for representing the connectivity between indoor and outdoor spaces but also facilitating seamless services for example by including the URI of radio map of the building for WiFi indoor positioning.

7.6 Subspacing

Indoor space often has hierarchical structures and a careful decomposition of an indoor space is required in many cases to reflect these hierarchical structures. A feature such as corridor or hall may be divided to accurately represent the geometric properties of indoor space based on the connectivity relationships among space objects. IndoorGML supports hierarchical subspacing by Multi-Layered Space Model explained in section 7.3.

The subspacing by the first option is shown in Figure 17. In the case of a corridor in Figure 17-(a), node n_6 in the NRG representing a corridor within the indoor space (Figure 17-(a), Figure 17-(b)) is considered as a consolidated *Master Node*, which is transformed to a sub-graph preserving connectivity relationship among the compartmentalized spaces of the corridor (Figure 17-(c)). It means that node n_6 in the original NRG is converted into $n_{6\text{-}1}$ and $n_{6\text{-}2}$ and edge e_1 in Figure 17-(c)) in the transformed NRG, which is a sub-graph representing a two-dimensional shape such as a hallway.

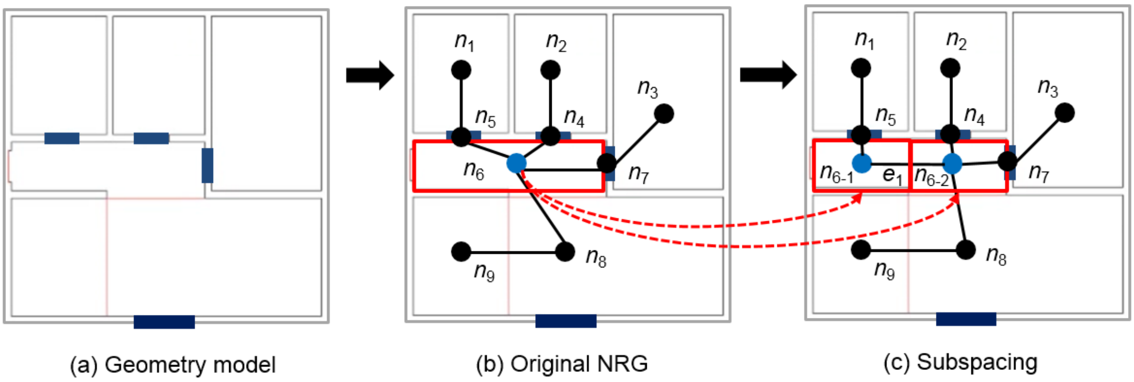
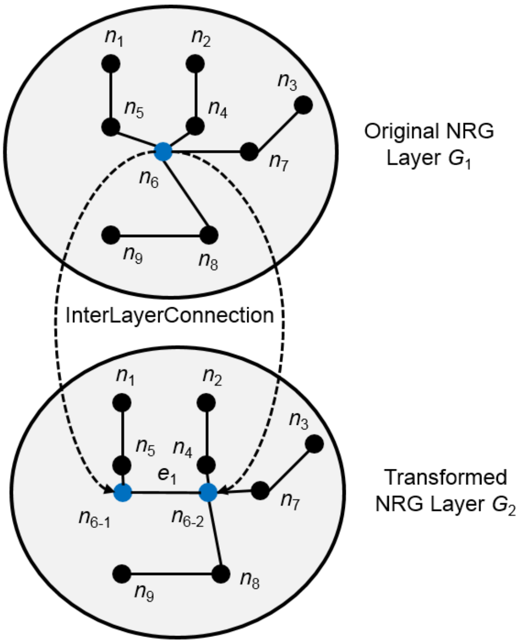


Figure 17: Example of subspacing – Connectivity NRG

IndoorGML supports the subspacing by means of multi-layered space model to reflect the hierarchical structure of indoor space as shown in Figure 18. The NRG G_1 is the original graph layer with node $n_{6,}$ while G_2 is a transformed graph layer with partitioned nodes $n_{6\text{-}1}$ and $n_{6\text{-}2}$. Then the hierarchical structure is represented by means of inter-layer connection of the multi-layered space model. Note that there are default one-to-one inter-layer connection between $G_1.n_k$, $G_2.n_k$ except n_6 as shown in Figure 18.



$\{(G_1.n_k, G_2.n_k, \text{equal}) \mid k \neq 6\}$: Defalut *InterLayerConnection*
$\{(G_1.n_6, G_2.n_{6\text{-}1}, \text{contains}), (G_1.n_k, G_2.n_k, \text{contains})\}$: *InterLayerConnection* for subspacing

Figure 18: Multi-layered space model for subspacing

In the case where hierarchical subspacing is not required, we may simply replace a space with subdivided spaces and describe the adjacency or connectivity relationships between subdivided spaces. For example in Figure 17, we just replace n_6 with $n_{6\text{-}1}$, and $n_{6\text{-}2}$ and append edges connecting them.

7.7 Modularization

Following the guidance in the OGC's policy "The Specification Model — A Standard for Modular specifications [15]", IndoorGML is split into a core module and extensions that have a mandatory dependency on the core. Therefore, the IndoorGML data model is thematically decomposed into a Core module and Thematic extension modules (see Figure 19). The core module comprises the basic concept and each extension module covers a specific thematic field such as navigation applications (e.g. pedestrians, wheel-chair, and robot). Each IndoorGML module is specified by an XML Schema definition file and is defined within an individual and globally unique XML target namespace as shown in Table 1. According to dependency relationships among modules, each module may, in addition, import namespaces associated to such related IndoorGML modules.

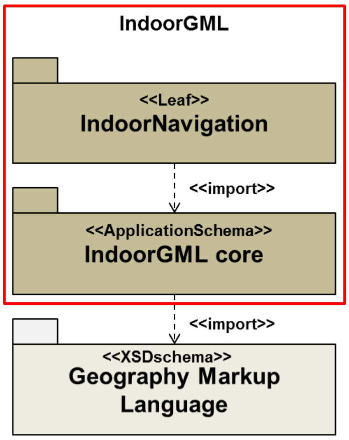


Figure 19: UML Package diagram

Module Name	IndoorGML core
XML Namespace Identifier	http://www.opengis.net/indoorgml/1.0/core
XML Schema File Name	indoorgmlcore.xsd
Namespace Prefix	IndoorCore
Module Description	The IndoorGML core module defines the basic components of IndoorGML data model. It includes the schema definitions of basic classes for cells, dual spaces and multi-layered space model. It is an application schema of GML 3.2.1.

Module Name	IndoorGML navigation
XML Namespace Identifier	http://www.opengis.net/indoorgml/1.0/navigation
XML Schema File Name	indoorgmlnavi.xsd
Namespace Prefix	IndoorNavi
Module Description	The IndoorGML navigation module defines the semantic extension of IndoorGML core module for indoor navigation. It defines the schema definitions of the classes for indoor navigation.

Table 1: IndoorGML Modules and Namespace Identifiers

The IndoorGML core module defines the basic concepts and component of the IndoorGML data model. Except semantic modelling, the aspects explained in section 7.1 are reflected into the core module. The extension modules contain the semantic modelling aspect (see section 7.1.3) of IndoorGML. Based on the IndoorGML core module, the extension module contains a logically separate thematic component of the IndoorGML data model. This version of IndoorGML introduces the first thematic extension module: *IndoorNavigation* module.

The dependency relationships among IndoorGML's modules are illustrated in Figure 19. Each module is represented by a package. 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. For IndoorGML modules, a dependency occurs where one schema <import>s other schema and accordingly the corresponding XML namespace. In the following sections the modules are described in detail.

8 IndoorGML Core Module for the Multi-Layered Space Model

In the preceding sections, we discussed the basic concepts and overall IndoorGML model. In the following sections, we present the detailed data model and XML schemas required to encode indoor sp-atial information following the terms, definitions, and relationships discussed in section 7. The UML diagram depicted in Figure 20 shows IndoorGML core module data model based on the multi-layered space model. The data model defines the classes and relations needed to describe the geometric and topological representations of each space layer in primal and dual spaces presented in section 7. The XML Schema for this data model is also defined as an application schema of GML 3.2.1. (see also Annex A)

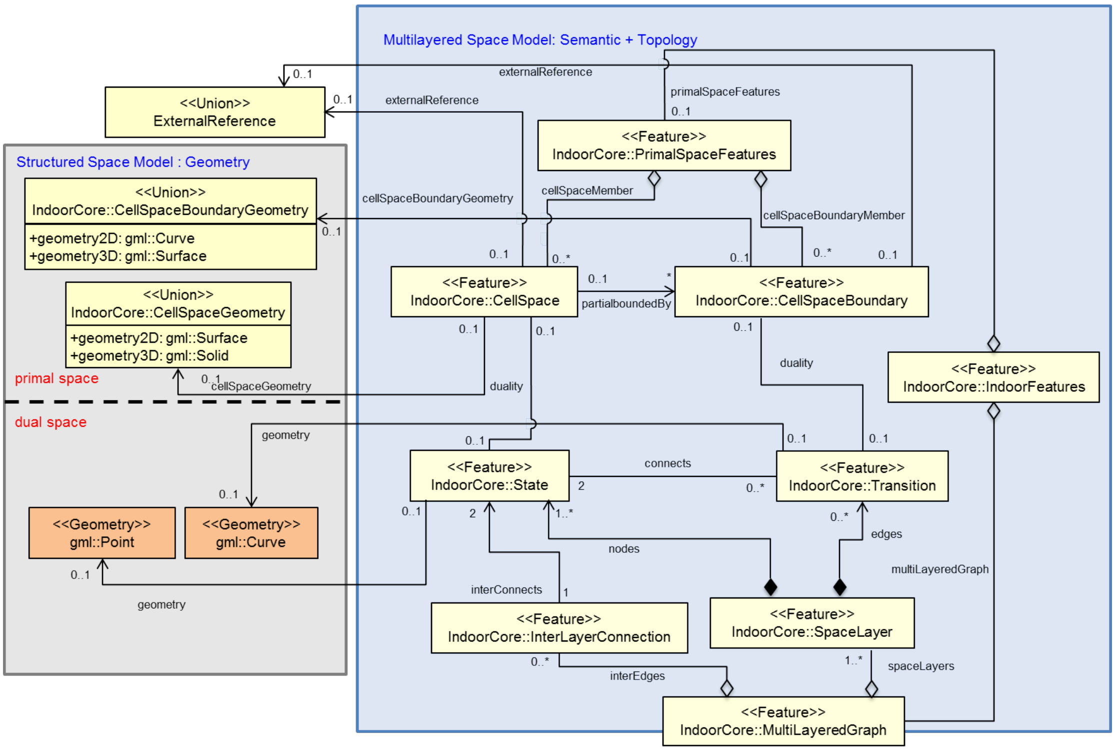


Figure 20: UML diagram of Multi-layered Space Model

(based on ISO19100 standards family and GML3.2.1) [figure updated]

The classes in Figure 20 are arranged according to the Structured Space Model explained in section 7.2 (cf. Figure 9). For each layer, its geometry and topology representations are modeled in primal and dual spaces based upon ISO 19107.

Some classes (CellSpace, CellSpaceBoundary) in IndoorGML may have a geometric object that is represented in 2D or 3D spaces. The PrimalSpaceFeatures consisting of CellSpace and CellSpaceBoundary classes represents real world objects in accordance with the notions of geographic features defined by ISO19109. A CellSpace is a semantic class corresponding to one space object in Euclidean primal space of one layer. Accordingly, a CellSpaceBoundary is used to semantically describe the boundary of each space object. Both classes are defined as interface classes which connect the Multi-layered Space Model to exterial geometric models.

According to the dimension of space, the CellSpace class is represented as gml:Solid or gml:Surface and CellSpaceBoundary class is represented as gml:Surface or gml:Curve in 3D or 2D space, respectively. In other words, when is represented on a 2 dimensional space, the CellSpace mapped on gml:Surface. If CellSpace is represented in the three dimensional space, it mapped on gml:Solid.

The separate layers of the Multi-layered Space Model are represented by the class SpaceLayer. A layer aggregates State and Transition objects. SpaceLayer can be connected through the InterLayerConnection class which represents a gml:Curve in Euclidean space connecting two states from separate layers. The inter-layer connections (InterLayerConnection) together with in intra-layer connections (State and Transition) finally generate the MultiLayeredGraph.

The IndoorGML core module defines the basic concepts and components of the multi-layered space model. The multi-layered space model allows for the coherent combination of different decompositions of space according to different semantics. A decomposition of space is represented by a separate space layer which is systematically subdivided into primal and dual space on the one hand and geometry and topology on the other hand. The multi-layered space model is generally considered as a conceptual framework for the generic representation of space and their topological relationships. Especially, the IndoorGML core module provides to represent the topological relationships of indoor spaces in dual space.

The UML diagram of IndoorGML’s core module is depicted in Figure 21, for XML schema definition see blow and annex A. The multi-layered space model consists of nine classes; State, Transition, CellSpace, CellSpaceBoundary, InterLayerConnection, SpaceLayer, MultilayeredGraph, PrimalSpaceFeatures and IndoorFeatures class.

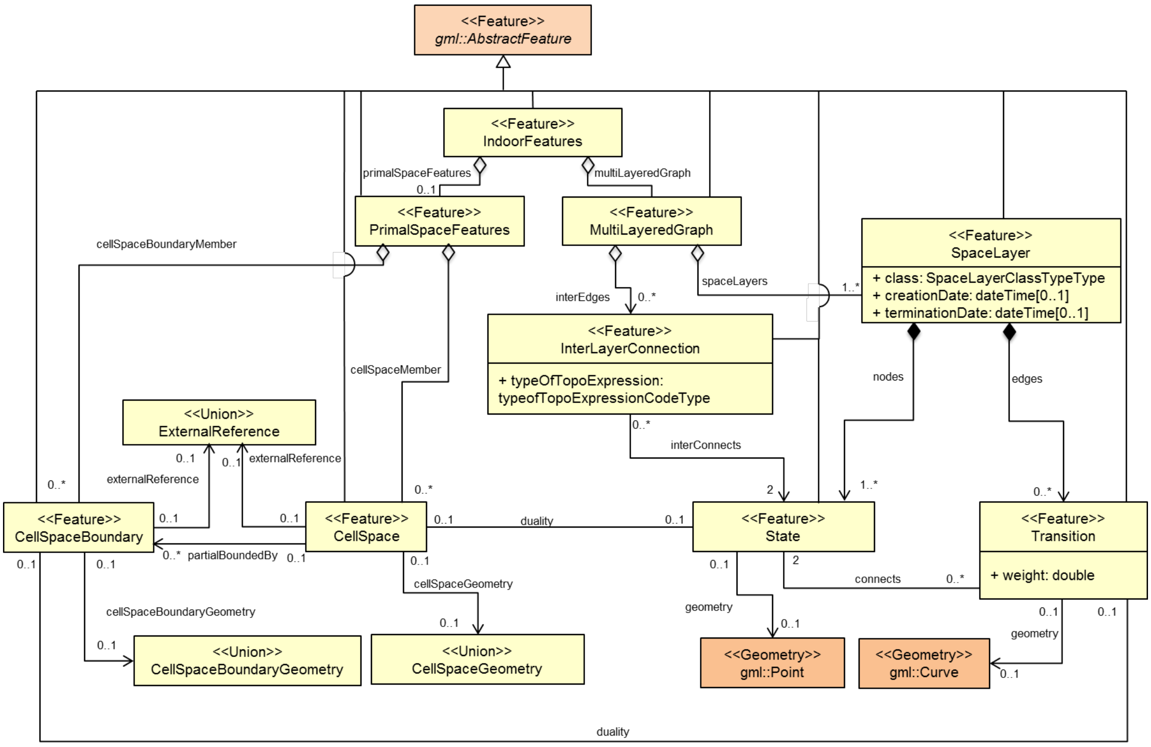


Figure 21: UML diagram of IndoorGML's core module(Multi-Layered Space Model) [figure updated]

The XML namespace of the IndoorGML core module is identified by the Uniform Resource Identifier (URI) http://www.opengis.net/indoorgml/core/1.0. Within the XML Schema definition of the core module, this URI is also used to identify the default namespace.

8.1 <State>

State represents a node in dual space of MLSM. State may be an isolated node, i.e. not connected to another State. Within the topographic space layer, a space can be associated with a room, corridor, door, etc. within a building of the primal space. It is represented geometrically as Point in IndoorGML. It also has association with the corresponding CellSpace class which represents a space in primal space (or referred to a geometry object in primal space). The attribute duality – which can only occur once – represents an association with the CellSpace. The connects element describes the relation of a Transition and two State object associated with the Transition itself on one layer (the same layer). The attribute connects can occur from zero to many times. For the geometrical representation of a State, a Point geometric primitive object defined in the GML is used.

```
<xs:element name="State" type="StateType" substitutionGroup="gml:AbstractFeature"/>
<!-- ====================================================================== -->
<xs:complexType name="StateType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="duality" type="CellSpacePropertyType" minOccurs="0" maxOccurs="1"/>
                <xs:element name="connects" type="TransitionPropertyType" minOccurs="0"
                               maxOccurs="unbounded"/>
                <xs:element name="geometry" type="gml:PointPropertyType" minOccurs="0" maxOccurs="1"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ====================================================================== -->
<xs:complexType name="StatePropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="State"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

8.2 <Transition>

Within a dual graph structure of one layer, Transition is an edge that represents the adjacency or connectivity relationships among nodes representing space objects in primal space. Transition always connects two States. In the topographic space layer, a Transition can be associated with a boundary of a room in the primary space. The attribute connects represents States that are boundary objects of Transition. The attribute duality represents an association relation with CellSpaceBoundary class. The attribute weight can be used for applications in order to deal with the impedance representing absolute barriers in transportation problems. For the geometrical representation of a Transition, a Curve geometric primitive object from the GML is used.

```
<xs:element name="Transition" type="TransitionType" substitutionGroup="gml:AbstractFeature"/>
<!-- ====================================================================== -->
<xs:complexType name="TransitionType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="weight" type="xs:double"/>
                <xs:element name="connects" type="StatePropertyType"  minOccurs="2"  maxOccurs="2"/>
                <xs:element name="duality" type="CellSpaceBoundaryPropertyType"  minOccurs="0"
                               maxOccurs="1"/>
                <xs:element name="geometry" type="gml:CurvePropertyType"  minOccurs="0" maxOccurs="1"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ====================================================================== -->
<xs:complexType name="TransitionPropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="Transition"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

8.3 <CellSpace>

CellSpace is a base class for representing the indoor space. The class CellSpace contains properties for space attributes and purely geometric representations of space. CellSpace also has references to thematic objects in external data sources; the geometrical representation in primal Euclidean space is referenced by xlink. The attribute externalReference is used for the reference of an object to its corresponding object in an external data set. Each CellSpace is associated with a geometry object which can be represented as several geometry primitive types such as 2D and 3D.

```
<xs:element name="CellSpace" type="CellSpaceType" substitutionGroup="gml:AbstractFeature"/>
<!-- ====================================================================== -->
<xs:complexType name="CellSpaceType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="cellSpaceGeometry" type="CellSpaceGeometryType" minOccurs="0"
                            maxOccurs="1"/>
                <xs:element name="duality" type="StatePropertyType"  minOccurs="0" maxOccurs="1"/>
                <xs:element name="externalReference" type="ExternalReferenceType" minOccurs="0"
                            maxOccurs="unbounded"/>
                <xs:element name="partialboundedBy" type="CellSpaceBoundaryPropertyType" minOccurs="0"
                            maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ====================================================================== -->
<xs:complexType name="CellSpacePropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="CellSpace"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!-- ====================================================================== -->
<xs:complexType name="CellSpaceGeometryType">
    <xs:choice>
        <xs:element name="Geometry3D" type="gml:SolidPropertyType"/>
        <xs:element name="Geometry2D" type="gml:SurfacePropertyType"/>
    </xs:choice>
</xs:complexType >
<!-- ====================================================================== -->
<xs:complexType name="ExternalReferenceType">
    <xs:sequence>
        <xs:element name="informationSystem" type="xs:anyURI" minOccurs="0" maxOccurs="1"/>
        <xs:element name="externalObject" type="externalObjectReferenceType"/>
    </xs:sequence>
</xs:complexType>
<!-- ====================================================================== -->
<xs:complexType name="externalObjectReferenceType">
    <xs:choice>
        <xs:element name="name" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="uri" type="xs:anyURI"/>
    </xs:choice>
</xs:complexType>
```

8.4 <CellSpaceBoundary>

CellSpaceBoundary is used to semantically describe the boundary of each geographical feature in space. The geometry of the CellSpaceBoundary normally will be described by a Surface geometric object in 3D Models. The attribute externalReference is used for the reference of a geometric object to its corresponding object in an external data set. Each CellSpaceBoundary is associated with a geometry primitive object which can be represented as gml:Surface or gml:Curve.

```
<xs:element name="CellSpaceBoundary" type="CellSpaceBoundaryType"
            substitutionGroup="gml:AbstractFeature"/>
<!-- ====================================================================== -->
<xs:complexType name="CellSpaceBoundaryType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureType">
         <xs:sequence>
            <xs:element name="duality" type="TransitionPropertyType" minOccurs="0" maxOccurs="1"/>
            <xs:element name="cellSpaceBoundaryGeometry"
                      type="CellSpaceBoundaryGeometryType" minOccurs="0"  maxOccurs="1"/>
            <xs:element name="externalReference" type="ExternalReferenceType" minOccurs="0"
                       maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
<!-- ====================================================================== -->
<xs:complexType name="CellSpaceBoundaryGeometryType">
   <xs:choice>
      <xs:element name="geometry3D" type="gml:SurfacePropertyType"/>
      <xs:element name="geometry2D" type="gml:CurvePropertyType"/>
   </xs:choice>
</xs:complexType >
<!-- ====================================================================== -->
<xs:complexType name="CellSpaceBoundaryGeometryPropertyType">
   <xs:sequence minOccurs="0">
      <xs:group ref="CellSpaceBoundaryGeometry"/>
   </xs:sequence>
   <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

8.5 <SpaceLayer>

SpaceLayer class is a top class used to represent a structured space model. A SpaceLayer represents each space layers such as topography, sensor, security space, etc. A SpaceLayer aggregates State and Transition which are directly associated with the corresponding geometry classes to represent dual space. To represent spatial objects in primal space, a SpaceLayer also aggregates CellSpace and CellSpaceBoundary which are directly associated with the corresponding geometry classes. The SpaceLayer class has attributes which are class, function, usage, creationDate and terminationDate. The attribute class – which can only occur once - represents a general classification of the layer. With the function and usage attributes, norminal and real functions of a space layer can be described.

The creationDate and terminationDate attributes can be used to describe the chronology of the layer. The points of time refer to real world times.

The SpaceLayer class has nodes and edges which represent a set of States and Transitions on the layer.

Figure 22 depicted an example of topographic space layer. Each space in real world mapping to State and the relationship between spatial objects is represented by Transition in a dual space of topographic space layer.

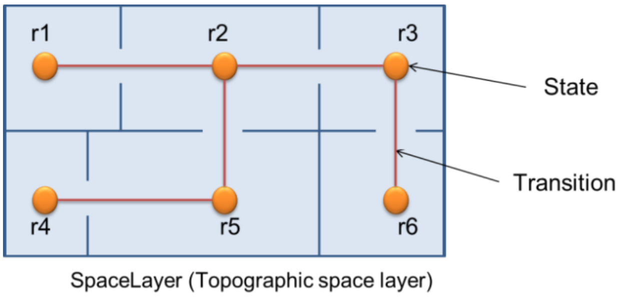


Figure 22: Example of *SpaceLayer* (Topographic space layer)

```
<xs:element name="SpaceLayer" type="SpaceLayerType" substitutionGroup="gml:AbstractFeature"/>
<!-- ====================================================================== -->
<xs:complexType name="SpaceLayerType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="terminationDate" type="xs:dateTime"  minOccurs="0" maxOccurs="1"/>
                <xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="creationDate" type="xs:dateTime"  minOccurs="0" maxOccurs="1"/>
                <xs:element name="class" type="SpaceLayerClassTypeType"  minOccurs="0" maxOccurs="1"/>
                <xs:element name="nodes" type="NodesType" minOccurs="1" maxOccurs="unbounded"/>
                <xs:element name="edges" type="EdgesType" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ====================================================================== -->
    <xs:complexType name="SpaceLayerPropertyType">
        <xs:sequence minOccurs="0">
            <xs:element ref="SpaceLayer"/>
        </xs:sequence>
        <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xs:complexType>
<!-- ====================================================================== -->
<xs:complexType name="NodesType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="stateMember" type="StateMemberType" minOccurs="1"
                               maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
            <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ====================================================================== -->
<xs:complexType name="StateMemberType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureMemberType">
            <xs:sequence minOccurs="0">
                <xs:element ref="State"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ==================================================================== -->
<xs:complexType name="EdgesType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="transitionMember" type="TransitionMemberType" minOccurs="0"
                                maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
            <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ==================================================================== -->
<xs:complexType name="TransitionMemberType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureMemberType">
            <xs:sequence minOccurs="0">
                <xs:element ref="Transition"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ==================================================================== -->
<xs:simpleType name="SpaceLayerClassTypeType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="TOPOGRAPHIC"/>
        <xs:enumeration value="SENSOR"/>
        <xs:enumeration value="LOGICAL"/>
        <xs:enumeration value="TAGS"/>
        <xs:enumeration value="UNKNOWN"/>
    </xs:restriction>
</xs:simpleType>
```

[A typography error is corrected. The one of dulplicated item “LOGICAL” is deleted.]

8.6 <InterLayerConnection>

InterLayerConnection class has two States. Each State is defined in different SpaceLayers. Intersecting the geometries of the layer combinations provides an edge if the intersection of their interior geometries is non-empty; the edge, called InterLayerConnection, may express one of following spatial relationships: contains, overlaps, or equals. InterLayerConnection is denoted relationships between States in different SpaceLayers. The interConnects attribute represents the States belonged into the InterLayerConnection object. The ConnectedLayers attribute represents the SpaceLayers which include each State of InterConnects. The typeOfTopoExpression attribute represents a relationship between two layers. The comment attribute can contain an additional description for the InterLayerConnection.

```
<xs:element name="InterLayerConnection" type="InterLayerConnectionType"
substitutionGroup="gml:AbstractFeature"/>
<!-- ==================================================================== -->
<xs:complexType name="InterLayerConnectionType">
    <xs:complexContent>
```

```
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="typeOfTopoExpression" type="typeOfTopoExpressionCodeType"
                                minOccurs="0" maxOccurs="1"/>
                <xs:element name="comment" type="xs:string" minOccurs="0" maxOccurs="1"/>
                <xs:element name="interConnects" type="StatePropertyType" minOccurs="2" maxOccurs="2"/>
                <xs:element name="ConnectedLayers" type="SpaceLayerPropertyType" minOccurs="2"
                                maxOccurs="2"/>

            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ====================================================================== -->
<xs:simpleType name="typeOfTopoExpressionCodeType">
    <xs:union memberTypes="typeOfTopoExpressionCodeEnumerationType
                                    typeOfTopoExpressionCodeOtherType"/>
</xs:simpleType>
<!-- ====================================================================== -->
<xs:simpleType name="typeOfTopoExpressionCodeEnumerationType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="CONTAINS"/>
        <xs:enumeration value="OVERLAPS"/>
        <xs:enumeration value="EQUALS"/>
        <xs:enumeration value="WITHIN"/>
        <xs:enumeration value="CROSSES"/>
        <xs:enumeration value="INTERSECTS"/>
    </xs:restriction>
</xs:simpleType>
<!-- ====================================================================== -->
<xs:simpleType name="typeOfTopoExpressionCodeOtherType">
    <xs:restriction base="xs:string">
        <xs:pattern value="other. \w{2,}"/>
    </xs:restriction>
</xs:simpleType>
```

8.7 <MultilayeredGraph>

MutliLayeredGraph is a key element of IndoorGML Core Module and is used to represent the Multi-layered Space Model. It aggregates SpaceLayers and InterLayerConnections. MutliLayeredGraph class consists of SpaceLayers and InterLayerConnections. MutliLayeredGraph has all the nodes (States) from all *n* layers (SpaceLayers) are included but they are separated into *n* partitions which are connected by the Transition. Furthermore the graph also contains the state-transition edge (InterLayerConnection). The MultiLayeredGraph contains a set of SpaceLayer as spaceLayers and a set of InterLayerConnection as interEdges.

```
<xs:element name="MultiLayeredGraph" type="MultiLayeredGraphType" substitutionGroup="gml:AbstractFeature"/>
<!-- ====================================================================== -->
<xs:complexType name="MultiLayeredGraphType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="spaceLayers" type="SpaceLayersType" minOccurs="1" maxOccurs="unbounded"/>
                <xs:element name="interEdges" type="InterEdgesType" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

```
<!-- ====================================================================== -->
<xs:complexType name="SpaceLayersType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="spaceLayerMember" type="SpaceLayerMemberType" minOccurs="1"
                                maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ====================================================================== -->
<xs:complexType name="SpaceLayerMemberType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureMemberType">
            <xs:sequence minOccurs="1">
                <xs:element ref="SpaceLayer"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ====================================================================== -->
<xs:complexType name="InterEdgesType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="interLayerConnectionMember" type="InterLayerConnectionMemberType"
                                minOccurs="1" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ====================================================================== -->
<xs:complexType name="InterLayerConnectionMemberType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureMemberType">
            <xs:sequence minOccurs="0">
                <xs:element ref="InterLayerConnection"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

8.8 <PrimalSpaceFeatures>

PrimalSpaceFeatures is a feature class for representing primal space features such as rooms. It consists of CellSpaces as cellSpaceMember and CellSpaceBoundary as cellSpaceBoundaryMember.

```
<xs:element name="PrimalSpaceFeatures" type="PrimalSpaceFeaturesType"
                substitutionGroup="gml:AbstractFeature"/>
<!-- ====================================================================== -->
<xs:complexType name="PrimalSpaceFeaturesType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="cellSpaceMember" type="CellSpaceMemberType " minOccurs="0"
```

```
                        maxOccurs="unbounded"/>
            <xs:element name="cellSpaceBoundaryMember" type="CellSpaceBoundaryMemberType " minOccurs="0"
                        maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

8.9 < IndoorFeatures>

IndoorFeatures is a root element of IndoorGML Core Module to represent the indoor space. It is an aggregated element with PrimalSpaceFeatures and MultiLayeredGraph. The IndoorFeatures contains both a set of CellSpace and CellSpaceBoundary as PrimalSpaceFeatures and a set of SpaceLayer as MultiLayeredGraph.

```
<xs:element name="IndoorFeatures" type="IndoorFeaturesType" substitutionGroup="gml:AbstractFeature"/>
<!-- ========================================================================== -->
<xs:complexType name="IndoorFeaturesType">
   <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureType">
         <xs:sequence>
            <xs:element name="primalSpaceFeatures" type="PrimalSpaceFeaturesType" minOccurs="0"
                        maxOccurs="1"/>
            <xs:element name="multiLayeredGraph" type=" MultiLayeredGraphPropertyType" minOccurs="0"
                        maxOccurs="1"/>
         </xs:sequence>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

8.10 Requirements for conformance

This clause specifies the conformance requirements for the IndoorGML Core Module. Although most of conformance requirements are already presented in the model and XML schema of the IndoorGML Core Module, and certain complementary conformance requirements are explicitly given in this clause;

> Requirement 1: The dimensions of CellSpace and CellSpaceBoundary should be consistent. If the geometry type of CellSpace is gml:Surface, then that of CellSpaceBoundary *shall* be gml:Curve. And if the geometry type of CellSpace is gml:Solid, then that of CellSpaceBoundary must be gml:Surface.
>
> Requirement 2: The instances of CellSpace belonging to the same instance of SpaceLayer *shall* not overlap.
>
> Requirement 3: When a CellSpace instance is divided into a set of subspaces, the subspace instances *shall* not belong to the same SpaceLayer instance of the original CellSpace instance but form a new SpaceLayer instance.
>
> Requirement 4: Every instance of InterLayerConnection *shall* connect two State instances, each of which belongs to different space layers.

9 Data Model of the Indoor Navigation Module

The Indoor navigation model provides semantic information for indoor space to support indoor navigation applications. Space features are classified into two groups: NavigableSpace and NavigableSpaceBoundary. NavigableSpace represents all indoor spaces (e.g. rooms, corridors, windows, stairs) that can be used by a navigation application. NavigableBoundary represents all features that connect the navigation spaces (e.g. door). Navigable Spaces and Navigable Boundaries are mapped on CellSpace and CellSpaceBoundary families. These are associated with corresponding classes such as State and Transition in IndoorGML Core Module.

The UML diagram of the conceptual indoor navigation model is depicted in Figure 23. The classes coloured in beige belong to the *IndoorGMLCore* UMLpackage and the classes colured in orange belong to the *GML* UMLpackage.

<<Feature>>
gml::AbstractFeatu
multiLayeredGraph
primalSpaceFeatures
<<Fe
IndoorCore::M
spaceLayers
interConnects
externalReference
cellSpaceBoundaryGeometry
duality
geometry
referencedTransition
connects
path
startNode
<<Feature>>
IndoorCore::CellSpace
<<Union>>
<<Feature>>
IndoorNavi::NavigableBoundary
<<Feature>>
IndoorNavi::NonNavigal
<<Feature>>
rCore::CellSpaceBoundary

Figure 23: UML diagram of Conceptual Indoor Navigation Model [figure updated]

The IndoorGML Navigation Module furthermore specifies in detail the generic concepts of the core module which are required in the context of indoor navigation. This might include the addressing/georeferencing schemas of indoor spaces, the concepts on communicating and visualizing navigable route sections, and the introduction of additional navigation constraints such as temporal access constraints as opening hours, or constraints resulting from material properties of the navigation path.

The XML namespace of the IndoorGML Navigation Module is identified by the Uniform Resource Identifier (URI) http://www.opengis.net/indoorgml/navigation/1.0. Within the XML Schema definition of the IndoorGML Navigation Module, this URI is also used to identify the default namespace.

The following Figure 24 shows an UML diagram of IndoorGML Navigation Module.

<<Feature>>
IndoorNavi::GeneralSpac
<<Featur
IndoorNavi::Tras
<<Feature>>
<<Feature>>
<<Union>>
cellSpaceGeometry
0..1
<<Feature>>
IndoorCore::CellSpa
duality
0..1
0..1
2
geometry
0..1
referencedState
<<Union>>
connects
0..1
0..*
referencedTransition
startRouteNode
endRouteNode
routeNodes
path
1..*
2..*
geometry2D
cellSpaceBoundaryGeometry
duality
<<Feature>>
0..1
gml::A
Ind

Figure 24: UML diagram of IndoorGML Navigation module [figure updated]

Figure 25 shows an example of indoor space mapped to IndoorGML Navigation module classes.

For mapping indoor space features to IndoorNavigation module classes, space features have to be classified by functions and usages of space. Each space class has attributes for functions, usages, and classes of space. Misspellings or different names for the same concept or feature can create problems in data interoperability. In order to overcome these interoperability errors, IndoorGML provides External CodeLists (see the Annex D) to specify the attribute values including space class types, space function types and space usage types. The CodeLists are from OmniClass (Table 13 and Table 14) created and used by the North American architectural, engineering and construction (AEC) industry.

The External CodeLists can be extended or redefined by users. They can have references to existing models. For example, GeneralSpace codes can be defined by the CityGML's code lists instead of referencing to IndoorGML's predefined values.

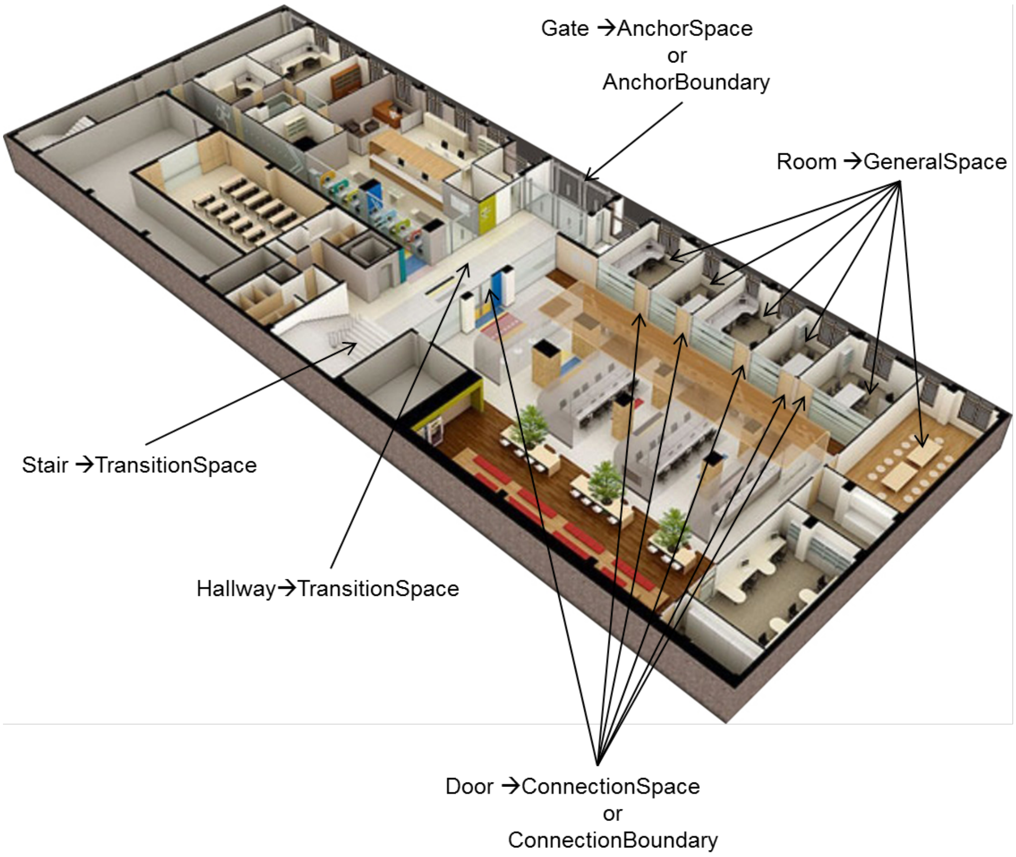
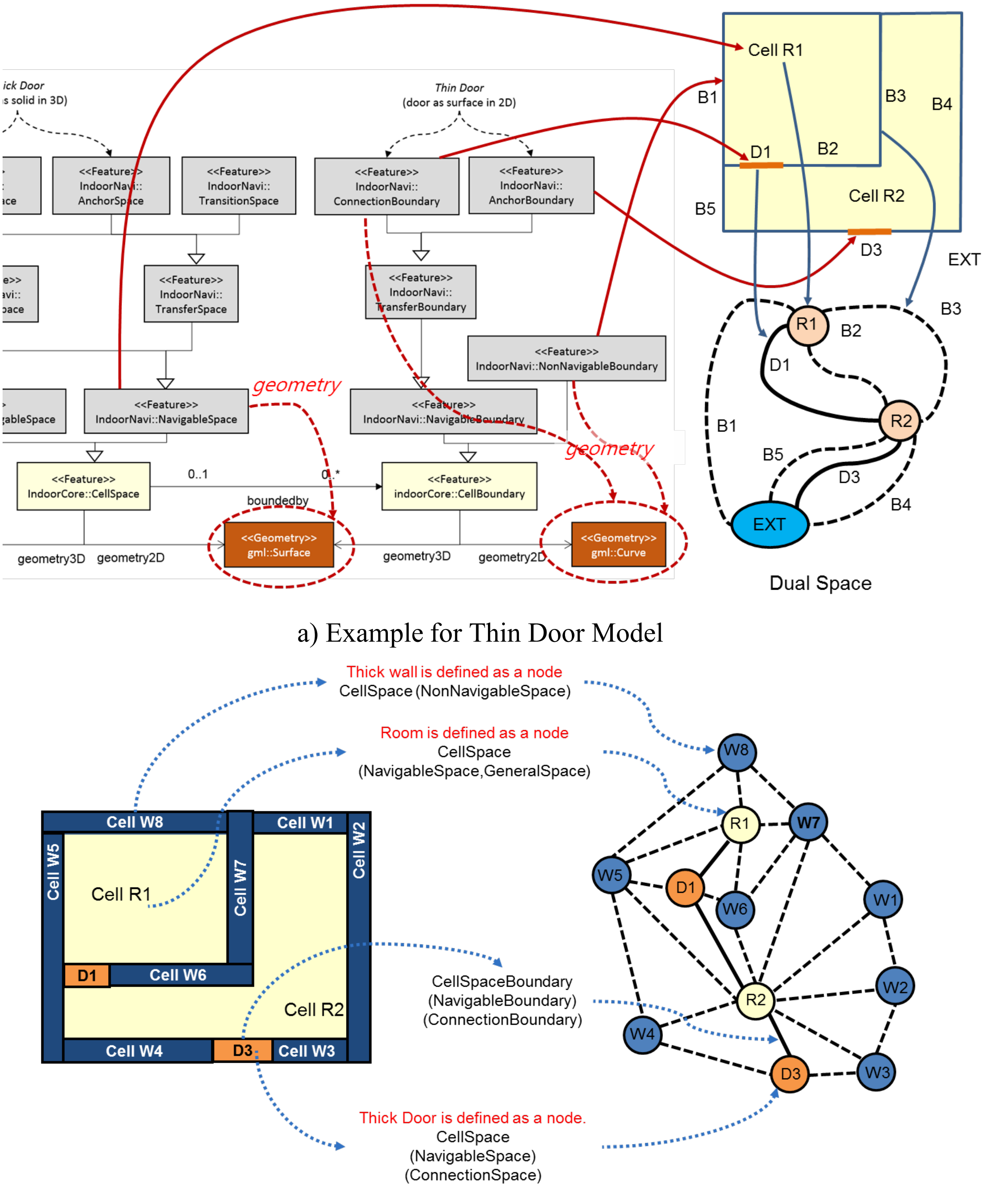


Figure 25: Indoor space mapped to IndoorGML Navigation module classes

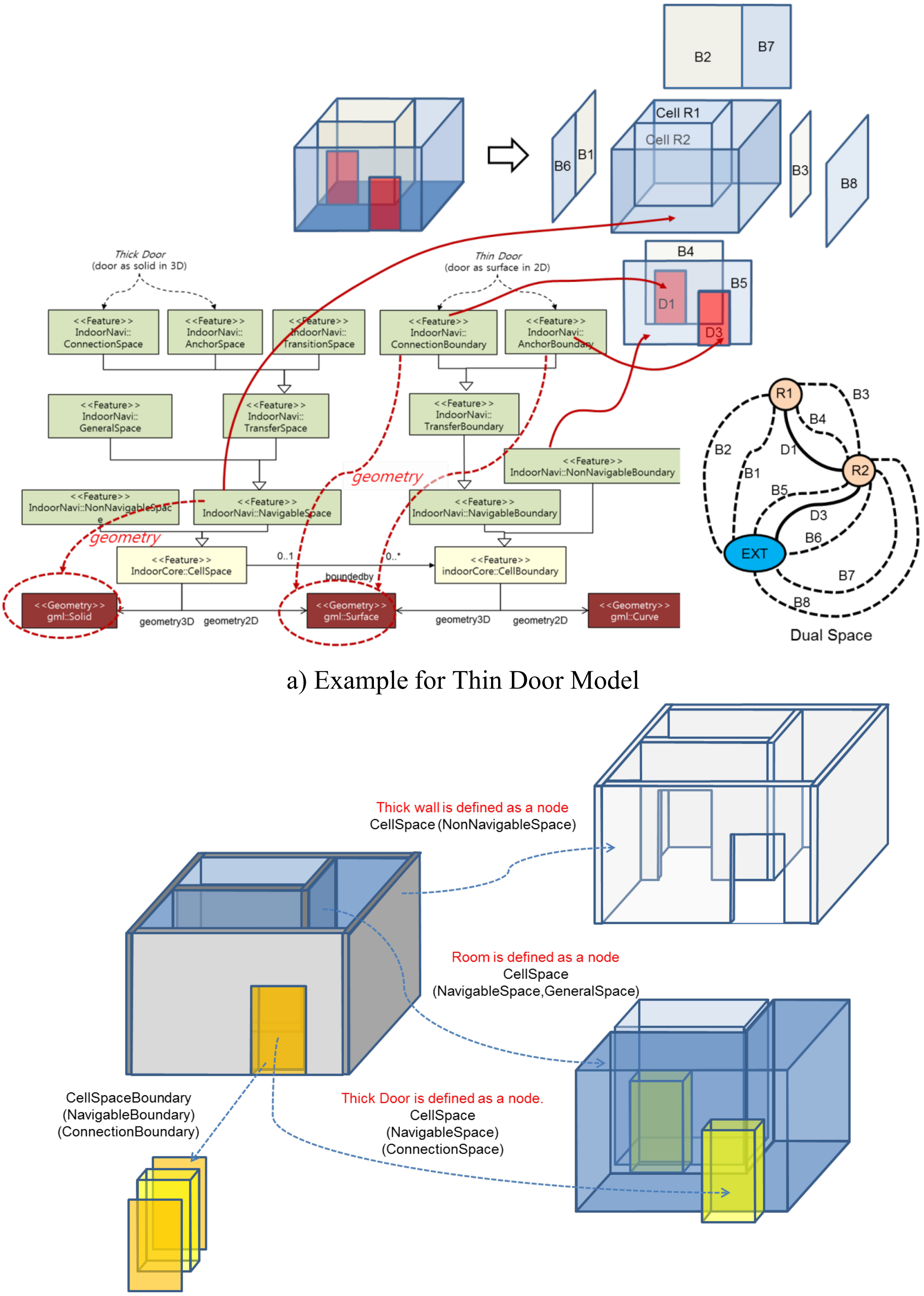
Figure 26 shows an example of geometric mapping in 2D Space. The Room feature mapped to CellSpace and represented as gml:Surface. In the Thin Door Model, the Door feature mapped to CellSpaceBoundary and represented as gml:Curve as seen in the Figure 26-a). In this case, a door is mapped to Transition on dual space. However, in the Thick Door model as seen in the Figure 26-b), the Door feature mapped to CellSpace and represented as gml:Surface. In this case, a door represented as a State in dual space.



a) Example for Thin Door Model

b) Example for Thick Door Model

Figure 26: Realization of CellSpace and CellSpaceBoundary for 2D Space Model


B2
B7
Cell R1
Cell R2
B6
B1
B3
B8
B4
B5
D1
D3
Thick Door
(door as solid in 3D)
Thin Door
(door as surface in 2D)
<<Feature>> IndoorNavi::ConnectionSpace
<<Feature>> IndoorNavi::AnchorSpace
<<Feature>> IndoorNavi::TransitionSpace
<<Feature>> IndoorNavi::ConnectionBoundary
<<Feature>> IndoorNavi::AnchorBoundary
<<Feature>> IndoorNavi::GeneralSpace
<<Feature>> IndoorNavi::TransferSpace
<<Feature>> IndoorNavi::TransferBoundary
<<Feature>> IndoorNavi::NonNavigableBoundary
<<Feature>> IndoorNavi::NonNavigableSpace
<<Feature>> IndoorNavi::NavigableSpace
<<Feature>> IndoorNavi::NavigableBoundary
geometry
geometry
<<Feature>> IndoorCore::CellSpace
0..1
0..*
boundedby
<<Feature>> indoorCore::CellBoundary
<<Geometry>> gml::Solid
<<Geometry>> gml::Surface
<<Geometry>> gml::Curve
geometry3D
geometry2D
geometry3D
geometry2D
R1
R2
EXT
B3
B4
B2
D1
B1
B5
D3
B6
B7
B8
Dual Space
Thick wall is defined as a node
CellSpace (NonNavigableSpace)
Room is defined as a node
CellSpace
(NavigableSpace,GeneralSpace)
CellSpaceBoundary
(NavigableBoundary)
(ConnectionBoundary)
Thick Door is defined as a node.
CellSpace
(NavigableSpace)
(ConnectionSpace)

a) Example for Thin Door Model

b) Example for Thick Door Model

Figure 27: Realization of CellSpace and CellSpaceBoundary for 3D Space Model

Figure 27 illustrates an example of geometric mapping for 3D Space. As seen in Figure 27-a) and b), the CellSpace is realized as gml:Solid in 3D space model. If the Door feature is represented as thin door, the geometry of door is represented by gml:Surface and the door is mapped to CellSpaceBoundary as shown as Figure 27-a).

In this case, a door is mapped to Transition on dual space. While in 3D data model, the door is represented by gml:Solid and mapped to CellSpace as shown as Figure 27-b). In this case, a door is represented as a State in dual space.

For example, the class CellSpace can be related to a Room in GityGML [11] or an IfcSpace in IFC. The class CellSpaceBoundary can be related to a _BoundarySurface feature in CityGML (e.g. WallSurface, ClosureSurface, InteriorWallSurface, etc) or an IfcWall in IFC. The geometric spaces and their topological relationships in the NRG are realized as gml:Point and gml:Curve.

For mapping indoor space features to IndoorGML Navigation module classes, space features have to be classified by classes, functions and usages of space. Each space class has attributes for functions, usages, and classes of space. Because misspellings or different names for the same notion brings the problems in data interoperability, in order to overcome the problems, an example of external code list is provided in annex D to specify the attribute values including space class types, space function types and space usage types. The code lists are proposed from OmniClass (Table 13 and Table 14) created and used by the North American Architectural, Engineering and Construction (AEC) industry.

The external code lists will be extended or redefined by users. They can have references to existing models. For example, GeneralSpace codes can be defined by the CityGML's code lists instead of referencing to IndoorGML's predefined values.

9.1 <NavigableSpace>

The NavigableSpace class denotes a space that users can move freely in. It has two subclasses GeneralSpace and TransferSpace. The subclasses are classified depending on the purpose of the space. The compartmentalized spaces such as corridor, lobby, hallway, big room are represented as NavigableSpace. Especially, on 3D data mode, door is represented as NavigableSpace as shown as Figure 26-b).

A geometry of NavigableSpace is represented as gml:Solid on 3D data model or gml: Surface on 2D data model as shown as Figure 26 and Figure 27.

The class attribute represents the classification of the NavigableSpace. The different functions and usage of NavigableSpace can be represented as function and usage.

```
<xs:element name="NavigableSpace" type="NavigableSpaceType" substitutionGroup="IndoorCore:CellSpace"/>
<!-- ====================================================================== -->
<xs:complexType name="NavigableSpaceType">
    <xs:complexContent>
        <xs:extension base="IndoorCore:CellSpaceType">
            <xs:sequence>
                <xs:element name="class" type="gml:CodeType"/>
                <xs:element name="function" type="gml:CodeType"/>
                <xs:element name="usage" type="gml:CodeType"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

9.2 <NonNavigableSpace>

The NonNavigableSpace class represents the space that is occupied by obstacles. On 3D data model, a wall is typical NonNavigablespace as shown as Figure 26-b). It is not implemented on XML schema.

9.3 <GeneralSpace>

The GeneralSpace class is one of the two subclasses of NavigableSpace. GeneralSpace is idenfied as any navigable spaces except Transferspace such as rooms, terraces, lobbies, etc as shown as Figure 28.

```
<xs:element name="GeneralSpace" type="GeneralSpaceType" substitutionGroup="NavigableSpace"/>
<!-- ====================================================================== -->
<xs:complexType name="GeneralSpaceType">
    <xs:complexContent>
        <xs:extension base="NavigableSpaceType"/>
    </xs:complexContent>
</xs:complexType>
```

9.4 <TransferSpace>

The class TransferSpace is derived from NavigableSpace. It is used to model a space for providing passages between GeneralSpaces. It has three subclasses as ConnectionSpace, AnchorSpace, and TransitionSpace. Figure 28 shows ConnectionSpace and AnchorSpace in 2D or 3D space Model. Especially, a door (Door in CityGML or IfcDoor in IFC) is refered to ConnectionSpace or AnchorSpace in 3D Thick Door Model. A hallway and stairs also are represented as TransitionSpace. These subclasses of TransferSpace are mapped to State of IndoorGML core module.

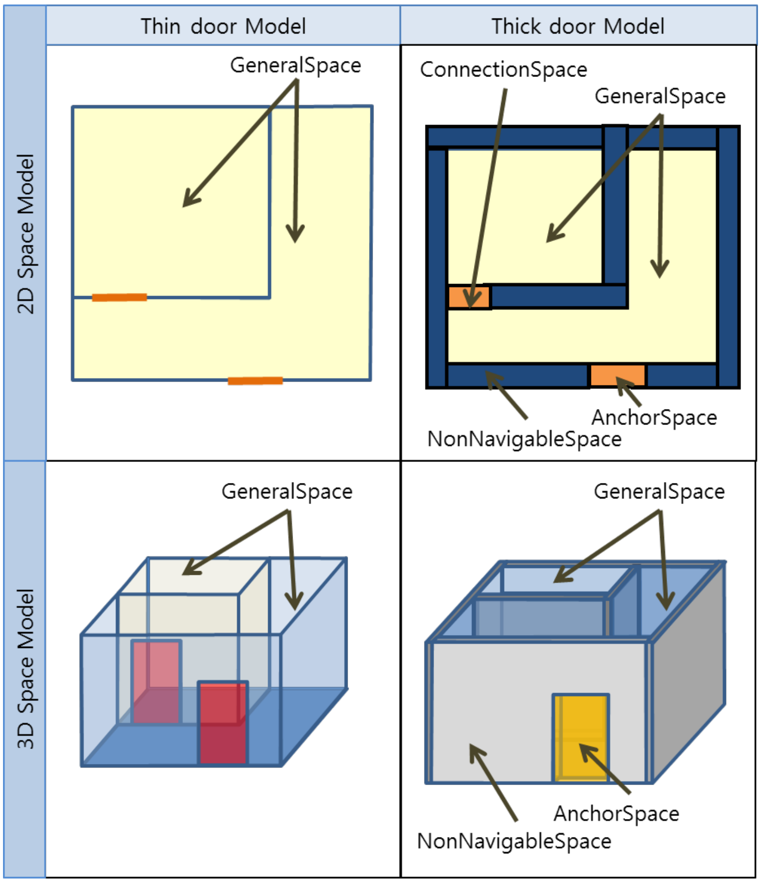


Figure 28: Examples of GeneralSpace, ConnectionSpace and AnchorSpace

```
<xs:element name="TransferSpace" type="TransferSpaceType" substitutionGroup="NavigableSpace"/>
    <!-- ====================================================================== -->
    <xs:complexType name="TransferSpaceType">
        <xs:complexContent>
            <xs:extension base="NavigableSpaceType"/>
        </xs:complexContent>
    </xs:complexType>
```

9.5 <ConnectionSpace>

ConnectionSpace represents an opening space that provides passages between two indoor spaces as shown as Figure 28. It refers to Door features in the Thick Door Model. As mentioned before, ConnectionSpace is mapped to State in the IndoorGML core module.

```
<xs:element name="ConnectionSpace" type="ConnectionSpaceType" substitutionGroup="TransferSpace"/>
<!-- ====================================================================== -->
<xs:complexType name="ConnectionSpaceType">
    <xs:complexContent>
        <xs:extension base="TransferSpaceType"/>
    </xs:complexContent>
</xs:complexType>
```

9.6 <AnchorSpace>

AnchorSpace represents a special opening space that provides connection between an indoor space and an outdoor space. It refers to *Entrance Doors*. It can be used as an AnchorNode, which is used as a control point for indoor-outdoor integrations.

```
<xs:element name="AnchorSpace" type="AnchorSpaceType" substitutionGroup="TransferSpace"/>
<!-- ====================================================================== -->
<xs:complexType name="AnchorSpaceType">
    <xs:complexContent>
        <xs:extension base="TransferSpaceType"/>
    </xs:complexContent>
</xs:complexType>
```

9.7 <TransitionSpace>

TransitionSpace represents a real world space that provides passage between two indoor spaces. It refers to corridors, stair and subspaces of hallway or corridor.

```
<xs:element name="TransitionSpace" type="TransitionSpaceType" substitutionGroup="TransferSpace"/>
<!-- ====================================================================== -->
<xs:complexType name="TransitionSpaceType">
    <xs:complexContent>
        <xs:extension base="TransferSpaceType"/>
    </xs:complexContent>
</xs:complexType>
```

9.8 <NavigableBoundary>

NavigableBoudary is defined as the boundary of NavigableSpace including the boundary of ConnectionSpace and AnchoreSpace.

As shown as Figure 29, a door is mapped to a NavigableBoundary in Thin Door

Model. Meanwhile in Thick Door Model, a door has two NavigableBoundary elements and two NonNavigableBoundary elements as shown as Figure 29. The boundary shared with other NavigableSpace is a NavigableBoundary and the others are mapped to NonNavigableBoundary class. The NavigableBoundary class has a subclass, called TrasferBoundary.

```
<xs:element name="NavigableBoundary" type="NavigableBoundaryType"
            substitutionGroup="IndoorCore:CellSpaceBoundary"/>
<!-- ====================================================================== -->
<xs:complexType name="NavigableBoundaryType">
    <xs:complexContent>
        <xs:extension base="IndoorCore:CellSpaceBoundaryType"/>
    </xs:complexContent>
</xs:complexType>
```

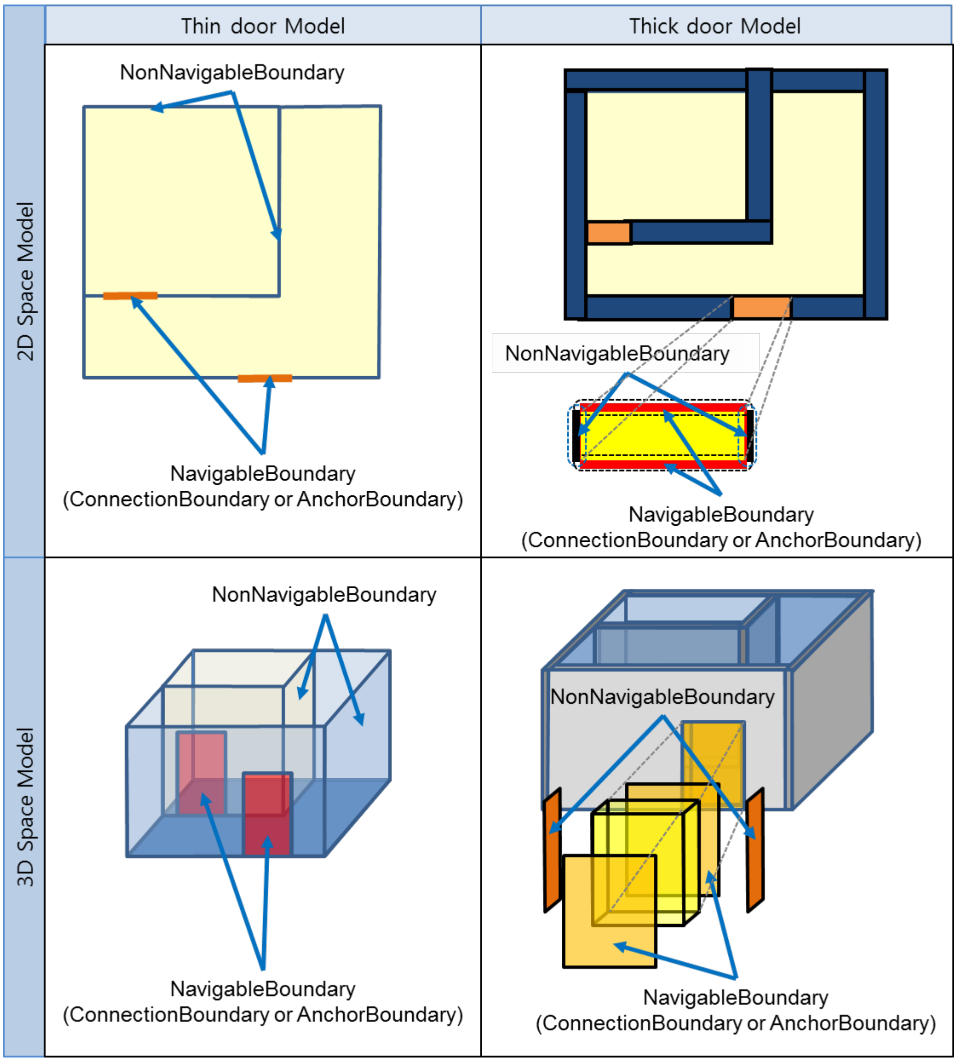


Figure 29: Example of NavigableBoundary

9.9 <TransferBoundary>

The class TransferBoundary is derived from NavigableBoundary. It is used to model a boundary for providing passages between NavigableSpace. NavigableBoundary has two subclasses, ConnectionBoundary and AnchorBoundary. As shown as Figure 29, in the Thin Door Model a door is mapped to ConnectionBoundary or to AnchorBoundary. In the Thick Door Model, some part of boundaries of door is mapped to ConnectionBoundary or AnchorBoundary. These subclasses of TransferSpace are mapped to Transition in the IndoorGML core module.

```
 <xs:element name="TransferBoundary" type="TransferBoundaryType"
substitutionGroup="NavigableBoundary"/>
<!-- ======================================================================= -->
<xs:complexType name="TransferBoundaryType">
    <xs:complexContent>
        <xs:extension base="NavigableBoundaryType">
            <xs:sequence/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

9.10 <ConnectionBoundary>

ConnectionBoundary represents a boundary which is connected with two adjacent NavigableSpaces. In a 2D space model, the ConnectionBoundary is represented as a gml:Curve. It is represented as a gml:Surface in 3D space model. As mentioned before, ConnectionBoundary is mapped to Transition in the IndoorGML core module.

```
<xs:element name="ConnectionBoundary" type="ConnectionBoundaryType"
                substitutionGroup="TransferBoundary"/>
<!-- ======================================================================= -->
<xs:complexType name="ConnectionBoundaryType">
    <xs:complexContent>
        <xs:extension base="TransferBoundaryType"/>
    </xs:complexContent>
</xs:complexType>
```

9.11 <AnchorBoundary>

AnchorBoundary represents a boundary which is the common boundary between a NavigableSpace and Outdoor space. It is represented as a part of boundary of AnchorSpace in Thick Door Model. In 2D space model, the AnchorBoundary is represented as a gml:Curve. It is represented as a gml:Surface in 3D space model. As mentioned before, AnchorBoundary is mapped to Transition in the IndoorGML core module.

```
<xs:element name="AnchorBoundary" type="AnchorBoundaryType" substitutionGroup="TransferBoundary"/>
<!-- ======================================================================= -->
<xs:complexType name="AnchorBoundaryType">
    <xs:complexContent>
        <xs:extension base="TransferBoundaryType"/>
    </xs:complexContent>
</xs:complexType>
```

9.12 <RouteNode>

RouteNode class represents a node associated with a routing path. It has an association relation with a State to get more information from the State. RouteNode also has the attribute of geometric location data for providing this information to application services on the client side.

```
<xs:element name="RouteNode" type="RouteNodeType"/>
<!-- ====================================================================== -->
<xs:complexType name="RouteNodeType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="referencedState" type="IndoorCore:StatePropertyType"/>
                <xs:element name="geometry" type="gml:PointPropertyType"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ====================================================================== -->
<xs:complexType name="RouteNodePropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="RouteNode"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

9.13 <RouteSegment>

RouteSegment represents connectivity relationships between spaces (e.g. a room within a building, door). RouteSegments are directed edges between RoutesNodes. Each edge will have at least two nodes. The RouteSegment contains two RouteNodes for representing start position and end position, which called as connects element. It also has association relation with Transition class in the Indoor Core Module, which named to referencedTrasition element.

```
<xs:element name="RouteSegment" type="RouteSegmentType"/>
<!-- ====================================================================== -->
<xs:complexType name="RouteSegmentType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="weight" type="xs:double"/>
                <xs:element name="connects" type="RouteNodePropertyType"  minOccurs="2" maxOccurs="2" />
                <xs:element name="referencedTransition" type="IndoorCore:TransitionPropertyType" />
                <xs:element name="geometry" type="gml:CurvePropertyType"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

9.14 <Route>

Route represents a possible path to navigate indoor space. It is ususally defined as a result of path find queries. The Route has a sequence of RouteNodes. The startRouteNode and endRouteNode represent a start position and end position of the path for indoor navigation. The attribute path contains RouteNode and RouteSegment of a possible path in indoor space and is represented as a sequence of RouteNodes and RouteSegments.

```
<xs:element name="Route" type="RouteType" substitutionGroup="gml:AbstractFeature"/>
<!-- ========================================================================== -->
<xs:complexType name="RouteType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="startRouteNode" type="RouteNodePropertyType" />
                <xs:element name="endRouteNode" type="RouteNodePropertyType" />
                <xs:element name="routeNodes" type="RouteNodesType" />
                <xs:element name="path" type="PathType" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ========================================================================== -->
<xs:complexType name="RouteNodesType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="nodeMember" type="RouteNodeMemberType" minOccurs="2"
                               maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ========================================================================== -->
<xs:complexType name="RouteNodeMemberType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureMemberType">
            <xs:sequence minOccurs="1">
                <xs:element ref="RouteNode"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ========================================================================== -->
<xs:complexType name="PathType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="routeMember" type="RouteSegmentMemberType" minOccurs="0"
                               maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ========================================================================== -->
<xs:complexType name="RouteSegmentMemberType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureMemberType">
            <xs:sequence>
                <xs:element ref="RouteSegment"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

9.15 Constraints

NavigableSpaceConstaints and NavigableBoundaryConstraint are linked to the topographic space semantic models through the relations. For providing indoor navigation, different categories of constraints can be identified: PassRestriction, WalkTypeRestriction, UserRestriction, TimeRestriction, and OneWayRestriction, as shown in Figure 30. Additional constraints can be defined as properties of the constraint elements.

The hierarchical conceptual constraint model could be used to create single or combined constraints for single or a series of semantic topographic space entities.

9.16 Requirements for conformance

This clause specifies the conformance requirements for the IndoorGML Indoor Navigation Module. Although most of conformance requiremens are already presented in the model and XML schema of the IndoorGML Indoor Navigation Module, and certain complementary conformance requirements are explicitly given in this clause;

> Requirement 5: Thick door model and thin door models *shall* not be defined in a same IndoorGML encoding.
>
> Requirement 6: Every thick door *shall* be encoded as an instance of either ConnectionSpace or AnchorSpace.
>
> Requirement 7: every thin door *shall* be encoded as an instance of either ConnectionBoundary or AnchorBoundary.
>
> Requirement 8: As shown in Figure 29, the boundary between a thick door and a thick wall *shall* be an instance of NonNavigableBoundary.

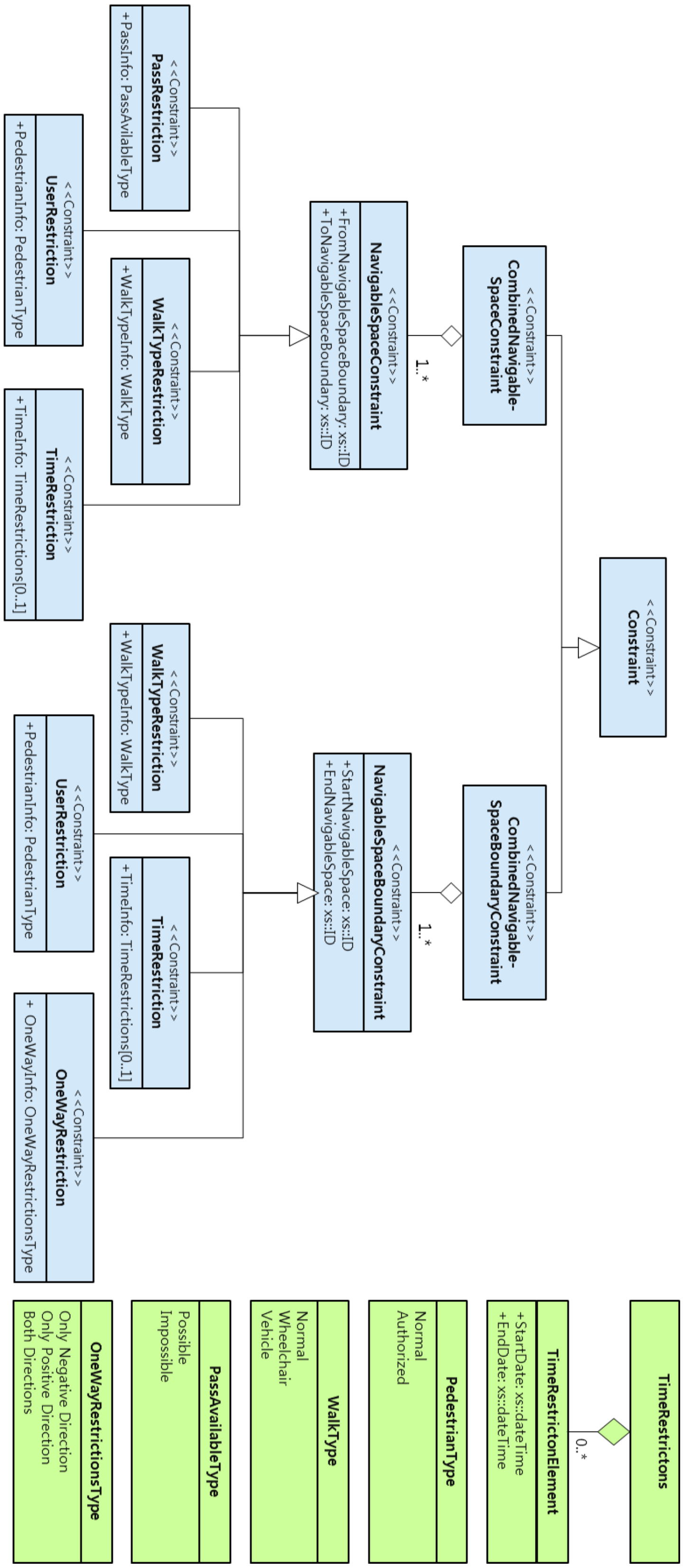


Figure 30: Conceptual model of indoor navigation constraints

Bibliography

[1] ISO 19107, *Geographic information – Spatial Schema*

[2] Lorenz, B., Ohlbach, H. J., Stoffel, E.-P., Rosner, M., *NL Navigation Commands from Indoor WLAN fingerprinting position data*, Technical Report of REWERSE-Project, Munich, Germany, September 2006

http://www.pms.ifi.lmu.de/publikationen/idefixStatic/rewerse-publications.html#REWERSE-DEL-2006-A1-D7, (accessed November 2010),

[3] Anagnostopoulus, C., Testsos, V., Kikiras, P., Hadjiefthymiades, S. P., *OntoNav: A Semantic Indoor Navigation System*, First International Workshop on Managing Context Information in Mobile and Pervasive Environments, Vol. 165, Ayia Napa, Zypern, 2003

[4] Kulyukin, V., Gharpure, C., Nicholson, J., *RoboCart: Toward Robot-Assisted Navigation of Crocery Stores by the Visually Impaired*, Proceedings of the international Conference on Intelligent Robots and Systems, IROS 2005, IEEE/RSJ

[5] Lefebvre, S., Hornus S. *Automatic cell-and-portal decomposition*. Technical Report 4898, INRIA, 2003. http://artis.imag.fr/Publications/2003/LH03/ (accessed November 2010).

[6] Lee, J., *3D GIS for Geo-coding Human Activity in Micro-scale Urban Environments,* M.J. Egenhofer, C. Freksa, and H.J. Miller (Eds.): GIScience 2004, Springer, Berlin, Germany

[7] Lee, J., Zlatanova, S., *A 3D data model and topological analyses for emergency response in urban areas*. Geospatial Information Technology for Emergency Response – Zlatanova & Li (eds), Taylor & Francis Group, London, UK, 2008

[8] Munkres, J. R., *Elements of Algebraic Topology*, Addison-Wesley, Menlo Park, CA, 1984

[9] Kolodziej, K. W., Hjelm, J., *Local Positioning Systems – LBS Applications and Services*, Taylor & Francis Group, London, UK, 2006

[10] OGC, CityGML, OGC 08-007r1 2008

[11] Kolbe, T.H., Gröger, G. & Plümer, L. *CityGML – Interoperable Access to 3D City Models,* P. van Oosterom, S. Zlatanova & E.M. Fendel (eds), Geo-information for Disaster Management; Proc. of the 1st International Symposium on Geo-information for Disaster Management', Delft, The Netherlands, March 21–23, 2005. Springer.

[12] Adachi, Y., Forester, J., Hyvarinen, J., Karstila, K., Liebich, T., Wix, J., *Industry Foundation Classes IFC2x Edition* 3, International Alliance for Interoperability, 2003, http://www.iai-international.org.

[13] Liao, L., Fox, D., Hightower, J. Kautz, H., Schulz, D., *Voronoi Tracking: Location Estimation Using Sparse and Noisy Sensor Data*, Proc. of the International Conference on Intelligent Robots and Systems, IROS 2003, IEEE/RSJ

[14] Egenhofer, M., Herring, J., *Categorizing Binary Topological Relations between Regions, Lines and Points in Geographical Databases*, NCGIA Technical Report, 1990

[15] OGC, *The Specification Model – A Standard for Modular specifications*, OGC 08-131r3, OGC Version: 1.0.0

[16] LaValle, S. M, *Planning Algorithms* Cambridge University Press, USA, 2006

[17] Lorenz, B., Ohlbach, H. J., Stoffel, E.-P., Rosner, M., *Towards a Semantic Spatial Model for Pedestrian Indoor Navigation,* Lecture Notes in Computer Science - Advances in Conceptual Modeling – Foundations and Applications, Volume 4802/2007, Springer, Berlin, Germany

[18] Morris, Sidney A., *Topology without Tears*, University of Ballarat, Australia, http://uob-community.ballarat.edu.au/~smorris/topology.htm, (accessed November 2010), 2007

[19] Fritsch, R. and Piccinini, R. A., *Cellular Structures in Topology*, Cambridge University Press, Cambridge, England, 2008

[20] Gold, C. M. and Ledoux, H., *Simultaneous Storage of Primal and Dual Three-Dimensional Subdivisions,* Computers, Environment and Urban Systems – Topology and Spatial Databases, Vol. 31, Issue 4, 2007, Elsevier

[21] Becker, T., Nagel, C., and Kolbe, T. H., *A Multilayered Space-Event Model for Navigation in Indoor Spaces*, Lecture Notes in Geoinformation and Cartography - 3D Geo-Information Science, 2008.

[22] Becker, T., Nagel, C., and Kolbe, T. H., 2009. *Supporting Contexts for Indoor Navigation using a Multilayered Space Model*, Tenth International Conference on Mobile Data Management:systems, services and middleware, 2009. MDM '09 ; 18 - 20 May 2009, Taipei, Taiwan. Online available http://ieeexplore.ieee.org/servlet/opac?punumber=5088898 (accessed November 2010)

[23] Booch, G., Rumbaugh, J., and Jacobson, I., *Unified Modeling Language User Guide*, Addison-Wesley, 1997

[24] Lee, Y.C., *Geographic information systems for urban applications: Problems and solutions*, Environment and Planning B: Planning and Design, Vol. 17:463-473, 1990.

[25] Lee, J. *2004. A Spatial Access Oriented Implementation of a Topological Data Model for 3D Urban Entities, GeoInformatica* 8(3), 235-262.

[26] Nagel, C., Becker B., Kaden, R., Li, K-J., Lee, J., and Kolbe, T. H., *Requirements and Space-Event Modeling for Indoor Navigation*, OGC 10-191r1, 2010

[27] Lee, J. *Indoor Spatial Data Model,* Proc. ISA Workshop, Seoul, Korea, Aug. 2008

[28] Wordnet, Princeton University, 2010, http://wordnet.princeton.edu

[29] OGC, OGC City Geography Markup Language (CityGML) Encoding Standard version 2.0, OGC 12-019, 2012

[30] OGC, OGC KML version 2.2, OGC 07-147r2, 2008

[31] buildingSmart, Industry Foundation Classes Release 2x3 version 2

Annex A
(Normative)
Abstract Test Suites for IndoorGML Instance Documents

A.1 Test Cases for mandatory conformance requirements

A.1.1 Valid IndoorGML instance document

a) Test purpose	Verify the validity of the IndoorGML instance document against the XML Schema definition of each IndoorGML module that is part of the IndoorGML profile employed by the instance document. This may be any combination of IndoorGML extension modules in conjunction with the IndoorGML core module.
b) Test method	Validate the IndoorGML XML instance document against the XML Schema definitions of all employed IndoorGML modules. The process may be using an appropriate software tool for validation or be a manual process that checks all relevant definitions from the respective XML Schema specification of the employed IndoorGML modules
c) Reference	Annex B
d) Test type	Basic Test

A.1.2 Conformance classes related to IndoorGML modules

a) Test purpose	Verify the validity of the IndoorGML instance document against the conformance classes of each IndoorGML module that is part of the IndoorGML profile employed by the instance document. This may be any combination of IndoorGML extension modules in conjunction with the IndoorGML core module. Note that only indoor navigation extension is defined in IndoorGML v.1.0 but other extension modules are expected to be included.
b) Test method	Follow the test cases provided by the conformance classes for each IndoorGML module in annex B.2.
c) Reference	Annex B.2
d) Test type	Basic Test

A.1.3 Spatial geometry objects

a) Test purpose	Verify that all spatial geometry objects within an IndoorGML instance document adhere to the XML Schema definition of the Geography Markup Language version 3.2.1 and to the IndoorGML spatial model
b) Test method	Inspect the instance document and check that spatial geometry objects are valid with respect to the XML Schema definition of GML version 3.2.1 and satisfy the rules of to the IndoorGML spatial model described in clause 8.
c) Reference	OGC Document No. 03-105r1, Annex B, chapter 8 and 9.
d) Test type	Capability Test

A.2 Conformance classes related to IndoorGML Modules

A.2.1 IndoorGML Core Module

A.2.1.1 Mandatory conformance requirements

a) Test purpose	Verify that the IndoorGML instance document follows the IndoorGML Core module's rules for encoding of objects and properties and adheres to all its conformance requirements. This test case is mandatory for all IndoorGML instance documents
b) Test method	Inspect the instance document and check that it satisfies the rules of the IndoorGML Core module described in Clause 8.
c) Reference	Clause 8
d) Test type	Capability Test

A.2.1.2 Valid IndoorGML instance document

a) Test purpose	Verify the validity of the IndoorGML instance document against the XML Schema definition of the IndoorGML Core module. This test case is mandatory for all IndoorGML instance documents.
b) Test method	Validate the IndoorGML XML instance document against the XML Schema definition of the IndoorGML Core module in annex B.1. The process may be using an appropriate software tool for validation or be a manual process that checks all relevant definitions from the IndoorGML Core module.
c) Reference	Annex B.1
d) Test type	Capability Test

A.2.2 IndoorGML Indoor Navigation Module

A.2.2.1 Mandatory conformance requirements

a) Test purpose	Verify that the IndoorGML instance document follows the IndoorGML Indoor Navigation module's rules for encoding of objects and properties and adheres to all its conformance requirements. This test case is mandatory for all IndoorGML instance documents which employ elements defined within the *IndoorNavigation* module
b) Test method	Inspect the instance document and check that it satisfies the rules of the IndoorGML Indoor Navigation module described in clause 9.
c) Reference	Clause 9
d) Test type	Capability Test

A.2.1.2 Valid IndoorGML instance document

a) Test purpose	Verify the validity of the IndoorGML instance document against the XML Schema definition of the IndoorGML Indoor Navigation module. This test case is mandatory for all IndoorGML instance documents.
b) Test method	Validate the IndoorGML XML instance document against the XML Schema definition of the IndoorGML Indoor Navigation module in annex B.2. The process may be using an appropriate software tool for validation or be a manual process that checks all relevant definitions from the IndoorGML Indoor Navigation module.
c) Reference	Annex B.2
d) Test type	Capability Test

Annex B
(informative)
XML Schema for IndoorGML

In addition to this document, this standard includes some normative XML Schema Documents. These XML Schema Documents are included in this document. These XML Schema Documents are also posted online at the URL http://schemas.opengis.net/indoorgml/1.0. In the event of a discrepancy between this document and online versions of the XML Schema Document files (posted at schemas.opengis.net), the online schema files SHALL be considered authoritative. **(normative)**

B.1 IndoorGML Core Module

```
<?xml version="1.0" encoding="UTF-8"?>
  <xs:schema xmlns="http://www.opengis.net/indoorgml/1.0/core"
            xmlns:xs=http://www.w3.org/2001/XMLSchema xmlns:gml="http://www.opengis.net/gml/3.2"
            xmlns:xlink="http://www.w3.org/1999/xlink"
            targetNamespace="http://www.opengis.net/indoorgml/1.0/core" elementFormDefault="qualified">
  <!-- ====================================================================== -->
  <xs:element name="IndoorFeatures" type="IndoorFeaturesType" substitutionGroup="gml:AbstractFeature"/>
  <!-- ====================================================================== -->
  <xs:complexType name="IndoorFeaturesType">
    <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureType">
        <xs:sequence>
          <xs:element name="primalSpaceFeatures"" type="PrimalSpaceFeaturesPropertyType"
                  minOccurs="0" maxOccurs="1"/>
          <xs:element name="multiLayeredGraph" type="MultiLayeredGraphPropertyType"
                  minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <!-- ====================================================================== -->
  <xs:element name="PrimakSpaceFeatures" type="PrimalSpaceFeaturesType"
              substitutionGroup="gml:AbstractFeature"/>
  <!-- ====================================================================== -->
  <xs:complexType name="PrimalSpaceFeaturesPropertyType">
    <xs:sequence minOccurs="0">
      <xs:element ref="PrimalSpaceFeatures"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
  </xs:complexType>
  <!-- ====================================================================== -->
  <xs:complexType name="PrimalSpaceFeaturesType">
    <xs:complexContent>
      <xs:extension base="gml:AbstractFeatureType">
        <xs:sequence>
          <xs:element name="cellSpaceMember" type="CellSpaceMemberType" minOccurs="0"
                 maxOccurs="unbounded"/>
          <xs:element name="cellSpaceBoundaryMember" type="CellSpaceBoundaryMemberType"
                minOccurs="0"  maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
      </xs:extension>
```

```
        </xs:complexContent>
    </xs:complexType>
    <!-- ======================================================================= -->
    <xs:complexType name="CellSpaceMemberType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureMemberType">
                <xs:sequence minOccurs="0">
                    <xs:element ref="CellSpace"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ======================================================================= -->
    <xs:complexType name="CellSpaceBoundaryMemberType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureMemberType">
                <xs:sequence minOccurs="0">
                    <xs:element ref="CellSpaceBoundary"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ======================================================================= -->
    <xs:element name="MultiLayeredGraph" type="MultiLayeredGraphType"
                   substitutionGroup="gml:AbstractFeature">
        <xs:annotation>
            <xs:documentation>
                   The overall structure of the Multilayered Space Model constitutes a multilayered graph, where all
                   the nodes from all n layers are included but are separated into n partitions which are connected by
                   the inter-space connections. Furthermore the graph also contains the state transition edges (intra-
                   space connections)
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <!-- ======================================================================= -->
    <xs:complexType name="MultiLayeredGraphType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="spaceLayers" type="SpaceLayersType" minOccurs="1"
                                   maxOccurs="unbounded"/>
                    <xs:element name="interEdges" type="InterEdgesType" minOccurs="0"
                                   maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ======================================================================= -->
    <xs:complexType name="SpaceLayersType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="spaceLayerMember" type="SpaceLayerMemberType" minOccurs="1"
                                   maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ======================================================================= -->
    <xs:complexType name="SpaceLayerMemberType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureMemberType">
                <xs:sequence minOccurs="1">
                    <xs:element ref="SpaceLayer"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ================================================================================ -->
    <xs:complexType name="InterEdgesType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="interLayerConnectionMember" type="InterLayerConnectionMemberType"
                                   minOccurs="1" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ================================================================================ -->
    <xs:complexType name="InterLayerConnectionMemberType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureMemberType">
                <xs:sequence minOccurs="0">
                    <xs:element ref="InterLayerConnection"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ================================================================================ -->
    <xs:element name="InterLayerConnection" type="InterLayerConnectionType"
                   substitutionGroup="gml:AbstractFeature">
        <xs:annotation>
            <xs:documentation>Denotin the interspace connections between the SpaceLayer
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <!-- ================================================================================ -->
    <xs:complexType name="InterLayerConnectionType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="typeOfTopoExpression" type="typeOfTopoExpressionCodeType"
                                   minOccurs="0" maxOccurs="1"/>
                    <xs:element name="comment" type="xs:string"  minOccurs="0" maxOccurs="1"/>
                    <xs:element name="interConnects" type="StatePropertyType" minOccurs="2" maxOccurs="2"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ================================================================================ -->
    <xs:complexType name="InterLayerConnectionPropertyType">
        <xs:sequence minOccurs="0">
            <xs:element ref="InterLayerConnection"/>
        </xs:sequence>
        <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xs:complexType>
    <!-- ================================================================================ -->
    <xs:element name="SpaceLayer" type="SpaceLayerType" substitutionGroup="gml:AbstractFeature">
        <xs:annotation>
                  <xs:documentation>
                         &lt;SpaceLayer&gt;s represent various space concepts such as topography, sensor, security, etc.
                         A SpaceLayer aggregates  &lt;State&gt; and &lt;Transition&gt; which are directly associated
                         with the corresponding geometry classes.
                  </xs:documentation>
        </xs:annotation>
    </xs:element>
    <!-- ====================================================================== -->
    <xs:complexType name="SpaceLayerType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="usage" type="gml:CodeType" minOccurs="0" maxOccurs="unbounded"/>
                    <xs:element name="terminationDate" type="xs:dateTime"  minOccurs="0" maxOccurs="1" />
                    <xs:element name="function" type="gml:CodeType" minOccurs="0" maxOccurs="unbounded"/>
                    <xs:element name="creationDate" type="xs:dateTime"  minOccurs="0" maxOccurs="1" />
                    <xs:element name="class" type="SpaceLayerClassTypeType" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="nodes" type="NodesType" minOccurs="1" maxOccurs="unbounded"/>
                    <xs:element name="edges" type="EdgesType" minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="NodesType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="stateMember" type="StateMemberType" minOccurs="1"
                                  maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
                <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="StateMemberType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureMemberType">
                <xs:sequence minOccurs="0">
                    <xs:element ref="State"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="EdgesType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="transitionMember" type="TransitionMemberType" minOccurs="0"
                                  maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
                <xs:attributeGroup ref="gml:OwnershipAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="TransitionMemberType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureMemberType">
                <xs:sequence minOccurs="0">
                    <xs:element ref="Transition"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:element name="State" type="StateType" substitutionGroup="gml:AbstractFeature">
        <xs:annotation>
            <xs:documentation>
                Within the dual graph structure of one layer a node in dual space represents a space (e.g. a room
                within a building) in primal space
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <!-- ====================================================================== -->
    <xs:complexType name="StateType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="duality" type="CellSpacePropertyType"  minOccurs="0" maxOccurs="1"/>
                    <xs:element name="connects" type="TransitionPropertyType" minOccurs="0"
                                maxOccurs="unbounded"/>
                    <xs:element name="geometry" type="gml:PointPropertyType"  minOccurs="0"
                                maxOccurs="1"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="StatePropertyType">
        <xs:sequence minOccurs="0">
            <xs:element ref="State"/>
        </xs:sequence>
        <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:element name="Transition" type="TransitionType" substitutionGroup="gml:AbstractFeature">
        <xs:annotation>
            <xs:documentation>
                Within the dual graph structure of one layer, an edge in dual space represents the adjacencies or
                connections (e.g. doors or passages as intra-space connections)
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <!-- ====================================================================== -->
    <xs:complexType name="TransitionType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="weight" type="xs:double"  minOccurs="0" maxOccurs="1"/>
                    <xs:element name="connects" type="StatePropertyType"  minOccurs="2" maxOccurs="2"/>
                    <xs:element name="duality" type="CellSpaceBoundaryPropertyType" minOccurs="0"
                                maxOccurs="1"/>
                    <xs:element name="geometry" type="gml:CurvePropertyType" minOccurs="0"
                                maxOccurs="1"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="TransitionPropertyType">
        <xs:sequence minOccurs="0">
            <xs:element ref="Transition"/>
        </xs:sequence>
        <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:element name="CellSpace" type="CellSpaceType" substitutionGroup="gml:AbstractFeature">
        <xs:annotation>
            <xs:documentation>
```

```
                Within the dual graph structure of one layer a node in dual space represents a space (e.g. a room
                within a building) in primal space
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <!-- ====================================================================== -->
    <xs:complexType name="CellSpaceType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="cellSpaceGeometry" type="CellSpaceGeometryType"
                                minOccurs="0" maxOccurs="1"/>
                    <xs:element name="duality" type="StatePropertyType"  minOccurs="0" maxOccurs="1"/>
                    <xs:element name="externalReference" type="ExternalReferenceType" minOccurs="0"
                                maxOccurs="unbounded"/>
                    <xs:element name="partialboundedBy" type="CellSpaceBoundaryPropertyType" minOccurs="0"
                                maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="CellSpacePropertyType">
        <xs:sequence minOccurs="0">
            <xs:element ref="CellSpace"/>
        </xs:sequence>
        <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="CellSpaceGeometryType">
        <xs:choice>
            <xs:element name="Geometry3D" type="gml:SolidPropertyType"/>
            <xs:element name="Geometry2D" type="gml:SurfacePropertyType"/>
        </xs:choice>
    </xs: complexType >
    <!-- ====================================================================== -->
    <xs:element name="CellSpaceBoundary" type="CellSpaceBoundaryType"
                substitutionGroup="gml:AbstractFeature">
        <xs:annotation>
            <xs:documentation>
                 Within the dual graph structure of one layer a edge in dual space represents a spaceboundary (e.g. a
                wall between adjacenced rooms in a building) in primal space
        </xs:documentation>
        </xs:annotation>
    </xs:element>
    <!-- ====================================================================== -->
    <xs:complexType name="CellSpaceBoundaryType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="duality" type="TransitionPropertyType" minOccurs="0"
                                maxOccurs="1"/>
                    <xs:element name="cellSpaceBoundaryGeometry"
                                 type="CellSpaceBoundaryGeometryType" minOccurs="0
                               maxOccurs="1"/>
                    <xs:element name="externalReference" type="ExternalReferenceType" minOccurs="0"
                                maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="CellSpaceBoundaryGeometryType">
        <xs:choice>
            <xs:element name="geometry3D" type="gml:SurfacePropertyType"/>
            <xs:element name="geometry2D" type="gml:CurvePropertyType"/>
        </xs:choice>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="CellSpaceBoundaryPropertyType">
        <xs:sequence>
            <xs:element ref="CellSpaceBoundary"/>
        </xs:sequence>
        <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="ExternalReferenceType">
        <xs:sequence>
            <xs:element name="informationSystem" type="xs:anyURI" minOccurs="0"/>
            <xs:element name="externalObject" type="externalObjectReferenceType"/>
        </xs:sequence>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="externalObjectReferenceType">
        <xs:choice>
            <xs:element name="name" type="xs:string" minOccurs="0"/>
            <xs:element name="uri" type="xs:anyURI"/>
        </xs:choice>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:simpleType name="SpaceLayerClassTypeType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="TOPOGRAPHIC"/>
            <xs:enumeration value="SENSOR"/>
            <xs:enumeration value="LOGICAL"/>
            <xs:enumeration value="TAGS"/>
            <xs:enumeration value="UNKNOWN"/>
        </xs:restriction>
    </xs:simpleType>
    <!-- ====================================================================== -->
    <xs:simpleType name="typeOfTopoExpressionCodeType">
        <xs:union memberTypes="typeOfTopoExpressionCodeEnumerationType
                               typeOfTopoExpressionCodeOtherType"/>
    </xs:simpleType>
    <!-- ====================================================================== -->
    <xs:simpleType name="typeOfTopoExpressionCodeEnumerationType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="CONTAINS"/>
            <xs:enumeration value="OVERLAPS"/>
            <xs:enumeration value="EQUALS"/>
            <xs:enumeration value="WITHIN"/>
            <xs:enumeration value="CROSSES"/>
            <xs:enumeration value="INTERSECTS"/>
        </xs:restriction>
    </xs:simpleType>
    <!-- ====================================================================== -->
    <xs:simpleType name="typeOfTopoExpressionCodeOtherType">
        <xs:restriction base="xs:string">
            <xs:pattern value="other. \w{2,}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

B.2 IndoorGML Indoor Navigation Module

```
<?xml version="1.0" encoding="UTF-8"?>
   <xs:schema xmlns="http://www.opengis.net/indoorgml/1.0/navigation"
           xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml/3.2"
           xmlns:xlink="http://www.w3.org/1999/xlink"
           xmlns:IndoorCore="http://www.opengis.net/indoorgml /1.0/core"
           targetNamespace="http://www.opengis.net/indoorgml /1.0/navigation"
           elementFormDefault="qualified">
   <xs:import namespace="http://www.opengis.net/gml/3.2"
             schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
   <xs:import namespace="http://www.opengis.net/indoorgml /1.0/core "
             schemaLocation="http://schemas.opengis.net/indoorgml/1.0/indoorgmlcore.xsd"/>
   <!-- ============================================================================ -->
   <xs:element name="NavigableSpace" type="NavigableSpaceType"
               substitutionGroup="IndoorCore:CellSpace">
      <xs:annotation>
         <xs:documentation>NavigableSpace
         </xs:documentation>
      </xs:annotation>
   </xs:element>
   <!-- ============================================================================ -->
   <xs:complexType name="NavigableSpaceType">
      <xs:complexContent>
         <xs:extension base="IndoorCore:CellSpaceType">
            <xs:sequence>
               <xs:element name="class" type="gml:CodeType"/>
               <xs:element name="function" type="gml:CodeType"/>
               <xs:element name="usage" type="gml:CodeType"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <!-- ============================================================================ -->
   <xs:element name="GeneralSpace" type="GeneralSpaceType" substitutionGroup="NavigableSpace">
      <xs:annotation>
         <xs:documentation>Denote general indoor space such as rooms, terraces, lobbies, etc.
         </xs:documentation>
      </xs:annotation>
   </xs:element>
   <!-- ============================================================================ -->
   <xs:complexType name="GeneralSpaceType">
      <xs:complexContent>
         <xs:extension base="NavigableSpaceType"/>
      </xs:complexContent>
   </xs:complexType>
   <!-- ============================================================================ -->
   <xs:element name="TransferSpace" type="TransferSpaceType" substitutionGroup="NavigableSpace">
      <xs:annotation>
         <xs:documentation>Denote the space which is purposed to tranfer between spaces.
         </xs:documentation>
      </xs:annotation>
   </xs:element>
   <!-- ============================================================================ -->
   <xs:complexType name="TransferSpaceType">
      <xs:complexContent>
         <xs:extension base="NavigableSpaceType"/>
      </xs:complexContent>
   </xs:complexType>
   <!-- ============================================================================ -->
   <xs:element name="AnchorSpace" type="AnchorSpaceType" substitutionGroup="TransferSpace">
      <xs:annotation>
         <xs:documentation>
            AnchorSpace represents a special opening space that provides connection between an indoor space
            and an outdoor space. It refers to Entrance Doors.
        </xs:documentation>
    </xs:annotation>
</xs:element>
<!-- ============================================================================ -->
<xs:complexType name="AnchorSpaceType">
    <xs:complexContent>
        <xs:extension base="TransferSpaceType"/>
    </xs:complexContent>
</xs:complexType>
<!-- ============================================================================ -->
<xs:element name="ConnectionSpace" type="ConnectionSpaceType" substitutionGroup="TransferSpace">
    <xs:annotation>
        <xs:documentation>
            ConnetionSpace represents an opening space that provides passages between two indoor spaces
        </xs:documentation>
    </xs:annotation>
</xs:element>
<!-- ============================================================================ -->
<xs:complexType name="ConnectionSpaceType">
    <xs:complexContent>
        <xs:extension base="TransferSpaceType"/>
    </xs:complexContent>
</xs:complexType>
<!-- ============================================================================ -->
<xs:element name="TransitionSpace" type="TransitionSpaceType" substitutionGroup="TransferSpace">
    <xs:annotation>
        <xs:documentation>
            TransitionSpace represents a real world space that provides passage between two indoor spaces
        </xs:documentation>
    </xs:annotation>
</xs:element>
<!-- ============================================================================ -->
<xs:complexType name="TransitionSpaceType">
    <xs:complexContent>
        <xs:extension base="TransferSpaceType"/>
    </xs:complexContent>
</xs:complexType>
<xs:element name="Route" type="RouteType" substitutionGroup="gml:AbstractFeature">
    <xs:annotation>
        <xs:documentation>Route
        </xs:documentation>
    </xs:annotation>
</xs:element>
<!-- ============================================================================ -->
<xs:element name="NavigableBoundary" type="NavigableBoundaryType"
                substitutionGroup="IndoorCore:CellSpaceBoundary">
    <xs:annotation>
        <xs:documentation> NavigableBoundary </xs:documentation>
    </xs:annotation>
</xs:element>
<!-- ============================================================================ -->
<xs:complexType name="NavigableBoundaryType">
    <xs:complexContent>
        <xs:extension base="IndoorCore:CellSpaceBoundaryType"/>
    </xs:complexContent>
</xs:complexType>
<!-- ============================================================================ -->
<xs:element name="TransferBoundary" type="TransferBoundaryType"
                substitutionGroup="NavigableBoundary">
    <xs:annotation>
        <xs:documentation> TransferBoundary </xs:documentation>
    </xs:annotation>
</xs:element>
<!-- ============================================================================ -->
<xs:complexType name="TransferBoundaryType">
```

```
        <xs:complexContent>
            <xs:extension base="NavigableBoundaryType">
                <xs:sequence/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:element name="ConnectionBoundary" type="ConnectionBoundaryType"
                substitutionGroup="TransferBoundary">
        <xs:annotation>
            <xs:documentation> ConnectionBoundary </xs:documentation>
        </xs:annotation>
    </xs:element>
    <!-- ====================================================================== -->
    <xs:complexType name="ConnectionBoundaryType">
        <xs:complexContent>
            <xs:extension base="TransferBoundaryType"/>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:element name="AnchorBoundary" type="AnchorBoundaryType" substitutionGroup="TransferBoundary">
        <xs:annotation>
            <xs:documentation> AnchorBoundary </xs:documentation>
        </xs:annotation>
    </xs:element>
    <!-- ====================================================================== -->
    <xs:complexType name="AnchorBoundaryType">
        <xs:complexContent>
            <xs:extension base="TransferBoundaryType"/>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="RouteType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="startRouteNode" type="RouteNodePropertyType" />
                    <xs:element name="endRouteNode" type="RouteNodePropertyType" />
                    <xs:element name="routeNodes" type="RouteNodesType" />
                    <xs:element name="path" type="PathType" />
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="RouteNodesType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureType">
                <xs:sequence>
                    <xs:element name="nodeMember" type="RouteNodeMemberType" minOccurs="2"
                                maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- ====================================================================== -->
    <xs:complexType name="RouteNodeMemberType">
        <xs:complexContent>
            <xs:extension base="gml:AbstractFeatureMemberType">
                <xs:sequence minOccurs="1">
                    <xs:element ref="RouteNode"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
```

```
<!-- ================================================================================ -->
<xs:complexType name="PathType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="routeMember" type="RouteSegmentMemberType" minOccurs="0"
                                maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attributeGroup ref="gml:AggregationAttributeGroup"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ================================================================================ -->
<xs:complexType name="RouteSegmentMemberType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureMemberType">
            <xs:sequence>
                <xs:element ref="RouteSegment"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ================================================================================ -->
<xs:element name="RouteSegment" type="RouteSegmentType">
    <xs:annotation>
        <xs:documentation>RouteSegment
        </xs:documentation>
    </xs:annotation>
</xs:element>
<!-- ================================================================================ -->
<xs:complexType name="RouteSegmentType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="weight" type="xs:double"/>
                <xs:element name="connects" type="RouteNodePropertyType" minOccurs="2"
                                maxOccurs="2"/>
                <xs:element name="referencedTransition" type="IndoorCore:TransitionPropertyType" />
                <xs:element name="geometry" type="gml:CurvePropertyType"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ================================================================================ -->
<xs:element name="RouteNode" type="RouteNodeType">
    <xs:annotation>
        <xs:documentation>Route Node
        </xs:documentation>
    </xs:annotation>
</xs:element>
<!-- ================================================================================ -->
<xs:complexType name="RouteNodeType">
    <xs:complexContent>
        <xs:extension base="gml:AbstractFeatureType">
            <xs:sequence>
                <xs:element name="referencedState" type="IndoorCore:StatePropertyType"/>
                <xs:element name="geometry" type="gml:PointPropertyType"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- ================================================================================ -->
<xs:complexType name="RouteNodePropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="RouteNode"/>
    </xs:sequence>
```

```
        <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xs:complexType>
    <!-- ==================================================================================== -->
</xs:schema>
```

Annex C
(Informative)
IndoorGML Sample Data

C.1 Introduction

A sample data is given at http://indoorgml.net, which comprises two datasets of IndoorGML core module and CityGML LoD 4. This sample data shows the basic structure of IndoorGML data and how IndoorGML and CityGML datasets are linked via external references.

C.2 Datasets of CityGML LoD 4 and IndoorGML

The CityGML data, called FJK-Haus CityGML LoD 4 (http://www.iai.fzk.de/www-extern/index.php?id=2196&L=1) is derived from an IFC data for a building of three floors as shown in Figure C-1. And the corresponding IndoorGML is also made based on this CityGML data in a semi-automatical way that most geometries of cells are extracted from CityGML data except State and Transition instances.

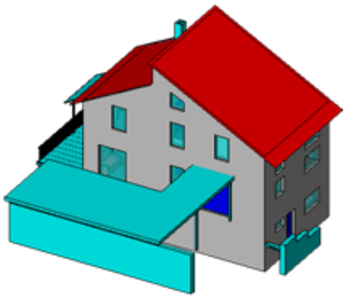


Figure 31[C-1]: FJK-Haus CityGML LoD 4 data (http://www.iai.fzk.de/www-extern/index.php?id=2196&L=1)

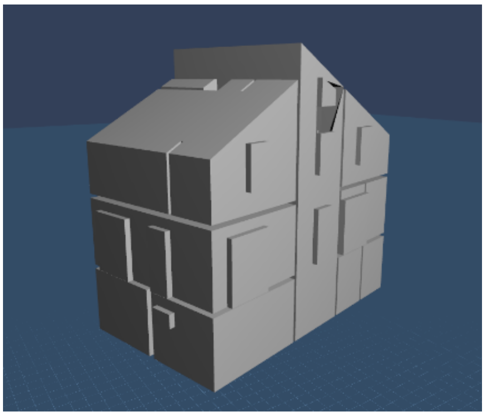


Figure 32 [C-2]: FJK-Haus IndoorGML data – Geometry of CellSpace

Figure C-2 shows the cell instances of CellSpace, where the 3D geometries of CellSpace are included in IndoorGML data. The network data of IndoorGML core module is illustrated by Figure C-3.

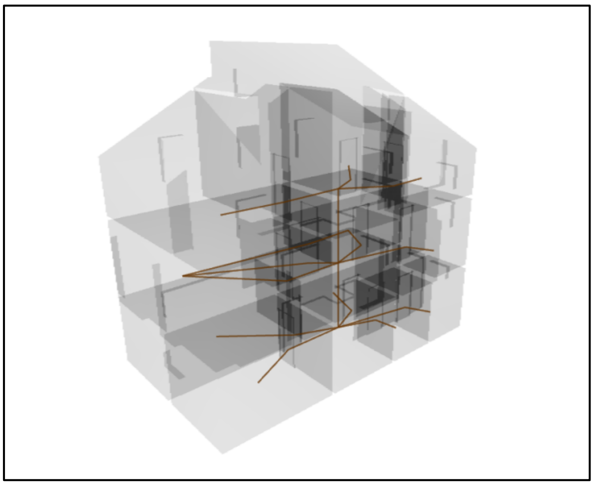


Figure 33 [C-3:] FJK-Haus IndoorGML data – Network

C.3 External References of IndoorGML to CityGML objects

The IndoorGML dataset also contains external references to objects in CityGML data. A Room object 102 of CityGML dataset in Figure C-4 is referenced by a CellSpace object of IndoorGML in Figure C-5.

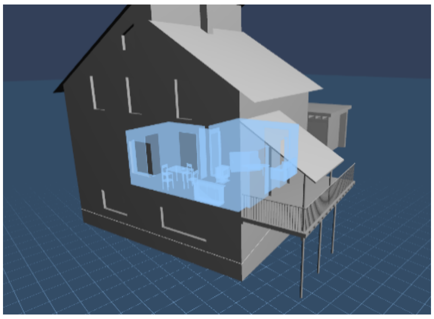


Figure 34 [C-4]: Room object 102
object 102 in
in CityGML dataset

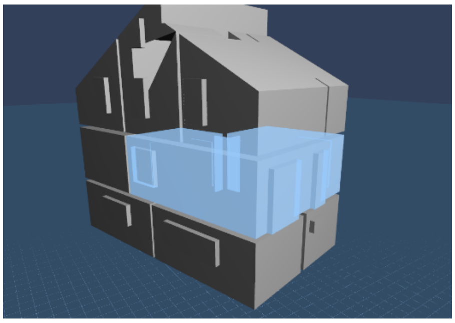


Figure 35 [C-5]: CellSpace

IndoorGML dataset

The CityGML document and IndoorGML document containing corresponding objects are as follows;

```
<bldg:interiorRoom>
    <bldg:Room  gml:id="GMLID_BUI198667_373_6703">
    <gml:description>Wohnen / Essen</gml: description >
        <gml:name>102</gml:name>
        <bldg:lod4Solid>
        <bldg:boundedBy />
        . . .

        <bldg:boundedBy />
    </bldg:Room>
</bldg:interiorRoom>
```

Figure 36 [C-6]: CityGML file (FJK-Haus-LoD4-V3.gml) containing Room object 102

```
<indoorcore:State gml:id="R15">
        <gml:name>102</gml:name>
        <duality gml:id="SR15">
            <gml:name>102</gml:name>
            <Geometry3D>
                <gml:Solid gml:id="sl15">
                </gml:Solid>
            </Geometry3D>
            <externalReference>
                <informationSystem>FJK-Haus-LoD4-V3.gml</informationSystem>
                <externalObject>
                    <name>GMLID_BUI198667_373_6703</name>
                </externalObject>
            </externalReference>
        </duality>
        <connects xlink:href="#T12"/>
        . . .
        <geometry>
            <gml:Point gml:id="P15">
                <gml:pos>445539.871000732 5444907.64979755 0.66</gml:pos>
            </gml:Point>
        </geometry>
</indoorcore:State>
```

Figure 37 [C-7]: IndoorGML document containing Room object 102

Annex D
(Informative)
Example of CodeList

For mapping indoor space features to IndoorNavigation module classes, space features have to be classified by functions and usages of space. Each space class has attributes for functions, usages, and classes of space. Because misspellings or different names for the same notion brings the problems in data interoperability, in order to overcome the typo errors, an example of CodeList is provided in this annex to specify the attribute values including space class types, space function types and space usage types. The CodeLists are defined from OmniClass (Table 13 and Table 14) created and used by the North American architectural, engineering and construction (AEC) industry.

D.1 GeneralSpace

GeneralSpaceClassType			
Code list derived from OmniClass Table 13 and CityGML			
1000	Administration	1060	Laboratory
1010	Business, trade	1070	Service
1020	Education, training	1080	Production
1030	Recreation	1090	Storage
1040	Art, performance	1100	Security
1050	Healthcare	1110	Accommodation, Waste management

GeneralSpaceFunctionType			
Code list derived from OmniClass Table 13			
1000	Elevator Machine Room	2550	Lecture Classroom
1010	Fire Command Center	2560	Lecture Hall
1020	Men’s Restroom	2570	Seminar Room
1030	Unisex Restroom	2580	Astronomy Teaching Laboratory
1040	Refrigerant Machinery Room	2590	Research/non-class Class Laboratory
1050	Incinerator Room	2600	Training Space
1060	Gas Room	2610	Woodshop/Metalshop
1070	Liquid Use, Dispensing and Mixing Room	2620	Religious Education Space
1080	Electrical Room	2630	Study Service
1090	Telecommunications Room	2640	Basketball Courts
1100	Hazardous Waste Storage	2650	Team Athletic Recreation Spaces
1110	Building Manager Office	2660	Volleyball Court
1120	Guard Stations	2670	Boxing Ring
1130	Women’s Restroom	2680	Circuit Training Course Area
1140	Furnace Room	2690	Aerobic Studio
1150	Fuel Room	2700	Swimming Pool
1160	Liquid Storage Room	2710	Firing Range
1170	Hydrogen Cutoff Room	2720	Hobby and Craft Center

1180	Switch Room	2730	Exercise Room
1190	Classrooms	2740	Skating Rink
1200	Assembly Hall	2750	Climbing Wall
1210	Physics Teaching Laboratory	2760	Diving Tank
1220	Open Class Laboratory	2770	Game Room
1230	Laboratory Service Space	2780	Fitness Center
1240	Computer Lab	2790	Weight Room
1250	Training Support Space	2800	Courtroom
1260	Study Room	2810	Jury Room
1270	Evidence Room	2820	Jury Assembly Space
1280	Witness Stand	2830	Judge’s Chambers
1290	Robing Area	2840	Hearing Room
1300	Council Chambers	2850	Legislative Hearing Room
1310	Armory	2860	Acting Stage
1320	General Performance Spaces	2870	Performance Rehearsal Space
1330	Orchestra Pit	2880	Banding Training Space
1340	Performance Hall	2890	Pre-Function Lobby
1350	Audience Space	2900	Supporting Performance Space
1360	Audience Seating Space	2910	Catwalk
1370	Projection Booth	2920	Motion Picture Screen Space
1380	Stage Wings	2930	Exhibit Gallery
1390	Art Gallery	2940	Display Space
1400	Sculpture Garden	2950	Artist’s Studio
1410	Recording Studio	2960	Media Production
1420	Photo Lab	2970	Museum Gallery
1430	Library	2980	Baptistery
1440	Mediation Chapel	2990	Cathedra
1450	Reflection Space	3000	Clean Room
1460	Chapel	3010	Data Center
1470	Shrine	3020	Computer Server Room
1480	Confessional Space	3030	Exam Room
1490	Tabernacle	3040	General Examination Space
1500	Choir Loft	3050	Labor, Delivery, Recovery, Postpartum Room
1510	Marriage Sanctuary	3060	Newborn Nursery
1520	Mental Health Quiet Room	3070	Patient Room
1530	Bone Densitometry Room	3080	Clean Supply Room
1540	CT Simulator Room	3090	Consultation Room
1550	Head Radiographic Room	3100	Equipment Storage Room
1560	Mobile Imaging System Alcove	3110	Nurse Workspace
1570	MRI System Component Room	3120	Nurse Triage Space
1580	PET/CT Simulator Room	3130	Mental Health Multipurpose room w/Control Room
1590	Radiographic Room	3140	Holding Room, Secured
1600	Stereotactic Mammography Room	3150	Anteroom
1610	Ultrasound/Optical Coherence Tomography Room	3160	Medical Information Computer System Room
1620	Angiographic Control Room	3170	Nursery Transport Unit Alcove
1630	Angiographic Procedure Control Area	3180	Clean Linen Storage Room
1640	Silver Collection Area	3190	Clean Utility Room
1650	Computer Image Processing Area	3200	Mental Health

			Interview/Counseling Room
1660	CT Control Area	3210	Medical Records Storage room
1670	Image Quality Control Room	3220	Nurse Station
1680	X-Ray, Plane Film Storage Space	3230	Soiled Utility Room
1690	MRI Control Room	3240	Resuscitation Cart Alcove
1700	MRI Viewing Room	3250	Angiographic Procedure Room
1710	Radiographic Control Room	3260	CT Scanning Room
1720	Tele-Radiology/Tele-Medicine Room	3270	Cystoscopic Radiology Room
1730	Radiation Diagnostic and Therapy Spaces	3280	Mammography Room
1740	Health Physics Laboratory	3290	MRI Scanning Room
1750	Linear Accelerator Entrance Maze, Healthcare	3300	PET/CT Scanning Room
1760	Radioactive Waste Storage Room, Healthcare	3310	Radiographic Chest Room
1770	Nuclear Medicine Scanning Room	3320	Radiology Computer Systems Room
1780	Patient Dose/Thyroid Uptake Room	3330	Ultrasound Room
1790	Radiopharmacy	3340	Whole Body Scanning Room
1800	Radiation Therapy, Mold Fabrication Shop	3350	Angiographic Instrument Room
1810	Hearth and Lung Diagnostic and Treatment Spaces	3360	Angiographic System Component Room
1820	Cardiac Catheter Instrument Room	3370	Computed Radiology Reader Area
1830	Cardiac Catheter Control Room	3380	X-Ray, Digital Image Storage Space
1840	Cardiac Electrophysiology Room	3390	CT power and Equipment Room
1850	Echocardiograph Room	3400	Image Reading Room
1860	Extended Pulmonary Function Testing Laboratory	3410	Mammography Processing Room
1870	Pacemaker ICD Interrogation Room	3420	MRI Equipment Storage Room
1880	Procedure Viewing Area	3430	PET/CT Control Room
1890	Pulmonary Function Treadmill Room	3440	Radiographic Darkroom
1900	Respiratory Therapy Clean-up Room	3450	Viewing/Consultation Room, Diagnostic Imaging
1910	General Diagnostic Procedure and Treatment Spaces	3460	Equipment Calibration Space, Radiation Diagnostic and Therapy
1920	Endoscopy/Gastroenterology Spaces	3470	Linear Accelerator Component Room, Healthcare
1930	Clinical Laboratory Spaces	3480	Linear Accelerator Room, Healthcare
1940	Pharmacy Spaces	3490	Nuclear Medicine Dose Calibration Space
1950	Rehabilitation Spaces	3500	Nuclear Medicine Patient “Hot” Waiting Room
1960	Medical Research and Development Spaces	3510	Radiation Dosimetry Planning Room
1970	Chemistry Laboratories	3520	Radium Cart Holding Space
1980	Physical Sciences Laboratories	3530	Sealed Source Room
1990	Earth and Environmental Sciences Laboratories	3540	Brachytherapy Room
2000	Psychology Laboratories	3550	Cardiac Catheter System

			Component Room
2010	Dry Laboratories	3560	Cardiac Catheter Laboratory
2020	Wet Laboratories	3570	Cardiac Testing Room
2030	Biosciences Laboratories	3580	EKG Testing Room
2040	Astronomy Laboratories	3590	Microvascular Laboratory
2050	Forensics Laboratories	3600	Pacemaker/Holter Monitor Room
2060	Bench Laboratories	3610	Pulmonary Function Testing Laboratory
2070	Integration Laboratories	3620	Pulmonary Screening Room
2080	Laboratory Storage Spaces	3630	Respiratory Inhalation Cubicle
2090	Office Spaces	3640	Eye and Ear Healthcare Spaces
2100	Dedicated Enclosed Workstation	3650	Surgical Spaces
2110	Open Team Setting	3660	Clinical Laboratory Support Spaces
2120	Shared Equipment Station	3670	Medical Services Logistic Spaces
2130	Banking Spaces	3680	Dental Spaces
2140	Automatic Teller Machine Space	3690	Press Conference Room
2150	Trading Spaces	3700	War Room
2160	Demonstration Spaces	3710	Waiting Space
2170	Checkout Space	3720	Waiting Room
2180	Fitting Space	3730	Office Service
2190	Auction Room	3740	Shared Open Workstation
2200	Commercial Service and Repair Spaces	3750	General File and Storage
2210	Hotel, Motel, Hostel, and Dormitory Service Spaces	3760	Lookout Gallery
2220	Hotel Residence Room	3770	Bank Teller Space
2230	Commercial Support Spaces	3780	Vault
2240	Dormitory	3790	Trading Floor
2250	Information Counter	3800	Sales Spaces
2260	Post Office Space	3810	Display Space
2270	Mail Room Space	3820	Vending Machine Area
2280	Conference Room	3830	Pet Shop Animal Space
2290	Grooming Activity Spaces	3840	Makeup Space
2300	Haircutting Space	3850	Food Service
2310	Cooking Spaces	3860	Kitchen Space
2320	Food Preparation Space	3870	Cooking Space
2330	Dishwashing Station	3880	Dining and Drinking Spaces
2340	Dining Room	3890	Banquet Hall
2350	Food Court	3900	Snack Bar
2360	Salad Bar	3910	Liquor Bar
2370	Beverage Station	3920	Table Bussing Station
2380	Serving Station	3930	Vending Perishable Product Space
2390	Cafeteria Vending Space	3940	Tray Return Space
2400	Food Discard Station	3950	Coffee stations
2410	Child Care Spaces	3960	Daycare sickroom
2420	Child Day Care Space	3970	Play Room
2430	CLD–Child Care	3980	Resting Spaces
2440	Rest Area	3990	Break Room
2450	Laundry/Dry Cleaning Space	4000	Smoking Space
2460	Locker Room	4010	Filing Space
2470	Supply Room	4020	Unit Storage

2480	On-call Room	4030	Bathroom
2490	Shower Space	4040	Toilet Space
2500	Ablution Room	4050	Combination Toilet and Bathing Space
2510	Mud Room	4060	Laundry Room
2520	Bedroom	4070	Mental Health Resident Bedroom
2530	Mental Health Resident Bedroom, Bariatric	4080	Nursery
2540	Kitchen		

GeneralSpaceUsageType
Code list identically specified as GeneralSpaceFuntionType

D.2 TransitionSpace

TransitionlSpaceClassType			
Code list derived from OmniClass Table 13			
1000	Horizontal Transition	1010	Vertical Transition

TransitionlSpaceFuntionType			
Code list derived from OmniClass Table 13			
1000	Corridor	1070	Concourse
1010	Breezeway	1080	Moving walkway
1020	Box Lobby	1090	Entry Lobby
1030	Elevator Lobby	1100	Jet way
1040	Landing	1110	Elevator Shaft
1050	Aisle	1120	Stair
1060	Ramp	1130	Chute

TransitionSpaceUsageType
Code list identically specified as TransitionlSpaceFuntionType

D.3 ConnectionSpace

ConnectionSpaceFunctionType			
Code list derived from OmniClass Table 13			
1000	Door	1060	Sally port
1010	Vestbule	1070	

ConnectionSpaceClassType			
Code list derived from OmniClass Table 13			
1000	Door	1020	Sally port
1010	Vestbule		

ConnectionSpaceUsageType

Code list identically specified as ConnectionSpaceFunctionType

D.3 AnchorSpace

AnchorSpaceClassType			
Code list derived from OmniClass Table 13			
1000	Vestibule	1020	Gate

AnchorSpaceFuntionType			
Code list derived from OmniClass Table 13			
1000	Entry Vestibule	1020	Gate
1010	Exterior door	1030	Emergency door

AnchorSpaceUsageType
Code list identically specified as AnchorSpaceUsageType

Annex E
(Informative)
IFC, CityGML LoD 4, and IndoorGML

E.1 Introduction

Prior to IndoorGML, several standards and data modelling have been developed for indoor space. They are summarized as Table E-1.

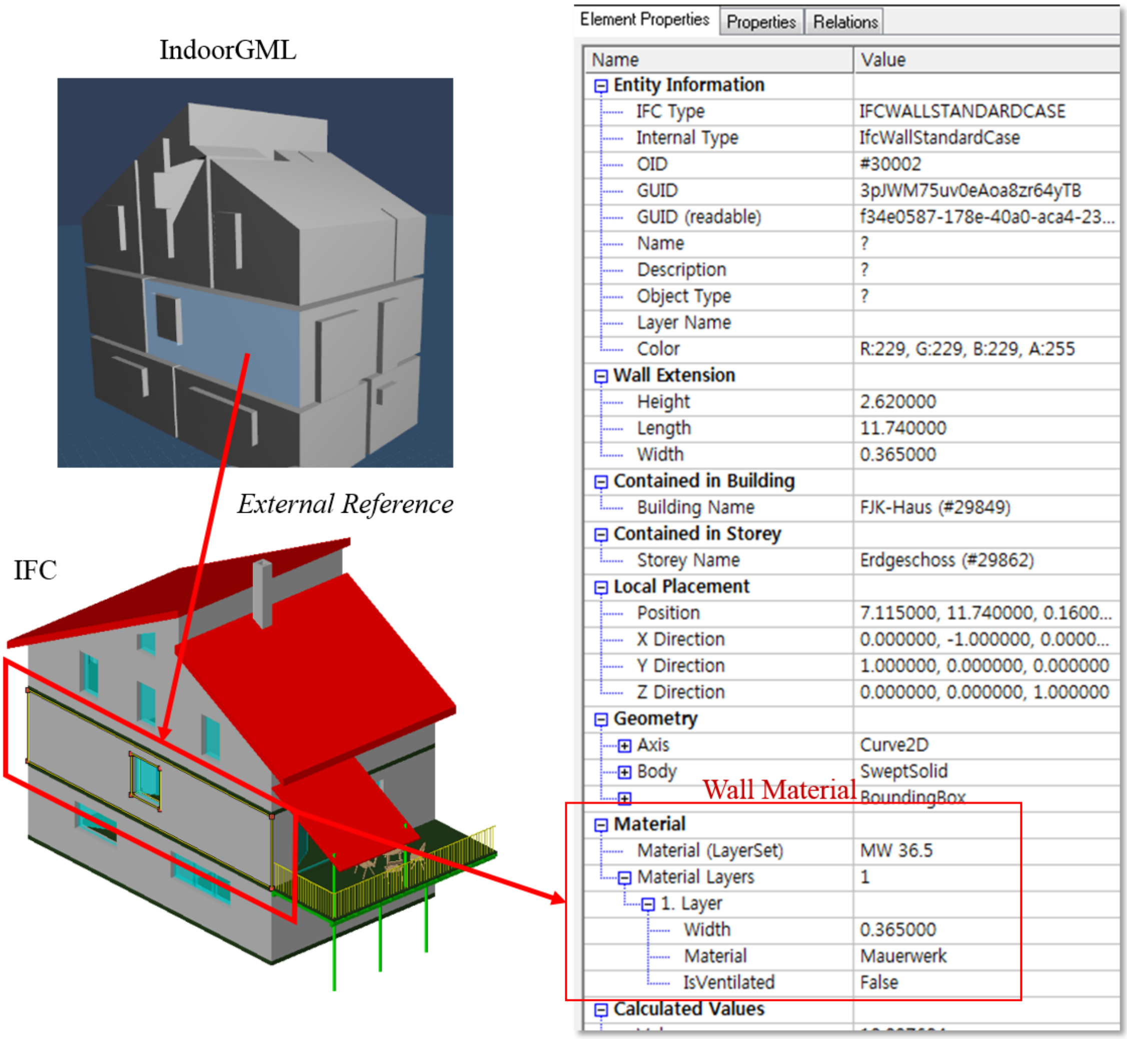
Standards	Description
IFC	Open and neutral data format for open BIM developed by buildingSmart and registered with ISO16739. It is defined in EXPRESS and XML.
CityGML	3D City Model developed by OGC. CityGML LoD 4 covers indoor space. It is defined as an application schema of GML 3.1.1

Table E-1. Related Standards with IndoorGML

E.2 IFC and IndoorGML

Since IFC is an object-oriented data model for building components, it is related with IndoorGML for many aspects. It is therefore important to understand the difference and relationship with IFC for better application of IndoorGML, particularly for two reasons.

First, IndoorGML is based on cellular space model, while IFC is designed for modelling building components. IFC is therefore better than IndoorGML to describe individual building component and can compensate the weakness of IndoorGML. For example, when computing the WiFi signal strength for indoor positioning, we need to have the information about the thickness and materials of walls. While this information is not sufficiently provided by IndoorGML in most cases, IFC basically include the thickness and material information of wall components. Then, the information in IFC can be easily accessed via external references from IndoorGML as shown in Figure E-1.



Name	Value
⊟ **Entity Information**	
IFC Type	IFCWALLSTANDARDCASE
Internal Type	IfcWallStandardCase
OID	#30002
GUID	3pJWM75uv0eAoa8zr64yTB
GUID (readable)	f34e0587-178e-40a0-aca4-23...
Name	?
Description	?
Object Type	?
Layer Name	
Color	R:229, G:229, B:229, A:255
⊟ **Wall Extension**	
Height	2.620000
Length	11.740000
Width	0.365000
⊟ **Contained in Building**	
Building Name	FJK-Haus (#29849)
⊟ **Contained in Storey**	
Storey Name	Erdgeschoss (#29862)
⊟ **Local Placement**	
Position	7.115000, 11.740000, 0.1600...
X Direction	0.000000, -1.000000, 0.0000...
Y Direction	1.000000, 0.000000, 0.000000
Z Direction	0.000000, 0.000000, 1.000000
⊟ **Geometry**	
⊞ Axis	Curve2D
⊞ Body	SweptSolid
⊞	BoundingBox
⊟ **Material**	
Material (LayerSet)	MW 36.5
⊟ Material Layers	1
⊟ 1. Layer	
Width	0.365000
Material	Mauerwerk
IsVentilated	False
⊟ **Calculated Values**	

Figure 38 [E-1] External reference to IFC object for material information

```
<CellSpaceBoundaryMember>
<indoorcore:CellSpaceBoundary gml:id="CSB1">
<geometry3D>
<gml:CompositeSurface gml:id="CS1">
</gml:CompositeSurface>
</geometry3D>
<externalReference>
<informationSystem>FJK-Haus.ifc</informationSystem>
        <externalObject>
          <name>3pJWM75uv0eAoa8zr64yTB</name>
        </externalObject>
</externalReference>
</indoorcore:CellSpaceBoundary>
</CellSpaceBoundaryMember>
```

Figure 39 [E-2]: IndoorGML document containing external reference to IFC object (GUID of IFC is used as the foreign key)

```
#33175=IFCEXTRUDEDAREASOLID(#33168,#33173,#33174,0.365);
```

```
#33177=IFCSHAPEREPRESENTATION(#11,'Body','SweptSolid',(#33175));
#30002=IFCWALLSTANDARDCASE('3pJWM75uv0eAoa8zr64yTB',#16,$,$,$,#29973,#29982,$);
#33185=IFCCARTESIANPOINT((1.110223024625157E-016,-8.881784197001252E-016,0.));
#33186=IFCBOUNDINGBOX(#33185,1.51,0.3650000000000011,1.385000000000001);
#33187=IFCSHAPEREPRESENTATION(#11,'','BoundingBox',(#33186));
```

Figure 40 [E-3]: IndoorGML document containing external reference to IFC object

Second, IFC may serve as an important source for building IndoorGML data as CityGML data can be partially derived from IFC. Although it would be impossible to convert IFC data to IndoorGML data in a fully automatic way due to the gap between two models, we expect that a careful definition of MVD of IFC may considerably improve the building process of IndoorGML from IFC.

E.3 CityGML and IndoorGML

CityGML LoD 4 shares many common aspects with IndoorGML, since they both deal with indoor space. In development of IndoorGML, it has been rigorously considered to avoid duplication of two standards. The major focus of IndoorGML is the cellular representation of indoor space including topology and multi-layered space model, while CityGML aims to represent topographic aspects of indoor space. It means that most missing parts of IndoorGML, such as visualization can be complemented by integration with CityGML. The integration can be easily handled by external references of IndoorGML.

Revision history

Date	Release	Authors	Paragraph modified	Description
Dec. 11, 2010	Discussion Paper for Indoor Navigation	Claus Nagel, Thomas Becker, Robert Kaden, Ki-Joune Li, Jiyeong Lee, Thomas H. Kolbe		Preliminary Document: Requirements and Space-Event Modeling for Indoor Navigation
June 6, 2012	v.0.1	Jiyeong Lee, Ki-Joune Li, Thomas H. Kolbe, Sisi Zlatanova, Jeremy Morley, Claus Nagel, Robert Kaden		Initial version for draft summary in slide format
Sept. 7, 2012	v.0.2	Jiyeong Lee, Ki-Joune Li, Thomas H. Kolbe, Sisi Zlatanova, Jeremy Morley, Claus Nagel, Robert Kaden		Revised version for draft summary and discussed at teleconference on Sept. 7, 2012
Oct. 9, 2012	v.0.3	Jiyeong Lee, Ki-Joune Li, Thomas H. Kolbe, Sisi Zlatanova, Jeremy Morley, Thomas Becker, Claus Nagel, Robert Kaden		Revised version for draft summary and discussed at Seoul Meeting on Oct. 9, 2012
Jan. 15, 2013	v.0.5	Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Thomas H. Kolbe	- navigation module - Deletion of route constraints	Initial version of full draft and discussed at Redland meeting
Jan. 31, 2013	v.0.6.0	Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Thomas H. Kolbe	- Data Model: navigation module - XML schema	Revised to reflect the discussion at Redland meeting
Feb. 5, 2013	v.0.6.1 v.0.6.2	Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Thomas H. Kolbe	- Data Model - XML Schema	Revised to reflect the teleconference meeting on Jan. 31
Mar. 11, 2013	v.0.6.3	Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Jeremy Morley, Thomas H. Kolbe	- Chapter 6, 7, 8, and 9	Revised to reflect the teleconference meeting on Feb. 5 and discussed at Abu Dhabi meeting

Date	Release	Authors	Paragraph modified	Description
Mar. 17, 2013	v.0.6.4	Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Thomas H. Kolbe	- Chapter 7, 8, and 9	Revised to reflect the Abu Dhabi meeting and discussed at the teleconference on Mar. 17
May 21, 2013	v.0.6.5	Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Jeremy Morley, Thomas H. Kolbe	- Chapter 7, 8, and 9	Revised to reflect the teleconference on Mar. 17 and discussed at teleconference on May 21
June 12, 2013	v.0.7.0	Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Jeremy Morley, Thomas H. Kolbe	- Chapter 7, 8, and 9	Revised version for Virtual TC meeting.
Sept. 17	v.0.8	Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Jeremy Morley, Claus nagel, Thomas Becker, Thomas H. Kolbe	- Chapter 7	Revised version prepared by editors (improvement of chapter 7)
Nov.11	v.0.8.1	Jiyeong Lee, Ki-Joune Li, Sisi Zlatanova, Jeremy Morley, Claus nagel, Thomas Becker, Thomas H. Kolbe	- Chapter 8	Revised version prepared by editors (improvement of chapter 8)
Dec. 19	v.0.8.2	Ki-Joune Li	- Chapter 2 and Annex B	Revision for conformance requirements and ATS
Apr. 10, 2014	v.0.9.0	Ki-Joune Li	- Chapter 3	Revision to reflect comments from OAB/NA
Apr. 21	v.0.9.1	Jiyeong Lee Ki-Joune Li	- Chapter 8 and 9	Revision to reflect public comments
May 26	v.0.9.2	Jiyeong Lee	- Chapter 8 and 9	Correction of XML shema definition
June 6	v.0.9.3	Ki-Joune Li		Minor revision before finalizing RFC
June 12	v.0.9.4	Jiyeong Lee Ki-Joune Li	-Chapter 8	Correction of UML Diagram and XML Schema
July 10	v.0.9.5	Jiyeong Lee	-Chapter 7 and 8	Removing Direction of transition
Sept.30	v.0.9.6	Carl Reed		Revision to correct typos and format
Oct. 2	v.0.9.7	Ki-Joune Li		Revision to reflect the comments received during TC voting

Oct. 13	v.0.9.8	Jiyeong Lee Ki-Joune Li		Revision to reflect the comments received during TC voting, modification of URI name