

OGC® DOCUMENT: 19-004

External identifier of this OGC® document: <http://www.opengis.net/doc/dp/indoorgml-anchornode>

OGC®
Making location count.

ANCHOR NODE EXTENSION IN INDOORGML - SEAMLESS NAVIGATION BETWEEN INDOOR AND OUTDOOR SPACE

DISCUSSION PAPER

APPROVED

Submission Date: 2019-02-05

Approval Date: 2019-02-28

Publication Date: 2019-12-11

Editor: Kyoung-Sook Kim, Jiyeong Lee

Warning: This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

License Agreement

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

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

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

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

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

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Copyright notice

Copyright © 2021 Open Geospatial Consortium
To obtain additional rights of use, visit <http://www.ogc.org/legal/>

Note

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

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.

CONTENTS

| | |
|---|-----|
| I. ABSTRACT | v |
| II. KEYWORDS | v |
| III. SECURITY CONSIDERATIONS | vi |
| IV. SUBMITTING ORGANIZATIONS | vii |
| V. SUBMITTERS | vii |
| 1. NORMATIVE REFERENCES | 9 |
| 2. TERMS, DEFINITIONS AND ABBREVIATED TERMS | 11 |
| 2.3. Abbreviated terms | 11 |
| 3. INTRODUCTION | 14 |
| 4. THE CONNECTION BETWEEN INDOOR AND OUTDOOR SPACES | 16 |
| 5. SEAMLESS NAVIGATION MODULE OF INDOORGML USING ANCHOR NODE | 19 |
| 5.1. Conversion method of the coordinate reference system | 20 |
| 5.2. UML diagram of the seamless navigation module | 23 |
| 6. USE CASES | 34 |
| 6.1. Use case with CityGML | 34 |
| 6.2. Use case with a specification for the pedestrian network model from Japan government | 39 |
| 6.3. Use case with GDF 5.0 | 45 |
| ANNEX A (NORMATIVE) XML SCHEMA FOR INDOORGML SEAMLESSNAVIGATION MODULE | 53 |
| ANNEX B (NORMATIVE) SEAMLESSNAVIGATION MODULE EXAMPLE DOCUMENT | 57 |
| ANNEX C (NORMATIVE) REVISION HISTORY | 62 |
| ANNEX D (INFORMATIVE) | 64 |
| D. BIBLIOGRAPHY | 64 |

LIST OF FIGURES

| | |
|--|----|
| Figure 1 – Anchor node connecting indoor and outdoor networks (OGC 14-005r5) | 16 |
| Figure 2 – Structure space model (OGC 14-005r5) | 19 |
| Figure 3 – The concept of SeamlessNavigation module | 20 |
| Figure 4 – Example of transform methods (2-dimensional case) | 21 |
| Figure 5 – Example of Yaw-Pitch-Roll rotation | 22 |
| Figure 6 – Example of Thin and Thick model (OGC 14-005r5) | 24 |
| Figure 7 – IndoorGML SeamlessNavigation module | 25 |
| Figure 8 – UML diagram for SeamlessNavigation module (simple version) | 26 |
| Figure 9 – UML diagram for SeamlessNavigation module based on IndoorGML modules | 27 |
| Figure 10 – The process of CRS conversion | 29 |
| Figure 11 – Example of mapping relation between ExternalAnchorState and externalNetworkReference | 31 |
| Figure 12 – UML diagram of transportation model in CityGML 2.0 (OGC 12-019) | 34 |
| Figure 13 – The use case of the CityGML Transportation module | 35 |
| Figure 14 – CityGML use case site – AIST Tokyo Waterfront Annex | 36 |
| Figure 15 – Part of sample data for AnchorState (seamlessNaviSample.gml) | 37 |
| Figure 16 – Part of sample data for ExternalAnchorState (seamlessNaviSample.gml) | 38 |
| Figure 17 – Part of sample data for TransportationComplex (cityTransSample.gml) | 38 |
| Figure 18 – Part of sample data for AnchorLink (seamlessNaviSample.gml) | 39 |
| Figure 19 – An example of mapping PNspec to IndoorGML in a stairs case_ | 40 |
| Figure 20 – An example of mapping PNspec to IndoorGML in changing points of barriers (gradient) case | 40 |
| Figure 21 – An example of mapping PNspec to IndoorGML in changing points of barriers (step) case | 41 |
| Figure 22 – An example of mapping PNspec to IndoorGML in entrance of building case | 41 |
| Figure 23 – PNSpec use case site – Tokyo station (Node case) | 42 |
| Figure 24 – Part of PNSpec sample data | 43 |
| Figure 27 – Part of sample data for ExternalAnchorState | 45 |
| Figure 28 – UML diagram of the overall conceptual data model of GDF 5.0 (GDF 5.0, 2011) | 46 |
| Figure 29 – UML diagram of the conceptual data model for ‘Relationships’ with ‘BuildingAlongRoadElement’ | 47 |
| Figure 30 – UML diagram of the conceptual data model for EntryPoint of General Feature | 48 |
| Figure 31 – UML diagram of the conceptual data model for ‘PlanarTopoSimpleFeature’ | 49 |
| Figure 32 – GDF 5.0 Use case site – AIST Tokyo Waterfront Annex | 50 |
| Figure 33 – GDF 5.0 sample data (GDF_5_0_sample.xml) | 51 |
| Figure 34 – Part of sample data for ‘ExternalAnchorState’ | 51 |

ABSTRACT

This OGC discussion paper provides an extension module of OGC Indoor Geography Markup Language (IndoorGML) for the seamless navigation between indoor and outdoor spaces. The OGC IndoorGML standard has an issue on the data model that affects the connection of indoor and outdoor spaces via an “Anchor Node,” which is a conceptual part for connecting indoor and outdoor spaces. This discussion paper aims to show use cases of how IndoorGML can connect with other geospatial standards that represent outdoor spaces (and road networks), such as OGC City Geography Markup Language (CityGML) and version 5.0 of the Geographic Data Files (GDF) format.

KEYWORDS

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

ogcdoc, OGC document, OGC, IndoorGML, Indoor space, Outdoor space, Seamless navigation, CityGML

SECURITY CONSIDERATIONS

No security considerations have been made for this document.

SUBMITTING ORGANIZATIONS

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

- National Institute of Advanced Industrial Science and Technology
- The University of Seoul
- All for Land Inc

SUBMITTERS

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

| NAME | AFFILIATION |
|----------------|--|
| Kyong-Sook Kim | National Institute of Advanced Industrial Science and Technology |
| Teahoon Kim | National Institute of Advanced Industrial Science and Technology |
| Jiyeong Lee | University of Seoul |
| Hye-Young Kang | All for Land Inc. |

1

NORMATIVE REFERENCES

NORMATIVE REFERENCES

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO: ISO 14825:2011, *Intelligent transport systems – Geographic Data Files (GDF) – GDF5.0*. International Organization for Standardization, Geneva (2011). <https://www.iso.org/standard/54610.html>

OGC: OGC 14-005r5, *OGC Indoor Geography Markup Language (IndoorGML) – with Corrigendum 1.0.3* (2018)

Gerhard Gröger, Thomas H. Kolbe, Claus Nagel, Karl-Heinz Häfele: OGC 12-019, *OGC City Geography Markup Language (CityGML) Encoding Standard*. Open Geospatial Consortium (2012). https://portal.ogc.org/files/?artifact_id=47842

2

TERMS, DEFINITIONS AND ABBREVIATED TERMS

TERMS, DEFINITIONS AND ABBREVIATED TERMS

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

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this Best Practice.

2.1. Indoor space

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

2.2. Cellular Space

A space where location is identified by a cell identifier

2.3. ABBREVIATED TERMS

The following abbreviated terms are used in this discussion paper:

- AIST National Institute of Advanced Industrial Science and Technology
- CityGML City Geography Markup Language
- CRS Coordinate Reference System
- GDF Geographic Data Files
- GML Geography Markup Language
- IndoorGML Indoor Geography Markup Language
- ISO International Organization for Standardization
- OGC Open Geospatial Consortium

- OSM OpenStreetMap
- UML Unified Modeling Language

3

INTRODUCTION

INTRODUCTION

This OGC document tries to extend the OGC IndoorGML core and navigation modules for supporting seamless navigation from an indoor to an outdoor space, and vice versa. Although there are many approaches to determine the indoor or outdoor location of a user, few services support both indoor and outdoor space due to the absence of a data model that covers/connects them both. The scope of this discussion paper is to design an extension data model of IndoorGML for linking between two anchor parts of indoor and outdoor space. This paper consists of three parts:

- The concept of anchor nodes for the connection between indoor and outdoor space,
- An extension of IndoorGML schema for seamless navigation, and
- A use case with other geospatial data model standards: CityGML 2.0, Geographic Data Files 5.0, and the specification for a pedestrian network model from the Government of Japan [2]

4

THE CONNECTION BETWEEN INDOOR AND OUTDOOR SPACES

THE CONNECTION BETWEEN INDOOR AND OUTDOOR SPACES

Designing a converged data model to represent indoor and outdoor spaces is one of the typical issues on geographic information systems. In particular, a route navigation service is primarily associated with the network model of connectivity of roads. Compared to many standard formats that represent outdoor networks, only IndoorGML provides a standard model to describe the connectivity of components in indoor space. Since IndoorGML has been published, various studies have been carried out to express indoor and outdoor space connections.

Figure 1 – Anchor node connecting indoor and outdoor networks (OGC 14-005r5)

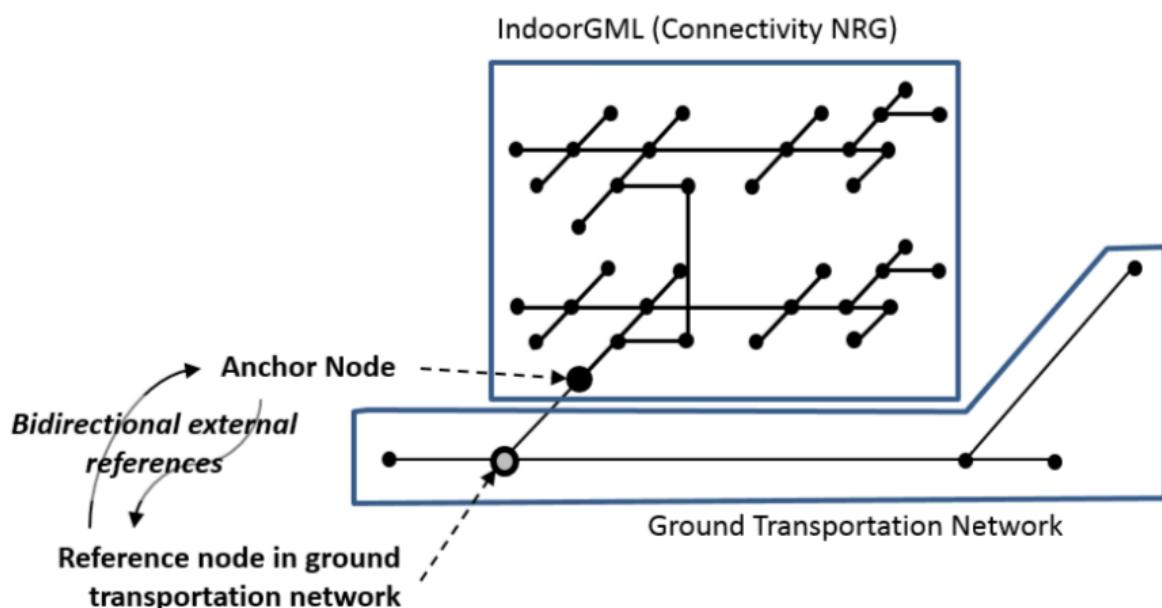


Figure 16: Anchor Node Connecting Indoor and Outdoor Networks [27]

As shown in Figure 1, IndoorGML introduces a simple concept of an “anchor node” for representing indoor and outdoor connections. For example, an “entrance” is represented as an anchor node, a topological node to connect an indoor and outdoor element.

However, there is no element in the IndoorGML Core (and Navigation) model to represent anchor node, and specific examples of how to apply IndoorGML for connecting outdoor elements, such as roads, junctions, and pedestrian ways, are excluded in the IndoorGML standard document. In [4], the indoor and outdoor connections are expressed by extending the *State* and *Transition* of the core module of IndoorGML to *SpecialState* (Anchor node) and *SpecialTransition* (Anchor edge). However, this extension brings cost overruns when constructing and managing all indoor and outdoor data in/for an IndoorGML document.

This discussion paper proposes an IndoorGML extension model to interconnect indoor and outdoor models for seamless navigation by defining anchor node in the model. To this end, this document includes:

- a) Definition of a conversion matrix between indoor and outdoor coordinate systems
- b) Definition of the element of anchor node that extends IndoorGML core and navigation modules

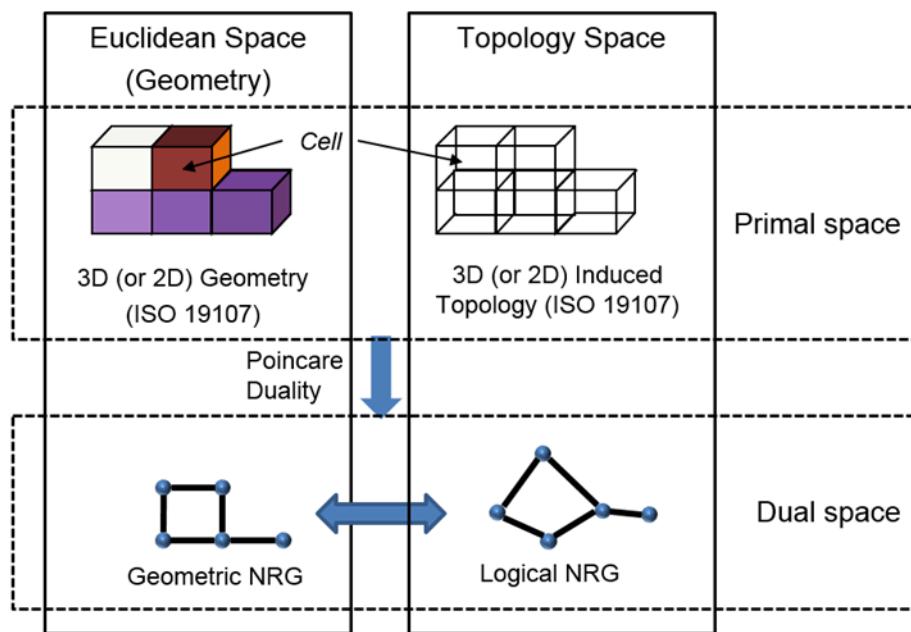
5

SEAMLESS NAVIGATION MODULE OF INDOORGML USING ANCHOR NODE

SEAMLESS NAVIGATION MODULE OF INDOORGML USING ANCHOR NODE

This section describes an IndoorGML SeamlessNavigation module for seamless navigation between indoors and outdoors.

Figure 2 – Structure space model (OGC 14-005r5)



OGC IndoorGML provides a standard data model for indoor space with two spatial models, as shown in Figure 2: Euclidean Space represents the shape of a three-dimensional (3D) cell space; Topology Space represents connectivity between cell spaces. Topology represents a duality transformation of the 3D cell space and is an essential component for indoor navigation and routing system. 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. In the current version of IndoorGML, a gate or entrance of building that connects indoor and outdoor spaces is represented by an *AnchorSpace* instance and can be represented by a *State* instance in dual space. However, the connectivity between an outdoor network and an indoor cell of the *AnchorSpace* class cannot be represented by the elements in IndoorGML.

Figure 3 – The concept of SeamlessNavigation module



In this discussion paper, a SeamlessNavigation model as an extension of IndoorGML is designed for making connections with other standards that represent outdoor spaces, as shown in Figure 3. Unlike defining a unified integration model, the SeamlessNavigation model defines a new element which has the following attributes to support the seamless traveling between indoor and outdoor spaces:

- Parameters for the conversion of coordinate reference systems
- External reference to the outdoor transportation network

5.1. CONVERSION METHOD OF THE COORDINATE REFERENCE SYSTEM

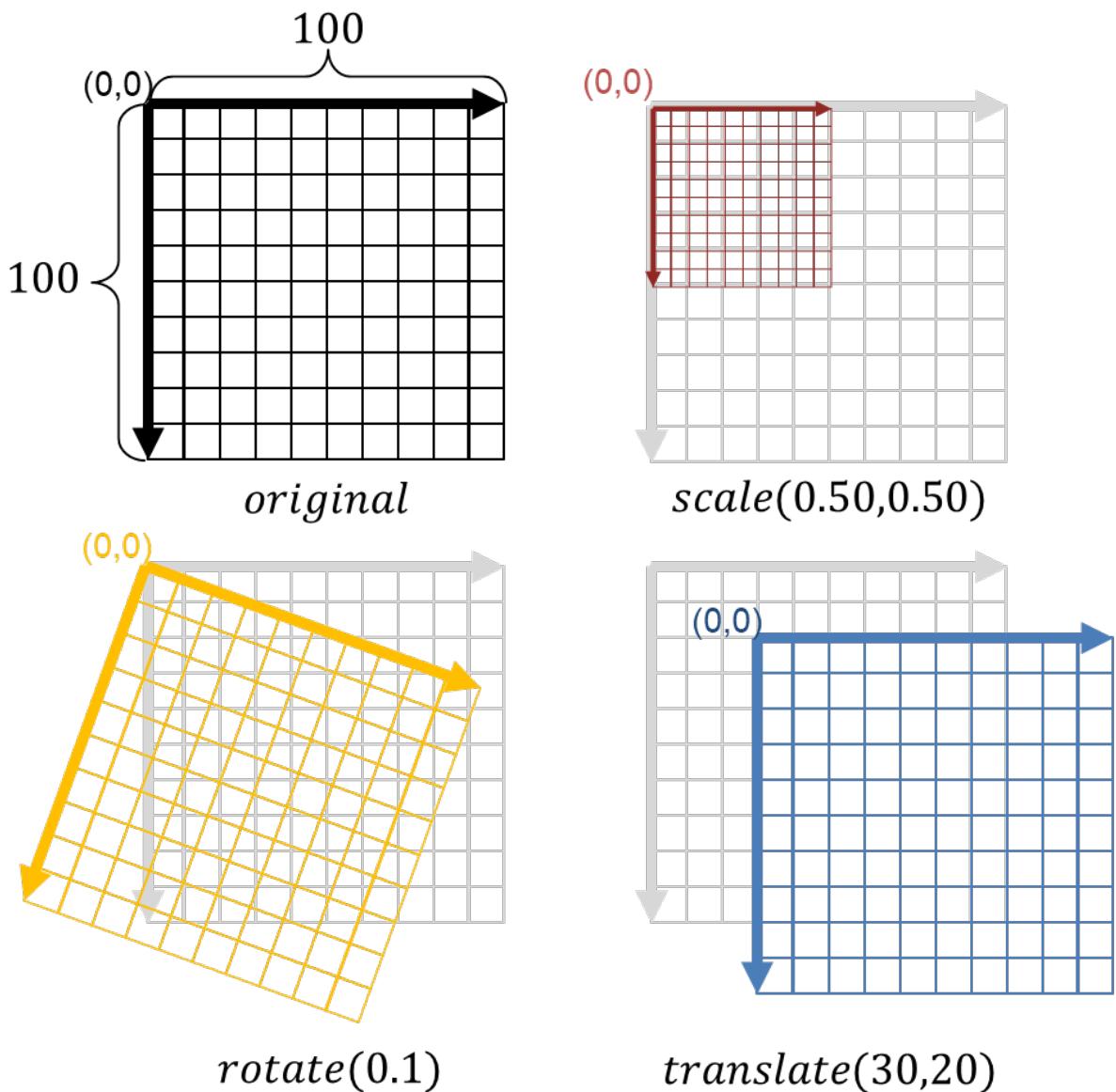
The conversion of the Coordinate Reference System (CRS) is an important process for seamless navigation between indoor and outdoor coordinate systems. In cases where the global CRS is used for indoor space, the conversion parameters are not necessary. However, many building datasets are represented in their own local CRS. In the case of using the local CRS, four parameters are required for Cartesian coordinate system conversion:

- the origin point of target CRS (or global CRS) $P_o(x_0, y_0, z_0)$,
- rescaling factor $R(s_x, s_y, s_z)$,
- rotation angles $A(\alpha, \beta, \gamma)$, along x, y, z -axis, and
- translation vector $T(t_x, t_y, t_z)$

Firstly, the origin P_o is required to perform the transformation. Next, a scale value R , between the local coordinate system and the global coordinate system, is required. Thirdly, the rotation angle of each axis A is required for the rotation movement between the coordinate systems.

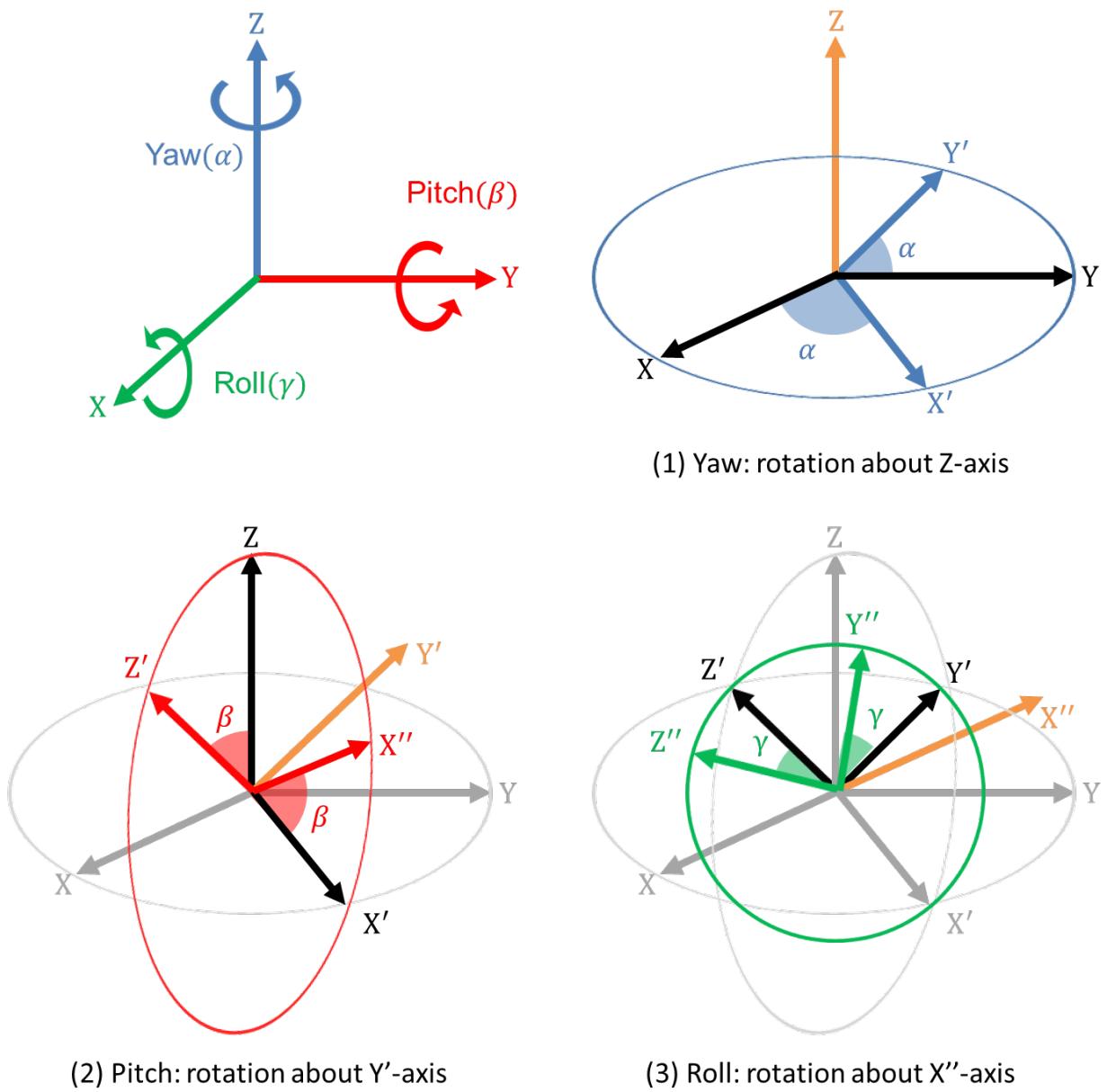
Lastly, a translation vector T is given for parallel movement between coordinate systems. Figure 4 shows examples of conversion methods.

Figure 4 – Example of transform methods (2-dimensional case)



Unlike scaling and translation, the rotation is affected by the order in which the parameters (rotation angles) are applied. Typically, the Euler angle for 3D rotation described in [1,5] can be used. Euler angles are described as a sequence of rotations about three mutually orthogonal coordinate axes fixed in R^3 Space. This discussion paper uses yaw, pitch, and roll rotation, as shown in Figure 5, one of the sequences of Euler angles.

Figure 5 – Example of Yaw-Pitch-Roll rotation



Yaw is a counterclockwise rotation of α about the z -axis, as shown in Figure 5. The rotation matrix is given by:

$$R_z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note that the upper left entries of $R_z(\alpha)$ from a 2D rotation applied to the x and y coordinates, whereas the z coordinate remains constant.

Similarly, a pitch is a counterclockwise rotation of β about the y -axis, and a roll is a counterclockwise rotation of γ about the x -axis, as shown in Figure 5. The rotation matrix of pitch and roll are given by:

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_x(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$$

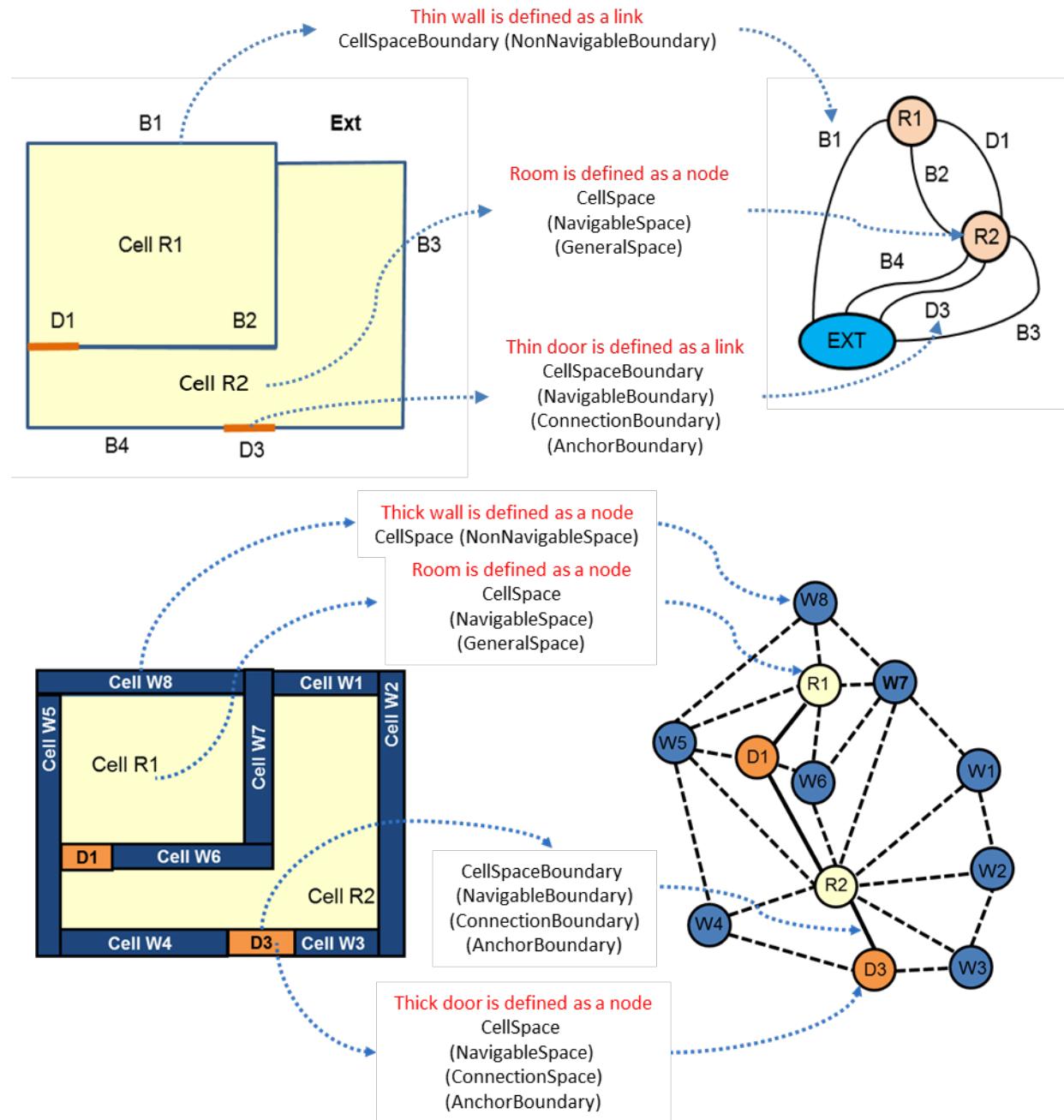
So, a 3D rotation matrix with α, β, γ is defined as follows:

$$R(\alpha, \beta, \gamma) = R_z(\alpha) R_y(\beta) R_x(\gamma) = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix}$$

5.2. UML DIAGRAM OF THE SEAMLESS NAVIGATION MODULE

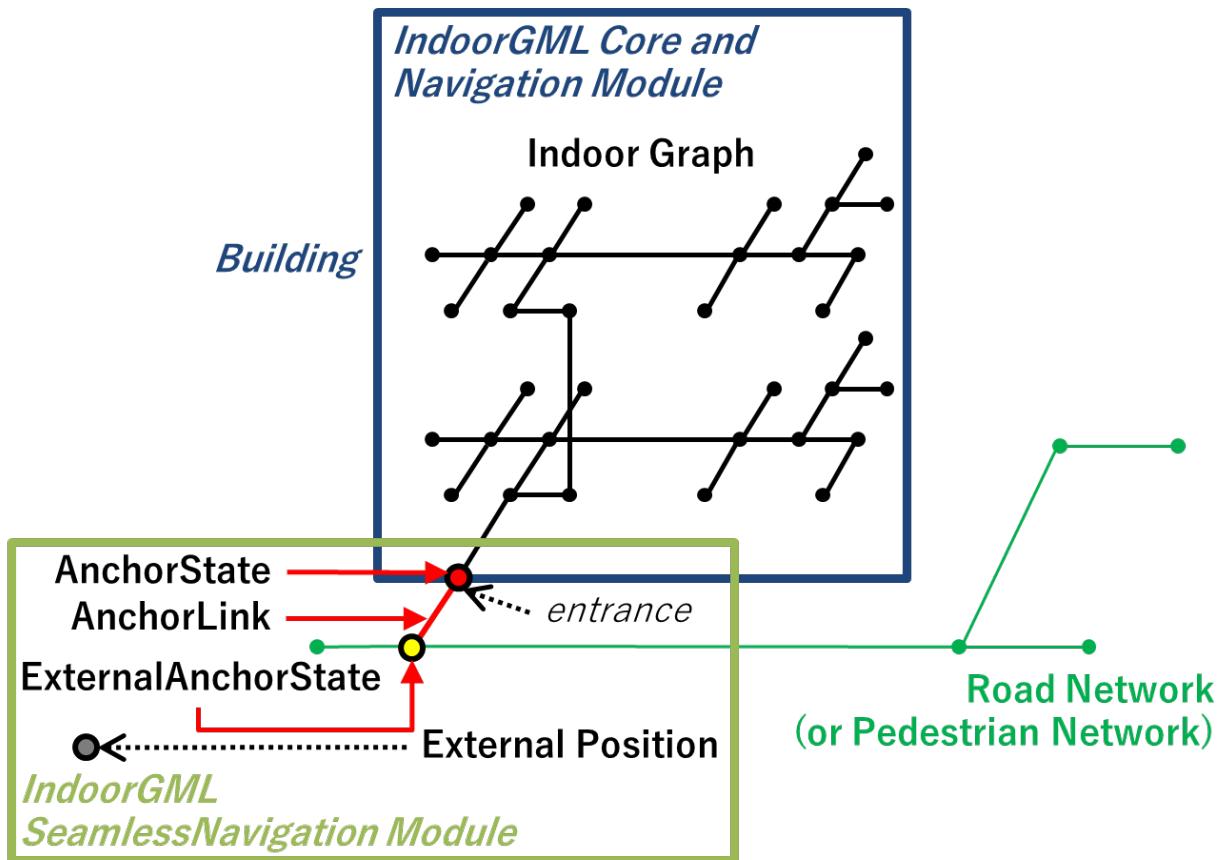
IndoorGML has a thick model that represents the wall thickness of a building and a thin model that does not, as shown in Figure 6. The SeamlessNavigation module can be defined by considering both models.

Figure 6 – Example of Thin and Thick model (OGC 14-005r5)



However, when expressing an entrance with a thin model, a *State* is required in the outdoor space according to the definition of transition. However, since *State* has a duality relation with *CellSpace*, it is necessary to express the outdoor space as *CellSpace* to create a *State* in outdoor space. However, this is not semantically equivalent to *CellSpace* defined in IndoorGML. In conclusion, the entrance should be expressed, as is the *State*, in the door of the thick model.

Figure 7 – IndoorGML SeamlessNavigation module



The proposed SeamlessNavigation module is shown in Figure 7. The SeamlessNavigation module consists of three elements: *AnchorState*, *AnchorLink*, and *ExternalAnchorState*. The UML diagram depicted in Figure 8 and Figure 9 shows the IndoorGML SeamlessNavigation module data model based on the IndoorGML core and navigation module.

Figure 8 – UML diagram for SeamlessNavigation module (simple version)

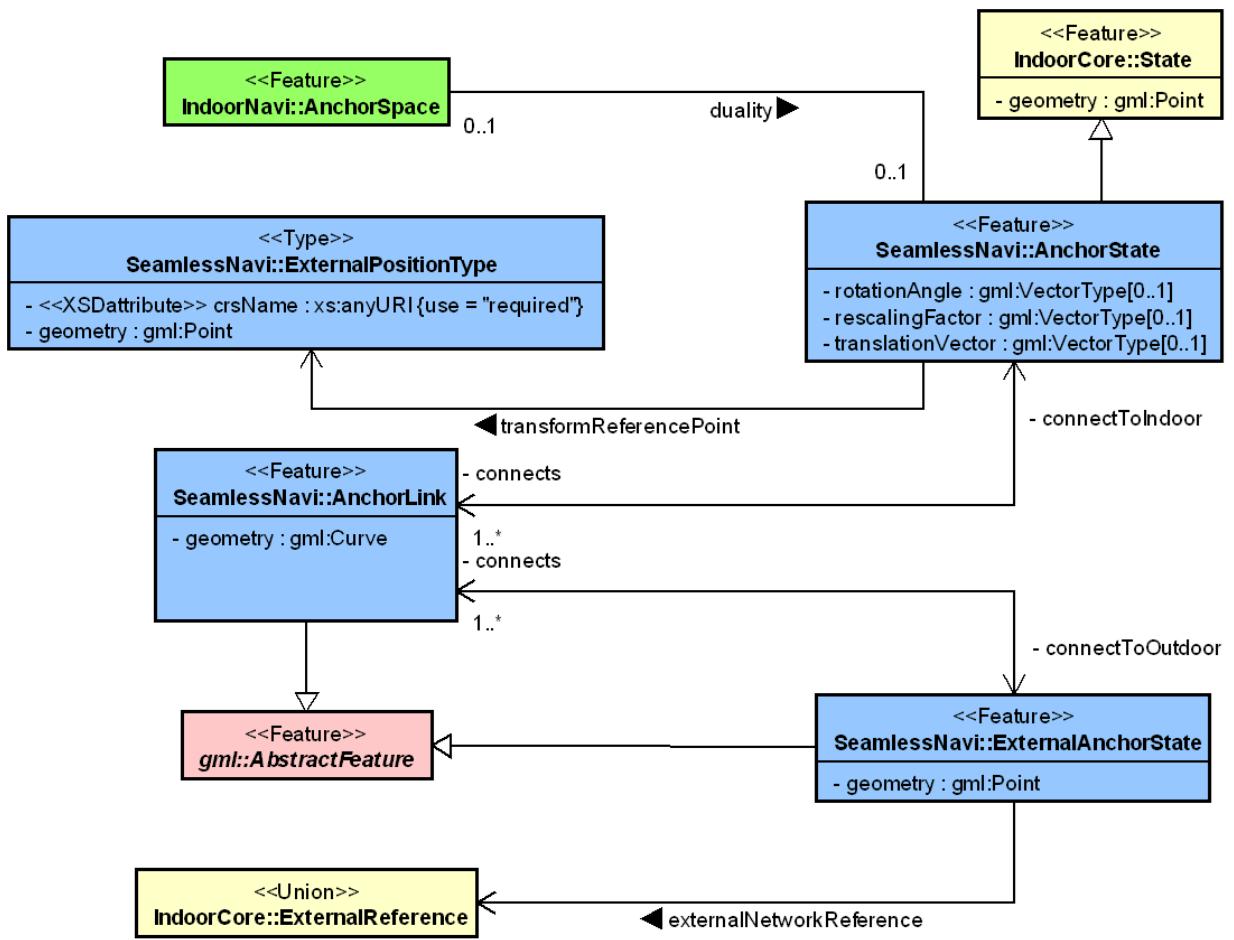
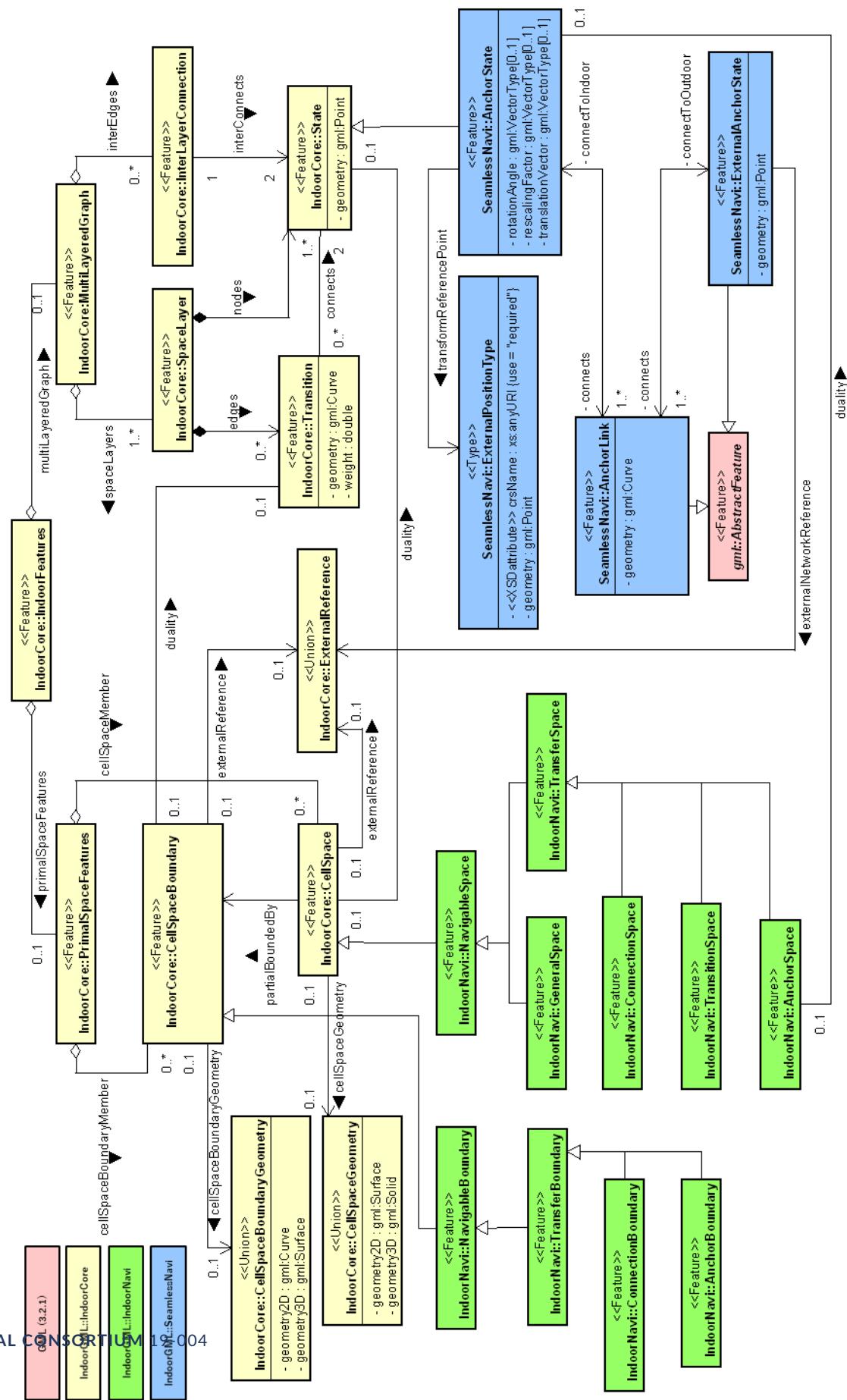


Figure 9 – UML diagram for SeamlessNavigation module based on IndoorGML modules



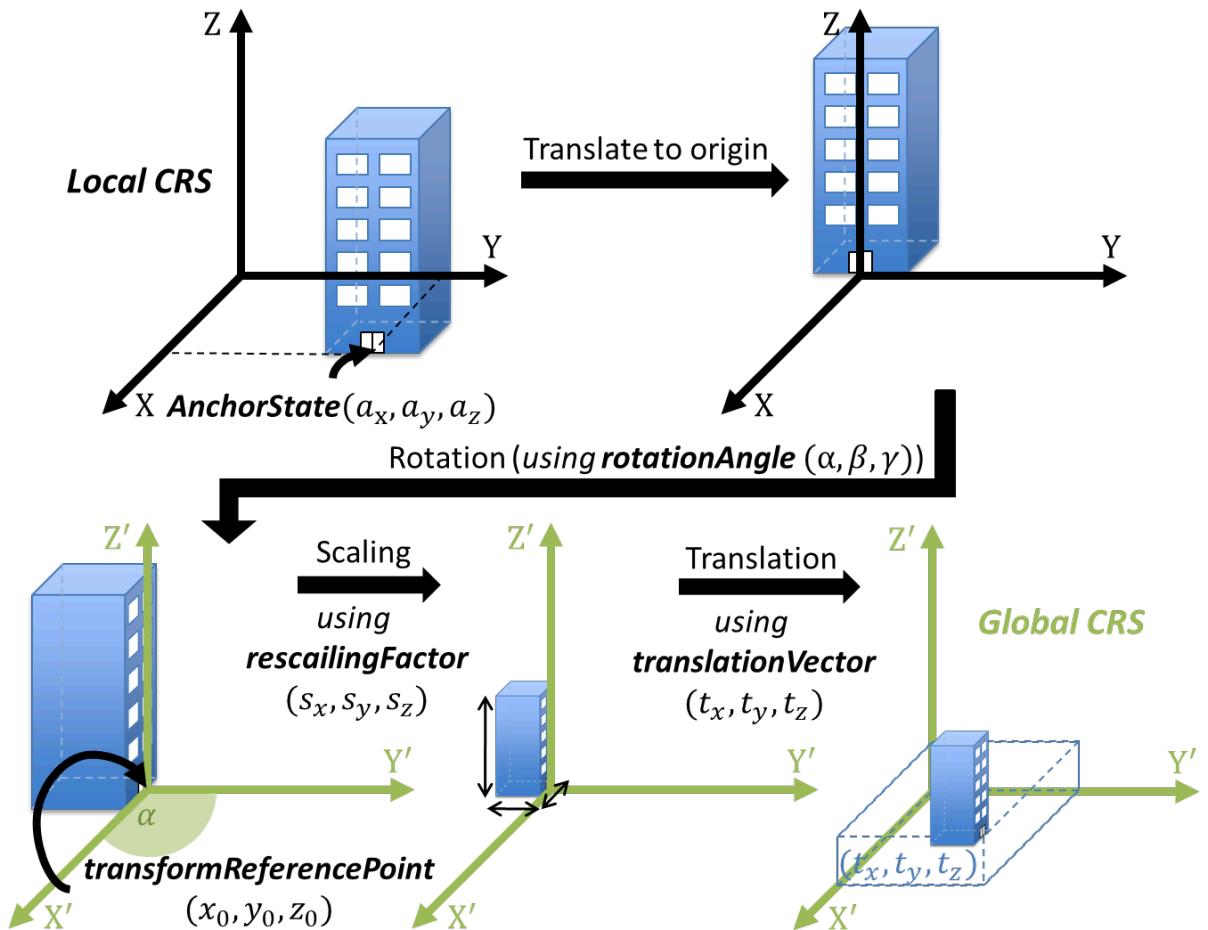
5.2.1. <AnchorState>

AnchorState represents a node that provides the connection between indoor space and outdoor space. It refers to entrance doors. It can be used as a control point for indoor-outdoor integrations. It contains conversion parameters for transforming the local CRS coordinates of indoor geometry. In cases where the global CRS is used for indoor space, the conversion parameters are not necessary. The *transformReferencePoint* element describes a reference point that is used for the conversion. *TransformReferencePoint* is a point in the global CRS. *TransformReferencePoint* is represented geometrically as a *Point* in Geography Markup Language (GML). *TransformReferencePoint* must have an attribute *crsName* to represent the used CRS of the outdoor network. The *duality* element represents an association with the corresponding *AnchorSpace* class, which represents a special opening space. *AnchorState* has a geometry that is derived from *State* class, and it is one of the endpoints of the curve geometry of *AnchorLink*.

```
<xs:element name="AnchorState" type="AnchorStateType" substitutionGroup="IndoorCore:State"/>
<!-- ===== -->
<xs:complexType name="AnchorStateType">
<xs:complexContent>
<xs:extension base="IndoorCore:StateType">
<xs:sequence>
<xs:element name="transformReferencePoint" type="ExternalPositionType"/>
<xs:element name="rotationAngle" type="gml:VectorType" minOccurs="0"/>
<xs:element name="rescalingFactor" type="gml:VectorType" minOccurs="0"/>
<xs:element name="translationVector" type="gml:VectorType" minOccurs="0"/>
<xs:element name="duality" type="AnchorSpacePropertyType" minOccurs="0"/>
<xs:element name="connects" type="AnchorLinkPropertyType" maxOccurs="unbounded"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!-- ===== -->
<xs:complexType name="AnchorState.PropertyType">
<xs:sequence minOccurs="0">
<xs:element ref="AnchorState"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!-- ===== -->
<xs:complexType name="AnchorSpace.PropertyType">
<xs:sequence minOccurs="0">
<xs:element ref="IndoorNavi:AnchorSpace"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!-- ===== -->
<xs:complexType name="ExternalPositionType">
<xs:sequence>
<xs:element name="geometry" type="gml:PointPropertyType"/>
</xs:sequence>
<xs:attribute name="srsName" type="xs:anyURI" use="required"/>
```

</xs:complexType>

Figure 10 – The process of CRS conversion



All AnchorState elements are used for conversion, except the duality and connects elements: transformReferencePoint $p_o(x_0, y_0, z_0)$, rotationAngle $R(s_x, s_y, s_z)$, rescalingFactor $A(\alpha, \beta, \gamma)$, and translationVector $T(t_x, t_y, t_z)$. The conversion using these parameters depends on the order in which they are applied. This document assumes that the transformation is performed in the order, as shown in Figure 10: Rotation Scaling Translation. In the case of rotation, the rotation should be performed after shifting to the origin based on the AnchorState point $p_a(a_x, a_y, a_z)$ for simplification of the problem. Finally, the method to obtain the conversion result, Convert($x, y, z, p_a, p_o, R, S, T$) using the given parameters is as follows:

$$\text{Convert}(x, y, z, p_a, p_o, R, S, T) = R_z(\alpha) R_y(\beta) R_x(\gamma) S(x - a_x, y - a_y, z - a_z) + p_o + T = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta \\ -\sin\beta & \cos\beta \end{bmatrix}$$

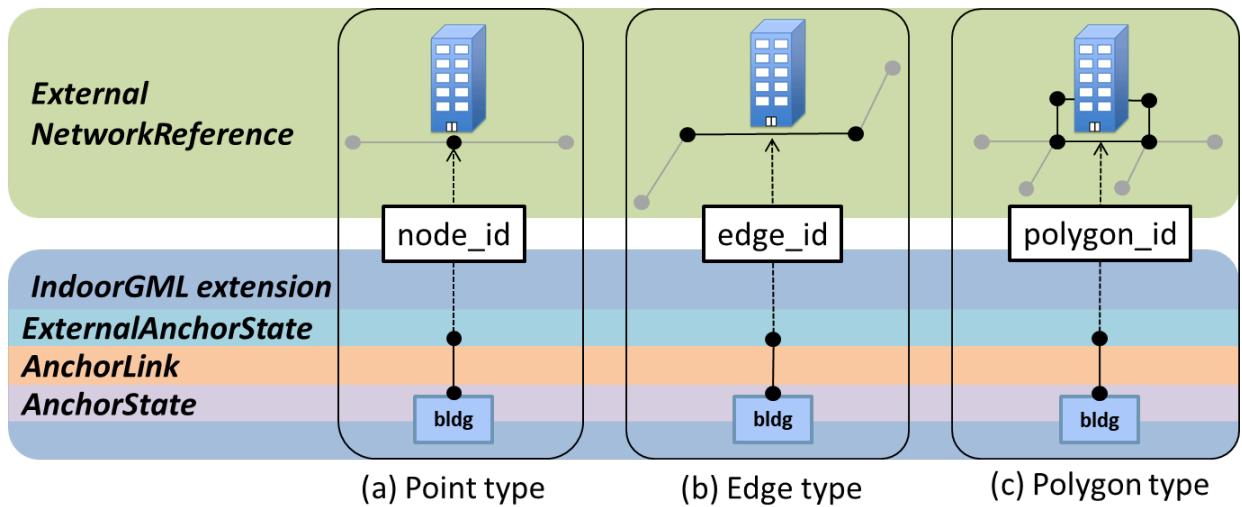
5.2.2. <ExternalAnchorState>

ExternalAnchorState represents a node that represents the position on the outdoor network. It is represented geometrically as a *Point* in GML and it is one of the endpoints of the curve geometry of *AnchorLink*. It also has references to outdoor networks in other standards; e.g., CityGML, GDF, etc.

```
<xss:element name="ExternalAnchorState" type="ExternalAnchorStateType" substitutionGroup="gml:AbstractFeature"/>
<!-- ===== -->
<xss:complexType name="ExternalAnchorStateType">
<xss:complexContent>
<xss:extension base="gml:AbstractFeatureType">
<xss:sequence>
<xss:element name="externalNetworkReference" type="IndoorCore:ExternalReferenceType"/>
<xss:element name="geometry" type="gml:PointPropertyType"/>
<xss:element name="connects" type="AnchorLinkPropertyType" maxOccurs="unbounded"/>
</xss:sequence>
<xss:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xss:extension>
</xss:complexContent>
</xss:complexType>
<!-- ===== -->
<xss:complexType name="ExternalAnchorState.PropertyType">
<xss:sequence minOccurs="0">
<xss:element ref="ExternalAnchorState"/>
</xss:sequence>
<xss:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xss:complexType>
```

Figure 11 depicts an example of mapping relation between *ExternalAnchorState* and *externalNetworkReference* for each case: The shape of *externalNetworkReference* should be represented as one of those types; (a) a point type, (b) an edge type and © a polygon type.

Figure 11 – Example of mapping relation between ExternalAnchorState and externalNetworkReference



In the case of (a) in Figure 11, externalNetworkReference is represented as a point that is the closest to the entrance of the building in the outside network. Similarly, in (b) in Figure 11, externalNetworkReference represents an edge that is the most adjacent to the opening of the building in the outside network. In this case, the geometry of *ExternalAnchorState* should be a point on the edge of externalNetworkReference. Lastly, in © in Figure 11, externalNetworkReference represents a polygon that expresses the area of the building. In this case, the geometry of *ExternalAnchorState* should be a central point of the polygon of externalNetworkReference.

5.2.3. <AnchorLink>

AnchorLink represents an edge between the indoor network and outdoor networks. *AnchorLink* always connects *AnchorState* and *ExternalAnchorState*. For the geometrical representation of an *AnchorLink*, a *Curve* geometric primitive object from the GML is used.

```

<xs:element name="AnchorLink" type="AnchorLinkType" substitutionGroup="gml:AbstractFeature"/>
<!-- ===== -->
<xs:complexType name="AnchorLinkType">
<xs:complexContent>
<xs:extension base="gml:AbstractFeatureType">
<xs:sequence>
<xs:element name="connectToIndoor" type="AnchorStatePropertyType"/>
<xs:element name="connectToOutdoor" type="ExternalAnchorStatePropertyType"/>
<xs:element name="geometry" type="gml:CurvePropertyType"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!-- ===== -->

```

```
<xs:complexType name="AnchorLinkPropertyType">
<xs:sequence minOccurs="0">
<xs:element ref="AnchorLink"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

6

USE CASES

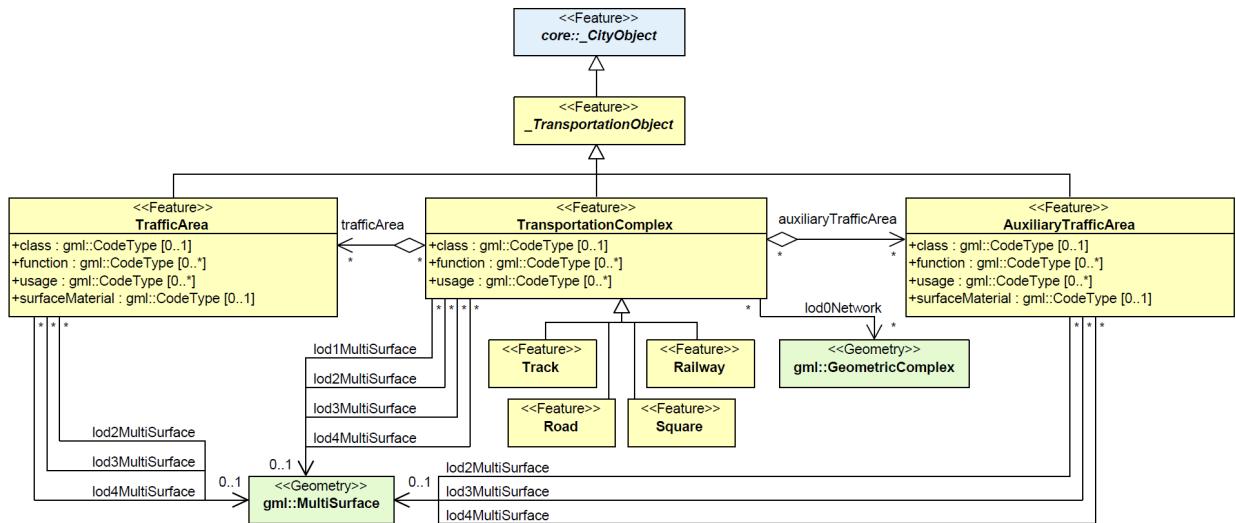
6.1. USE CASE WITH CITYGML

For better understanding, this document provides a use case of the SeamlessNavigation module for the Transportation model of CityGML 2.0. This section briefly introduces the CityGML Transportation model and suggests some guidelines for using the sample data to create the SeamlessNavigation module data.

6.1.1. CityGML 2.0 Transportation model

CityGML is an OGC standard data model and exchange format for storing digital 3D models of cities and landscapes. CityGML supports a transportation model that focuses on thematic as well as on geometrical/topological aspects, as shown in Figure 12.

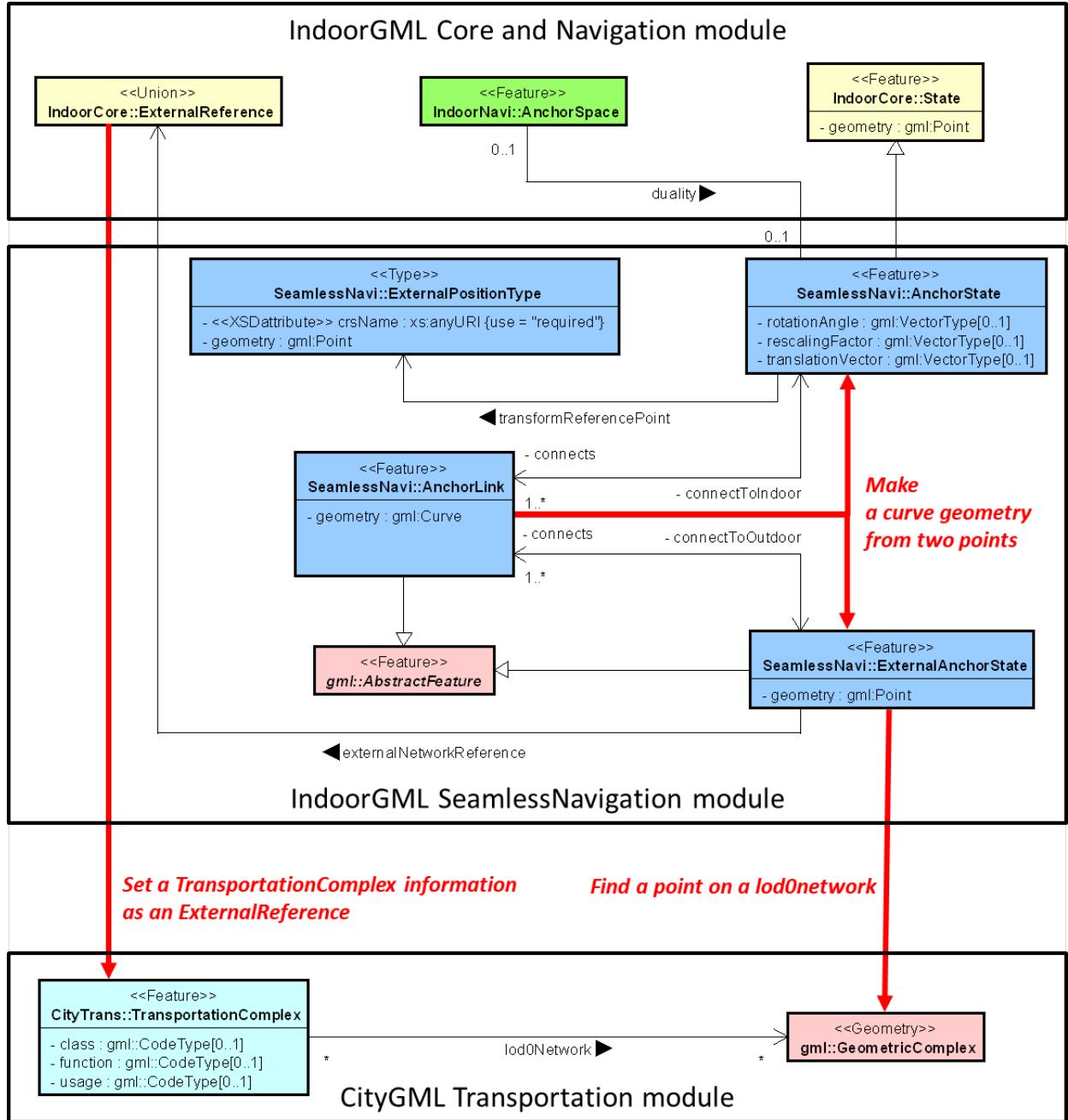
Figure 12 – UML diagram of transportation model in CityGML 2.0 (OGC 12-019)



The main class is `TransportationComplex`, which is composed of the parts `TrafficArea` and `AuxiliaryTrafficArea`. In the 'LOD 0' level of detail, the transportation complexes are modeled by line objects establishing a linear network. On this level, pathfinding algorithms or similar analyses can execute.

Therefore, the geometry of `ExternalAnchorState` should be created as a point on the `lod0Network` and must have the source data information of the `TransportationComplex` using the GML ID or URL information, as shown in Figure 13. The geometry of `AnchorLink` can be derived from two points, which are the geometries of `AnchorState` and `ExternalAnchorState`.

Figure 13 – The use case of the CityGML Transportation module

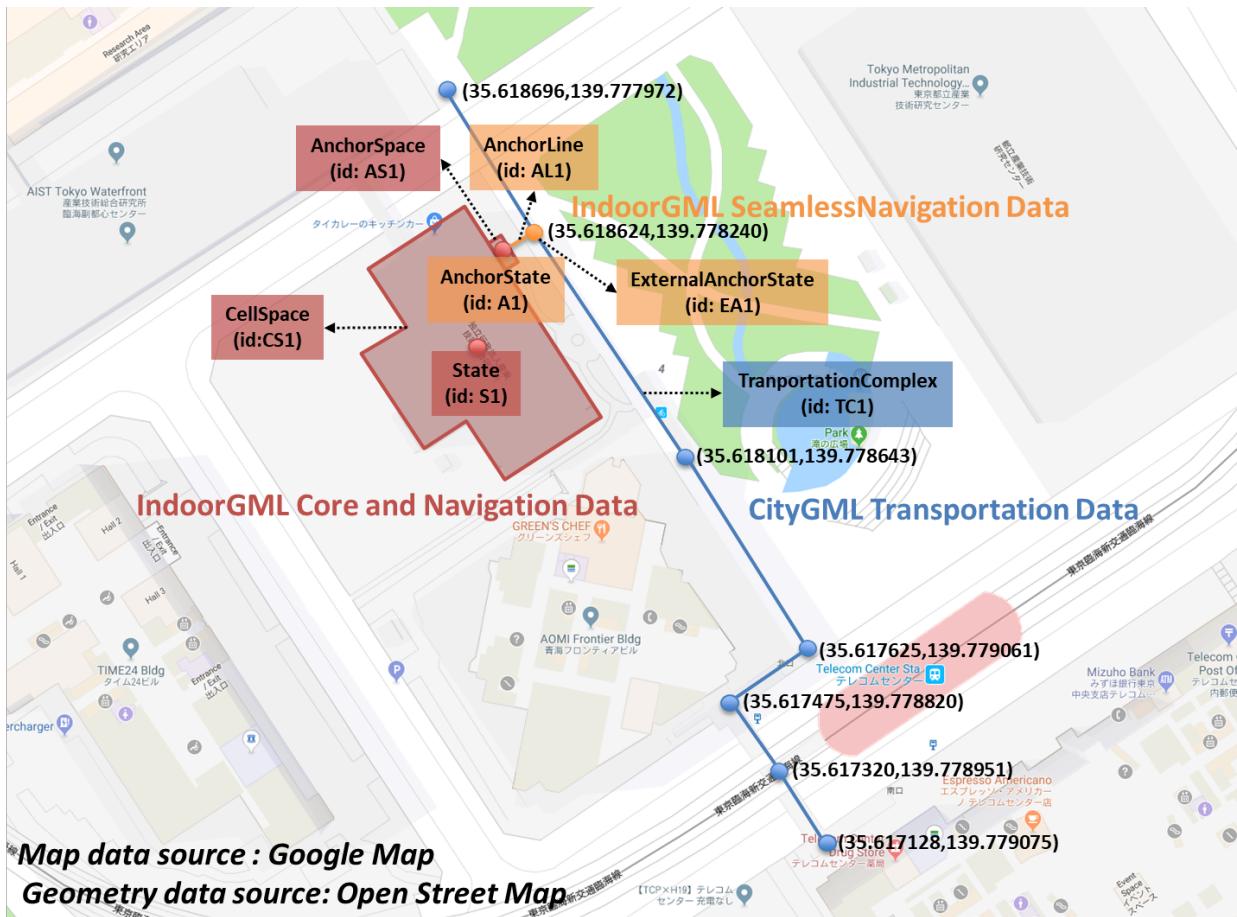


6.1.2. Use case site – AIST Tokyo Waterfront Annex building

The sample data for the use case have been conducted at a real site: AIST Tokyo Waterfront Annex in Japan, as shown in Figure 14. This sample data shows the basic structure of SeamlessNavigation module data and how the SeamlessNavigation module and CityGML Transportation model datasets are linked via external references.

For simplicity, the detailed structure inside the AIST building is not represented, and the data are constructed using only 2D geometry. All geometric data in the sample data are derived from the Open Street Map (OSM[3]) data and have the same CRS; EPSG:4326 (WGS 84). The IndoorGML data consists of two spaces: one *CellSpace* and one *AnchorSpace*. The CityGML data has only one *TransportationComplex* instance.

Figure 14 – CityGML use case site – AIST Tokyo Waterfront Annex



The detailed contents of sample data for the *AnchorState* class are as shown in Figure 15. This data consists of IndoorGML Core and Navigation modules. *AnchorState* can have a geometry derived from the *State* class. This geometry is used to create the geometry of an *AnchorLink*. It can have several elements for conversion. However, in this case, all geometries have the same CRS, so these elements are omitted. *AnchorState* must have a *transformReferencePoint* and *connects* element. The *TransformReferencePoint* must have CRS information, as shown in the yellow part of Figure 15. The *connects* element is represented by an xlink for the GML ID of the *AnchorLink* class instance, as shown in the green part of Figure 15 and Figure 18. *AnchorState* can have a duality association with the *AnchorSpace* class instance, as shown in the blue part of Figure 15.

Figure 15 – Part of sample data for *AnchorState* (*seamlessNaviSample.gml*)

```

<core:IndoorFeatures gml:id="IFs">
  ...
  <core:cellSpaceMember>
    <navi:AnchorSpace gml:id="AS1">
      <gml:name>Entrance</gml:name>
      <core:cellSpaceGeometry>
        <core:Geometry2D>
          <gml:Polygon srsName="EPSG:4326">
            <gml:exterior>
              <gml:LinearRing>
                <gml:posList> 35.6185243 139.7781466 35.61857293778 139.77810644557 35.61857882026
                139.77811733809 35.61852967618 139.77815783109 35.6185243 139.7781466 </gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </core:Geometry2D>
      </core:cellSpaceGeometry>
      <navi:class> 1000 </navi:class>
      <navi:function>1000</navi:function>
      <navi:usage>1000</navi:usage>
    </navi:AnchorSpace>
  </core:cellSpaceMember>
  ...
  <core:stateMember>
    <AnchorState gml:id="A1">
      <core:geometry>
        <gml:Point srsName="EPSG:4326">
          <gml:pos>35.61855174466 139.77813125399</gml:pos>
        </gml:Point>
      </core:geometry>
      <transformReferencePoint srsName="EPSG:4326">
        <geometry>
          <gml:Point srsName="EPSG:4326">
            <gml:pos>35.61855174466 139.77813125399</gml:pos>
          </gml:Point>
        </geometry>
      </transformReferencePoint>
      <duality xlink:href="#AS1"/>
      <connects xlink:href="#AL1"/>
    </AnchorState>
  </core:stateMember>
  ...
</core:IndoorFeatures>
```

Figure 16 shows the *ExternalAnchorState* sample data. It consists of three properties: *externalNetworkReference*, *geometry*, and *connects*. *ExternalNetworkReference* is a corresponding object in the *TransportationComplex* instance, as shown in the blue part of Figure 16 and Figure 17. The geometry of *ExternalAnchorState* is derived from one of the points on a *lod0network*, as shown in the yellow part of Figure 16 and Figure 17. *connects* is represented by *xlink* for GML id of *AnchorLink* class instance, as shown in the green part of Figure 16 and Figure 18.

Figure 16 – Part of sample data for *ExternalAnchorState* (seamlessNaviSample.gml)

```
<ExternalAnchorState gml:id="EA1">
  <externalNetworkReference>
    <core:informationSystem>cityTransSample.gml</core:informationSystem>
    <core:externalObject>
      <core:name>GMLID_TC1</core:name>
    </core:externalObject>
  </externalNetworkReference>
  <geometry>
    <gml:Point>
      <gml:pos>35.618624 139.778240</gml:pos>
    </gml:Point>
  </geometry>
  <connects xlink:href="#AL1"/>
</ExternalAnchorState>
```

Figure 17 – Part of sample data for *TransportationComplex* (cityTransSample.gml)

```
<cityObjectMember>
  <tran:TransportationComplex gml:id="TC1">
    <tran:lod0Network>
      <gml:CompositeCurve srsName="EPSG:4326">
        <gml:curveMember>
          <gml:LineString>
            <gml:pos>35.618696 139.777972</gml:pos>
            <gml:pos>35.618624 139.778240</gml:pos>
            <gml:pos>35.618101 139.778643</gml:pos>
            <gml:pos>35.617625 139.779061</gml:pos>
            <gml:pos>35.617475 139.778820</gml:pos>
            <gml:pos>35.617320 139.778951</gml:pos>
          </gml:LineString>
        </gml:curveMember>
      </gml:CompositeCurve>
    </tran:lod0Network>
  </tran:TransportationComplex>
</cityObjectMember>
```

AnchorLink sample data is shown in Figure 18. The association elements (*connectToIndoor* and *connectToOutdoor*) are represented by *xlinks* for GML ID of each class instance. The curve geometry is derived from the geometry of *connectToIndoor* and *connectToOutdoor* instances.

Figure 18 – Part of sample data for *AnchorLink* (*seamlessNaviSample.gml*)

```
<AnchorLink gml:id="AL1">
  <connectToIndoor xlink:href="#A1"/>
  <connectToOutdoor xlink:href="#EA1"/>
  <geometry>
    <gml:LineString srsName="EPSG:4326">
      <gml:posList>
        35.61855174466 139.77813125399
        35.618624 139.778240
      </gml:posList>
    </gml:LineString>
  </geometry>
</AnchorLink>
```

6.2. USE CASE WITH A SPECIFICATION FOR THE PEDESTRIAN NETWORK MODEL FROM JAPAN GOVERNMENT

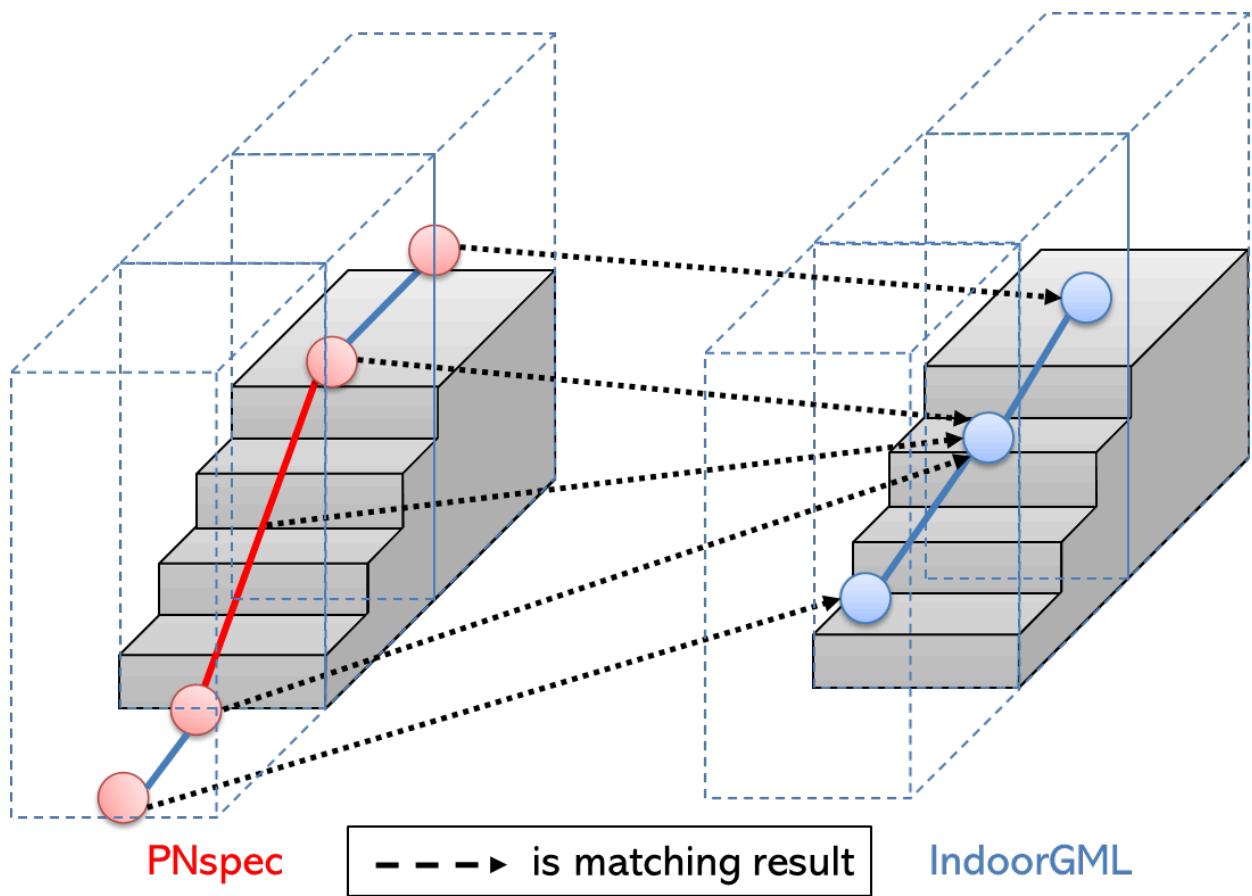
This section briefly introduces the “Development Specification for Spatial Network Model for Pedestrians (for short, PNspec) [2]” and suggests guidelines for conversion of the SeamlessNavigation module data using the specific cases.

6.2.1. Conversion method from PNspec to IndoorGML

PNspec includes both indoor and outdoor network information. To use the SeamlessNavigation module, we need to convert the indoors content of the PNspec into IndoorGML.

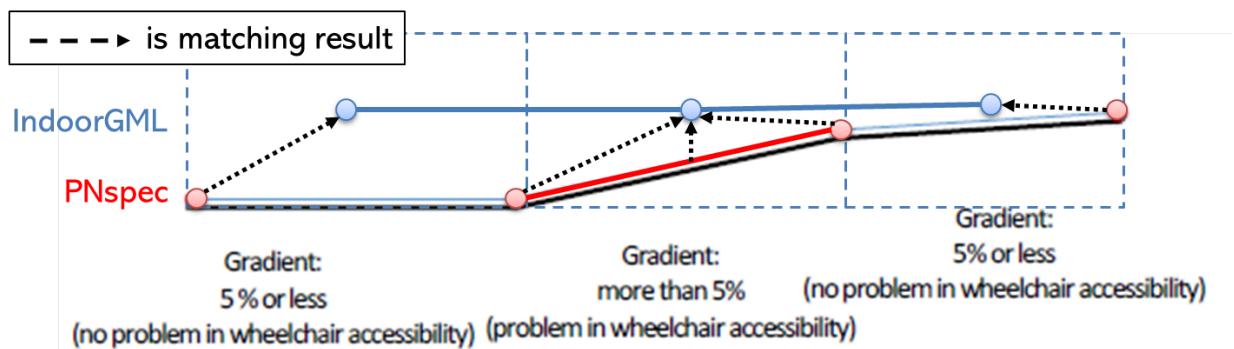
This chapter describes how to convert PNspec to IndoorGML. Because both IndoorGML and PNspec have node and link-based network models, they can easily convert between each schema. However, some special cases have different parts. In the case of a staircase, the PNspec places nodes at the beginning and end of the staircase, then links the two nodes. However, in IndoorGML, a staircase is expressed as a single space, so it is expressed as a single *State*. To resolve these conflicts, we need the mapping rules shown in Figure 19.

Figure 19 – An example of mapping PNspec to IndoorGML in a stairs case



Similarly, for gradients, PNspec places nodes at the beginning and end of the sloped space, then links the two nodes. However, in IndoorGML, the space with slope is expressed as a single space, so it is expressed as a single State. To resolve these conflicts, we need the mapping rules shown in Figure 20.

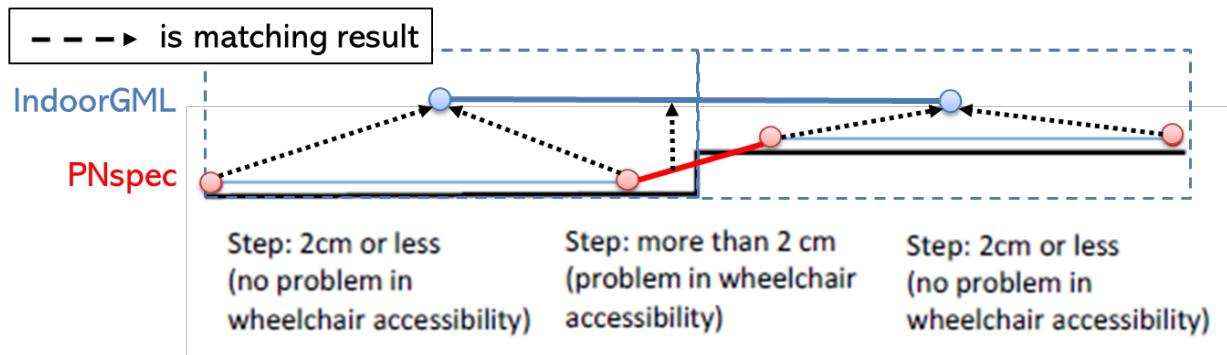
Figure 20 – An example of mapping PNspec to IndoorGML in changing points of barriers (gradient) case



Finally, in the case of a step, PNspec places nodes before and after a step. In the case of IndoorGML, we can divide space around a step. However, we do not create a state around the

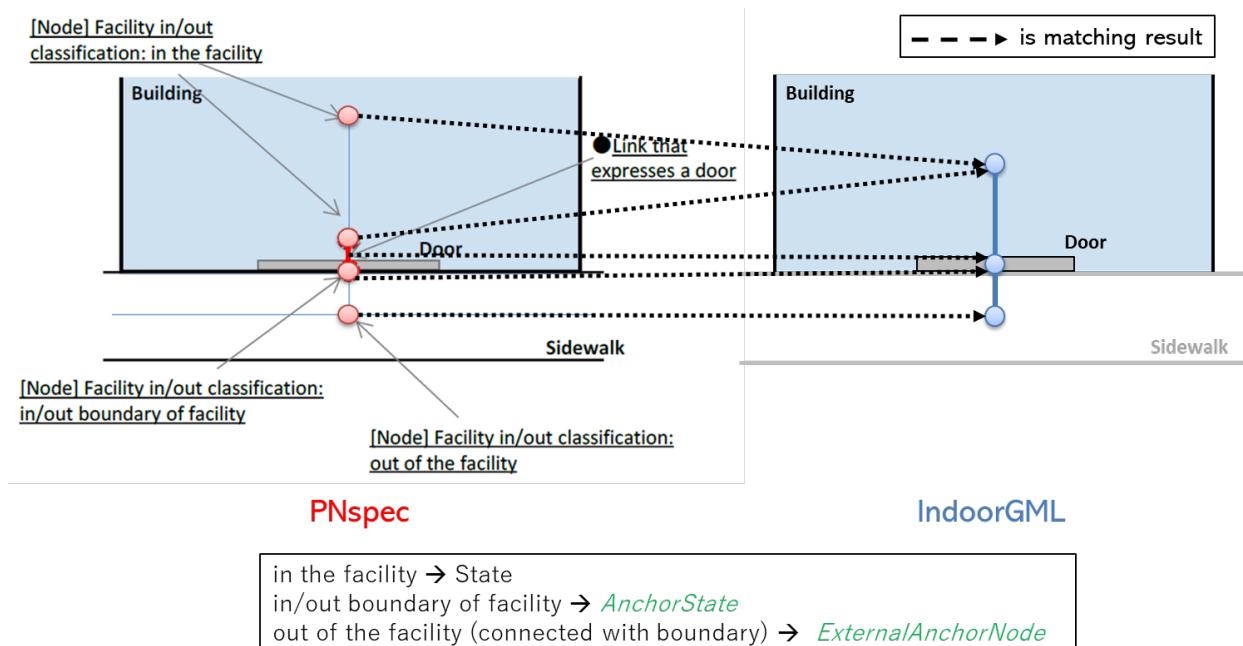
step. Depending on the concept of cellular space of IndoorGML, multiple nodes will be mapped to a single *State*, as shown in Figure 21.

Figure 21 – An example of mapping PNspec to IndoorGML in changing points of barriers (step) case



In addition, PNspec expresses a node having the same concept as an anchor node of this document as ‘the in/out boundary of the facility.’ However, in PNspec, an entrance is supposed to be a link. Depending on the characteristics of the *AnchorState* defined in this discussion paper, the entrance should be represented as a *State*. Therefore, the nodes and links representing the entrance in the PNspec should be represented by classes of IndoorGML Core and SeamlessNavigation module as shown in Figure 22.

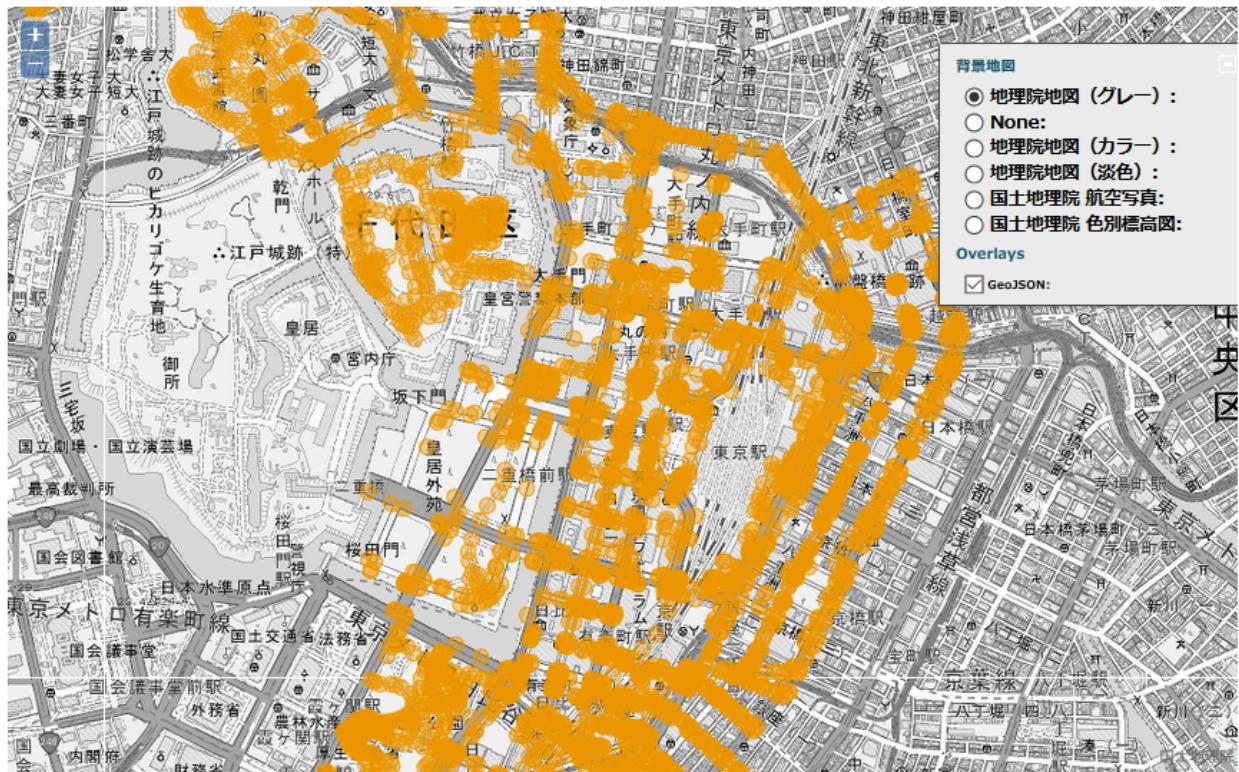
Figure 22 – An example of mapping PNspec to IndoorGML in entrance of building case



6.2.2. Use case site – Tokyo Station

The sample data for the use case have been conducted at a real site: Tokyo station, as shown in Figure 23. PNSpec sample data is derived from data by the Government of Japan, and is provided through an open license¹. This chapter shows how the SeamlessNavigation module and PNSpec datasets are linked via external references.

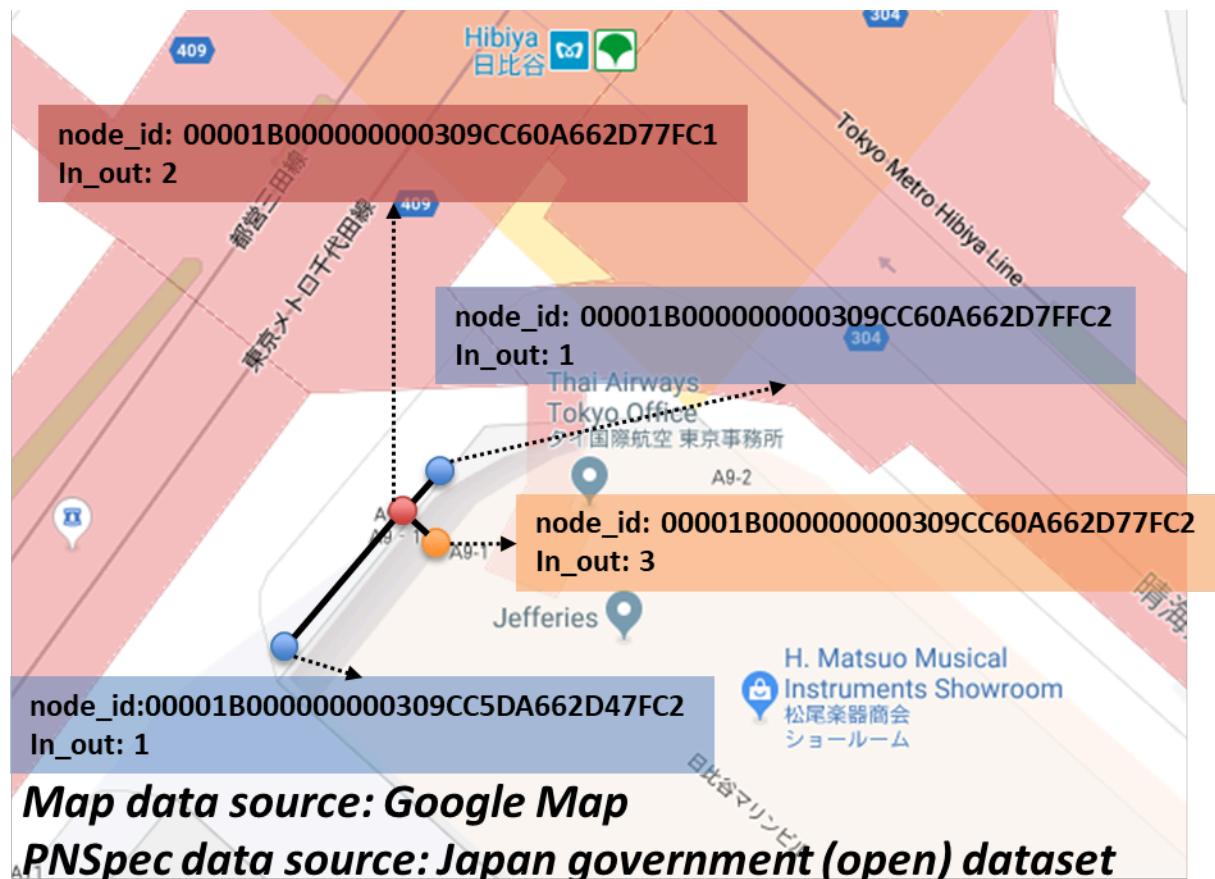
Figure 23 – PNSpec use case site – Tokyo station (Node case)



For simplicity, we used a part of Tokyo station data, as shown in Figure 24: four nodes and three edges.

¹<https://www.geospatial.jp/ckan/dataset/0401>

Figure 24 – Part of PNSpec sample data



The detailed contents of sample data for the nodes is shown in Figure 25. The PNSpec data provided by the GeoJSON format. Nodes can be distinguished by the ‘in_out’ attribute. In the case of an ‘in_out’ attribute of value ‘2’, this node represents the entrance of the building. And then, we can choose one node for connecting *ExternalAnchorState*, using the link attribute of the node. Figure 26 shows the sample data for edges that linked with the entrance of the building node with an ID of ‘00001B000000000309CC60A662D77FC1’. For making an *ExternalAnchorState* instance, we have to choose one of the connected nodes with the entrance of the building, and the ‘in_out’ value is ‘1’. In this sample data, there are three Nodes connected to the entrance of the building node. However, one of these nodes is located in indoor space: with an ‘in_out’ value of ‘2’. Therefore, we have to choose one node from the remaining two nodes. The mapped *ExternalAnchorState* result is shown in Figure 27.

```
{
  "type": "FeatureCollection",
  "name": "node",
  "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:EPSG::6668" } },
  "features": [
    { "type": "Feature", "properties": { "FID": 1603, "node_id": "00001B000000000309CC60A662D77FC1", "lat": 35.674696, "lon": 139.759514, "floor": "0", "in_out": 2, "link1_id": "00001B000000000309CC5F2662D5FFC1", "link2_id": "00001B000000000309CC60A662D77FC3", "link3_id": "00001B000000000309CC60A662D77FC4", "link4_id": " ", "link5_id": " ", "link6_id": " " } }
  ]
}
```

```

"link7_id": " ", "link8_id": " " }, "geometry": { "type": "Point", "coordinates": [ 139.759513964000121, 35.674695651000036 ] } },
{ "type": "Feature", "properties": { "FID": 1595, "node_id": "00001B000000000309CC5DA662D47FC2", "lat": 35.674545, "lon": 139.759354, "floor": "0", "in_out": 1, "link1_id": "00001B000000000309CC5D2662D3FFC2", "link2_id": "00001B000000000309CC5F2662D5FFC1", "link3_id": "00001B000000000309CC58A662DBFFC1", "link4_id": " ", "link5_id": " ", "link6_id": " ", "link7_id": " ", "link8_id": " " }, "geometry": { "type": "Point", "coordinates": [ 139.759354253000083, 35.674545111000043 ] } },
{ "type": "Feature", "properties": { "FID": 1607, "node_id": "00001B000000000309CC60A662D7FFC2", "lat": 35.674714, "lon": 139.759534, "floor": "0", "in_out": 1, "link1_id": "00001B000000000309CC60A662D77FC3", "link2_id": "00001B000000000309CC60A662D97FC2", "link3_id": "00001B000000000309CC62A662D4FFC2", "link4_id": " ", "link5_id": " ", "link6_id": " ", "link7_id": " ", "link8_id": " " }, "geometry": { "type": "Point", "coordinates": [ 139.759534400000121, 35.674714440000059 ] } },
{ "type": "Feature", "properties": { "FID": 1604, "node_id": "00001B000000000309CC60A662D77FC2", "lat": 35.674695, "lon": 139.759525, "floor": "0", "in_out": 3, "link1_id": "00001B000000000309CC60A662D77FC4", "link2_id": "00001B000000000309CC60A662D7FFC3", "link3_id": " ", "link4_id": " ", "link5_id": " ", "link6_id": " ", "link7_id": " ", "link8_id": " " }, "geometry": { "type": "Point", "coordinates": [ 139.759525074000067, 35.674694875000057 ] } }
]
}

```

Figure 25 – Part of sample data for PNSpec Node (PNSpec_node.json)

```

{
  "type": "FeatureCollection",
  "name": "link",
  "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:EPSG::6668" } },
  "features": [
    { "type": "Feature", "properties": { "FID": 2197, "link_id": "00001B000000000309CC5F2662D5FFC1", "start_id": "00001B000000000309CC5DA662D47FC2", "end_id": "00001B000000000309CC60A662D77FC1", "distance": "22.100000", "rt_struct": 1, "route_type": 1, "direction": 1, "width": 4, "vtcl_slope": 1, "lev_diff": 1, "tfc_signal": 1, "tfc_s_type": 1, "brail_tile": 2, "elevator": 1, "roof": 1 }, "geometry": { "type": "LineString", "coordinates": [ [ 139.759354253000083, 35.674545111000043 ], [ 139.759513964000121, 35.674695651000036 ] ] } },
    { "type": "Feature", "properties": { "FID": 2200, "link_id": "00001B000000000309CC60A662D77FC3", "start_id": "00001B000000000309CC60A662D77FC1", "end_id": "00001B000000000309CC60A662D7FFC2", "distance": "2.800000", "rt_struct": 1, "route_type": 1, "direction": 1, "width": 4, "vtcl_slope": 1, "lev_diff": 1, "tfc_signal": 1, "tfc_s_type": 1, "brail_tile": 2, "elevator": 1, "roof": 1 }, "geometry": { "type": "LineString", "coordinates": [ [ 139.759513964000121, 35.674695651000036 ], [ 139.75953440000121, 35.674714440000059 ] ] } },
    { "type": "Feature", "properties": { "FID": 4294, "link_id": "00001B000000000309CC60A662D77FC4", "start_id": "00001B000000000309CC60A662D77FC2", "end_id": "00001B000000000309CC60A662D77FC1", "distance": "1", "rt_struct": 7, "route_type": 6, "direction": 1, "width": 4, "vtcl_slope": 3, "lev_diff": 2, "tfc_signal": 1, "tfc_s_type": 1, "brail_tile": 1, "elevator": 1, "roof": 1 }, "geometry": { "type": "LineString", "coordinates": [ [ 139.759525074000067, 35.674694875000057 ], [ 139.759513964000121, 35.674695651000036 ] ] } },
  ]
}

```

Figure 26 – Part of sample data for PNSpec Edge (PNSpec_edge.json)

Figure 27 – Part of sample data for ExternalAnchorState

```
<ExternalAnchorState gml:id="EA1">
  <externalNetworkReference>
    <core:informationSystem>PNSpec_node.json</core:informationSystem>
    <core:externalObject>
      <core:name>node_id_00001B000000000309CC60A662D7FFC2</core:name>
    </core:externalObject>
  </externalNetworkReference>
  <geometry>
    <gml:Point>
      <gml:pos>139.75953440000121 35.674714440000059</gml:pos>
    </gml:Point>
  </geometry>
  ...
</ExternalAnchorState>
```

6.3. USE CASE WITH GDF 5.0

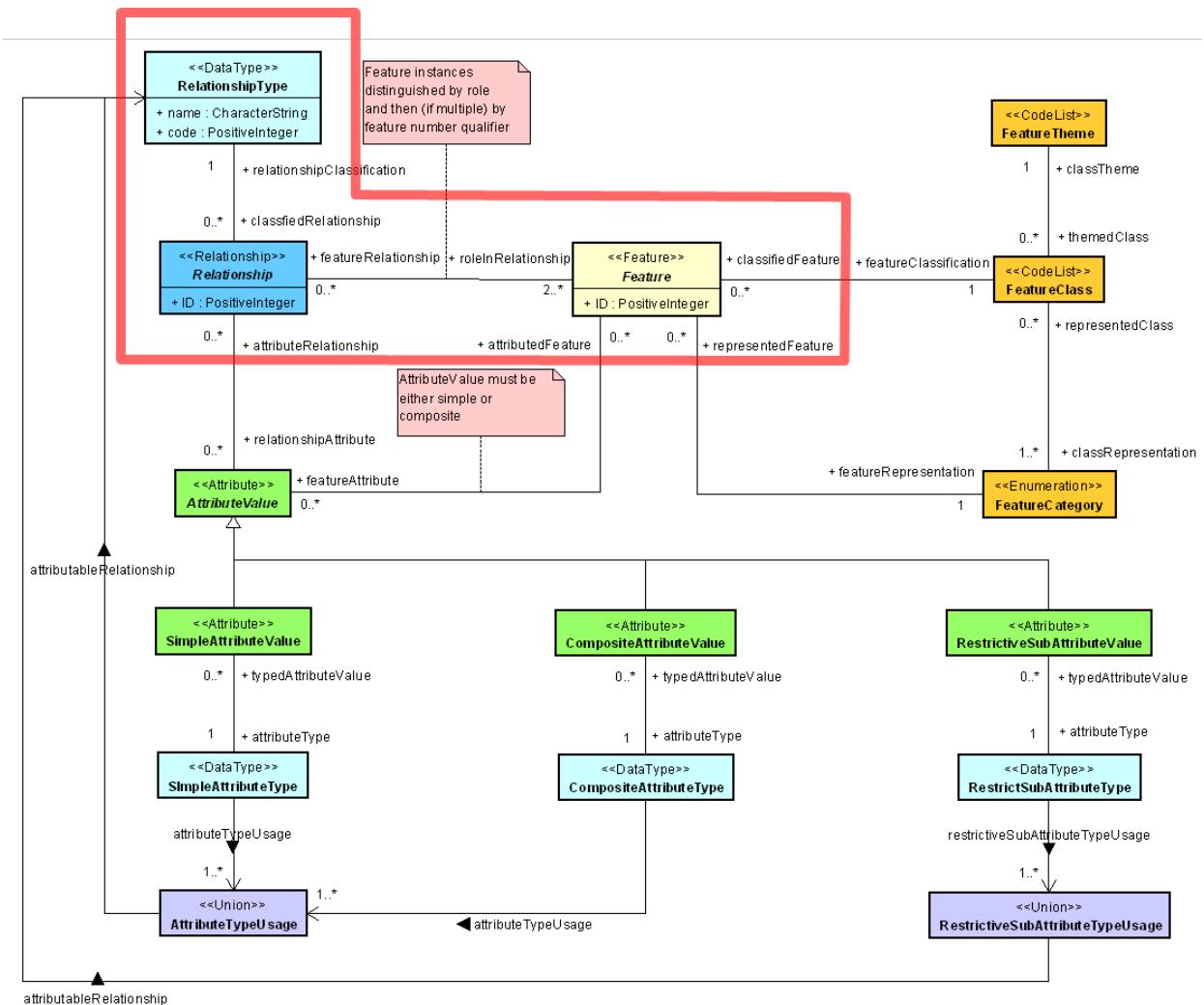
This section briefly introduces the Geographic Data Files (GDF) 5.0 format and suggests guidelines for the conversion of SeamlessNavigation module data using the specific cases.

6.3.1. GDF 5.0

GDF is an ISO international standard that specifies the conceptual and logical data models and physical encoding formats for geographic databases for Intelligent Transportation Systems (ITS) applications and services. It has the reference number ISO 14825:2011.

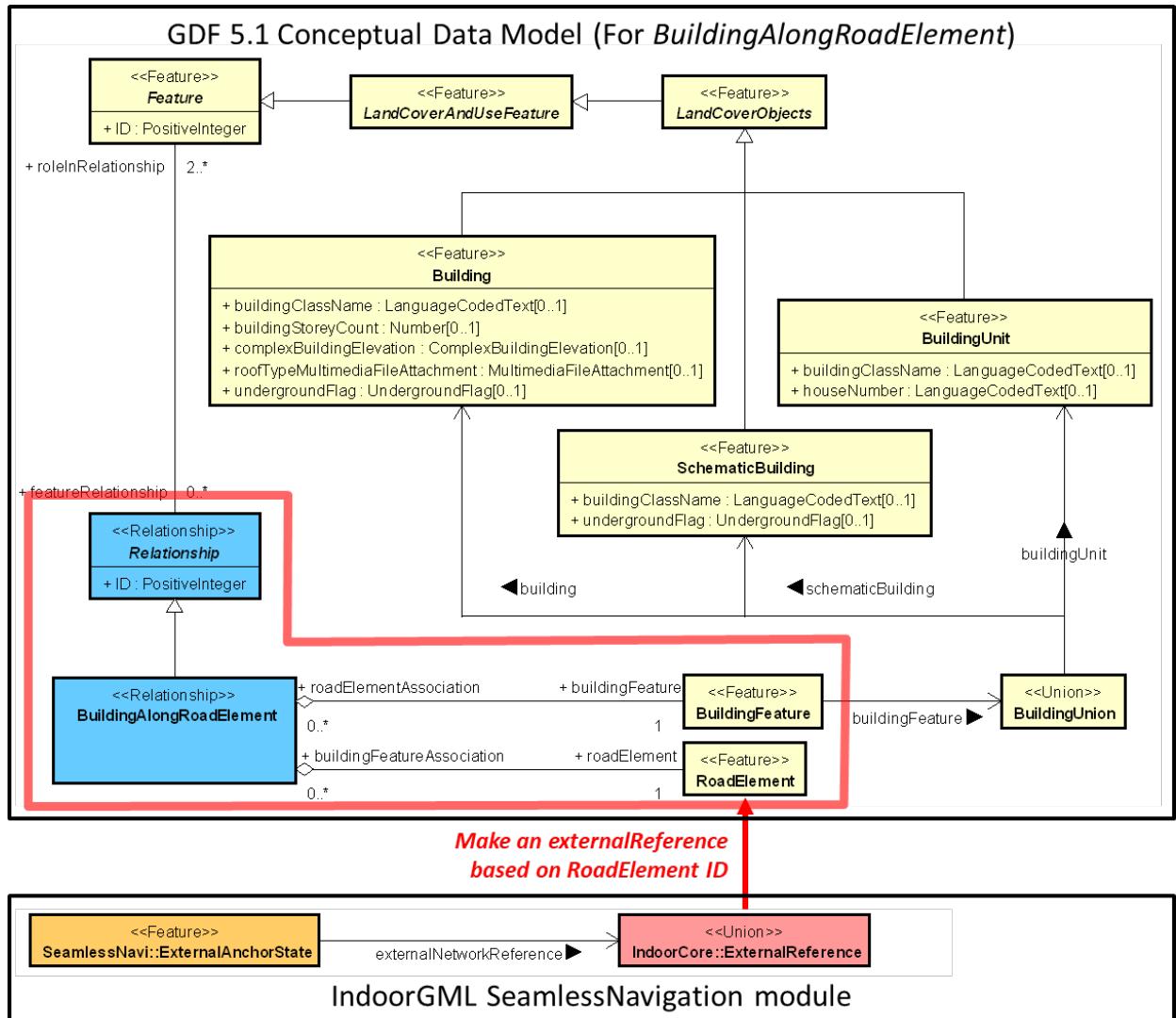
Figure 28 shows that the overall conceptual data model of GDF 5.0. Within the GDF 5.0 model, a *Feature* is a database representation of a real-world geographic object: roads, buildings, etc. Each *Feature* must belong to exactly one *FeatureClass* and *FeatureTheme*. A *Feature* may have zero or more *AttributeValue* instances that serve to represent characteristics of a *Feature*. A *Relationship* is used to associate two or more *Features* together and may have zero or more *AttributeValue* instances.

Figure 28 – UML diagram of the overall conceptual data model of GDF 5.0 (GDF 5.0, 2011)



In this discussion paper, we focused on how to connect the outdoor network and indoor network using the IndoorGML SeamlessNavigation module. Therefore, we have to make a connection between *ExternalAnchorState* and a specific feature of GDF 5.0; i.e., the entrance of the building, the element of pedestrian network, etc. Firstly, the entrance of the building element can be expressed in two ways: Using *Relationship*, Using *Feature*.

Figure 29 – UML diagram of the conceptual data model for ‘Relationships’ with ‘BuildingAlongRoadElement’



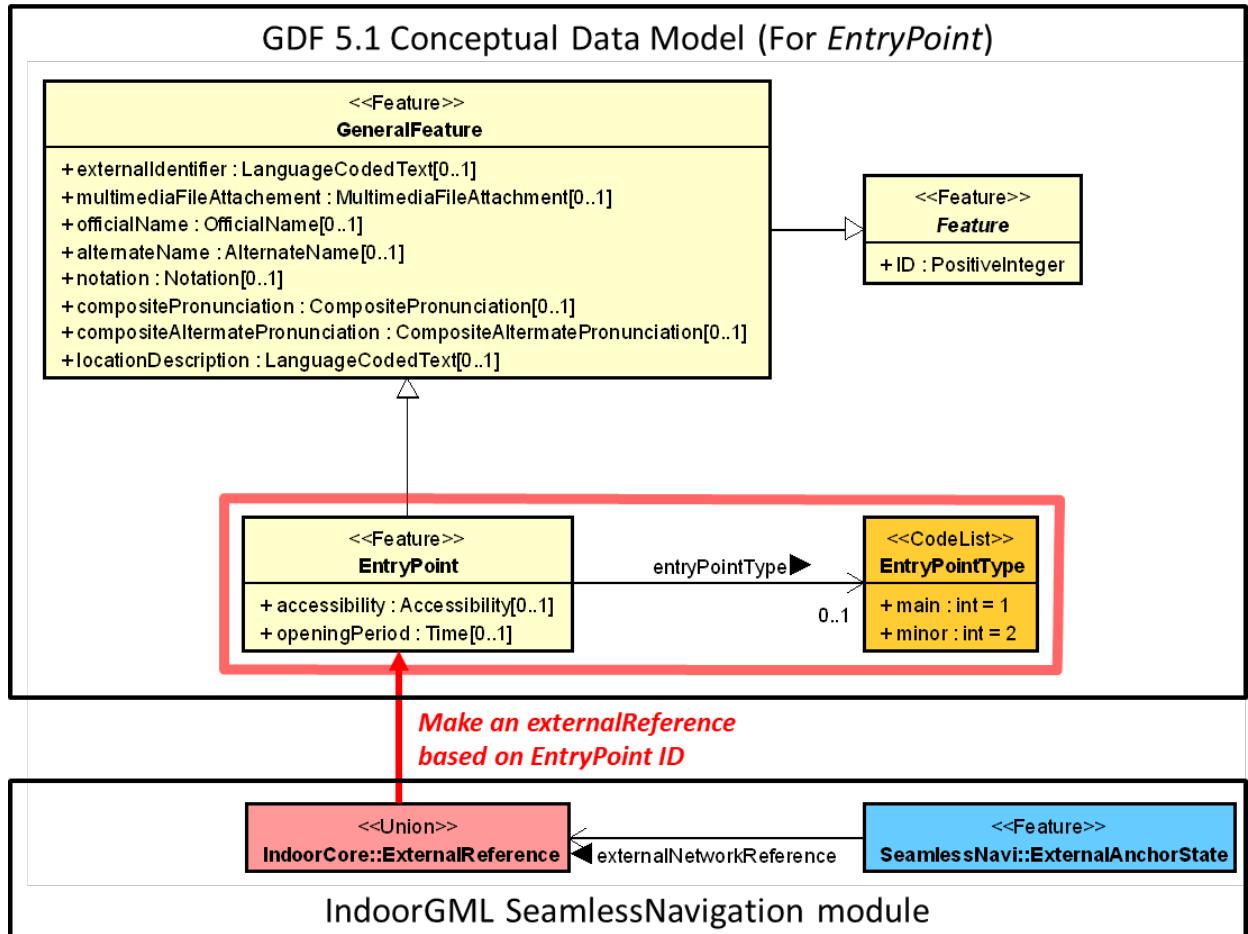
As shown in Figure 29, the entrance of the building can be expressed as a *BuildingAlongRoadElement*, one of the types of *Relationship*. *BuildingAlongRoadElement* identifies the *RoadElement* along which the entrance of the *Building*, *SchematicBuilding*, or *BuildingUnit* is situated. In the case of using *BuildingAlongRoadElement*, for connecting to *ExternalAnchorState*, we have to make *externalReference* based on *roadElement ID*, an element of *BuildingAlongRoadElement*.

As shown in Figure 30, the entrance of the building can be expressed as an *EntryPoint*, one of the *Types of GeneralFeature*. *EntryPoint* can be distinguished through the characteristics of the entrance. A “main” entrance is generally characterized by the following:

- It coincides with the address of the selected Service;
- It is provided with a reception/lobby for the visitor;

- It is the entrance which attracts the most attention;
- It is the entrance to which road signs (if present) point. And at least one of the *EntryPoint* instances of a service shall be attributed as “Main.”

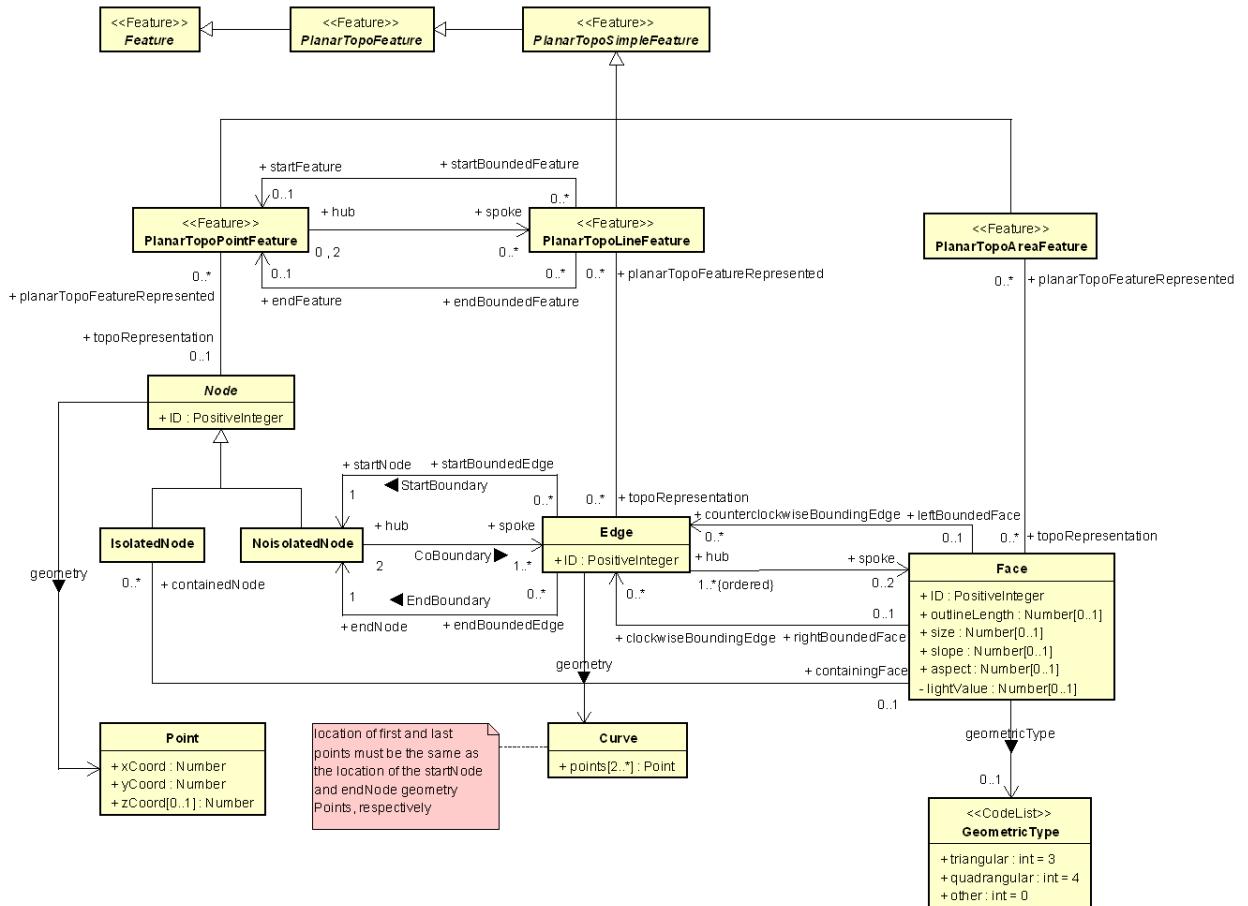
Figure 30 – UML diagram of the conceptual data model for *EntryPoint* of General Feature



In the case of using *EntryPoint*, for connecting to *ExternalAnchorState*, we have to make an *externalReference* based on the *EntryPoint* ID.

The geometry of all *Features* in GDF 5.0 shall be expressed by *Node*, *Edge*, and *Face*. Figure 31 shows a UML diagram of the conceptual data model for *PlanarTopoSimpleFeature*, one of the types of graph topology.

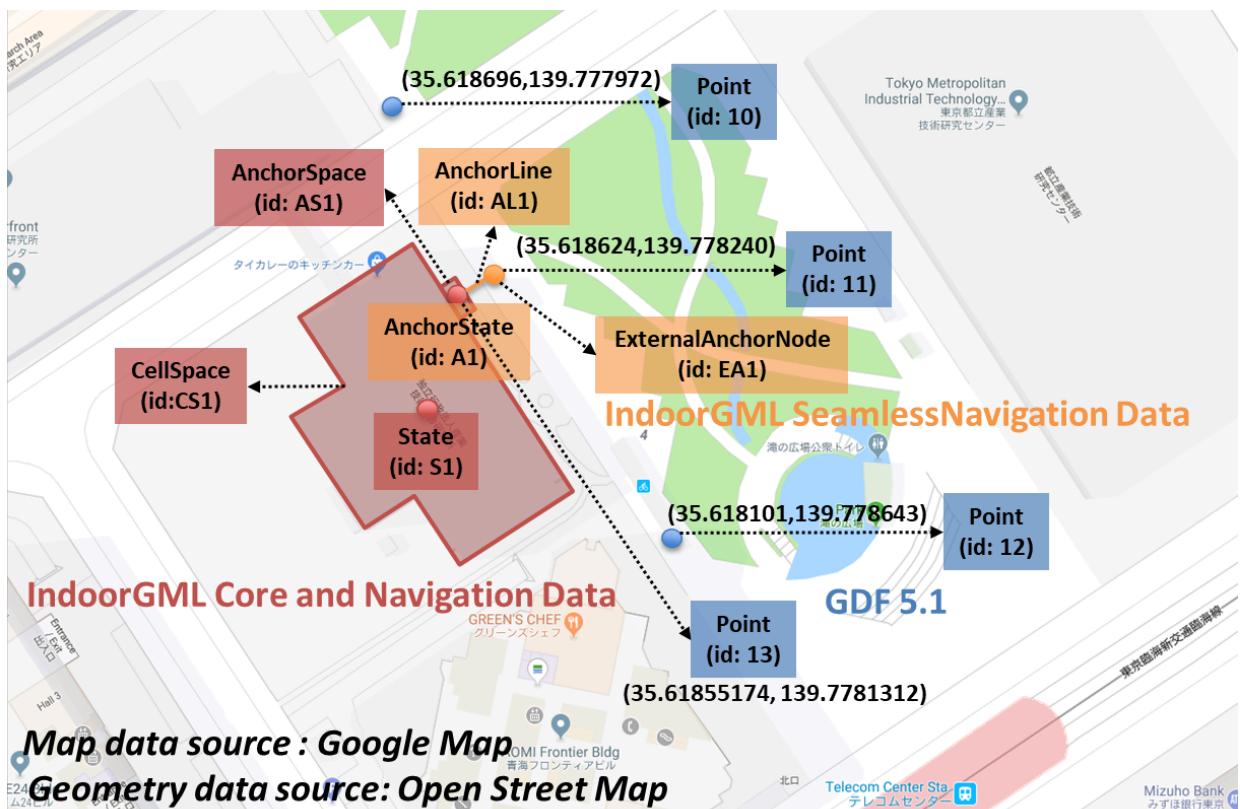
Figure 31 – UML diagram of the conceptual data model for ‘PlanarTopoSimpleFeature’



6.3.2. Use case site – AIST Tokyo Waterfront Annex building

The sample data for the use case have been conducted at a real site: AIST Tokyo Waterfront Annex in Japan, as shown in Figure 32. The GDF 5.0 sample data created was based on the XML schema that is provided in Chapter 13 of GDF 5.0. For simplicity, we elide DLS (*Dataset*, *Layer*, *Section*) information in sample data. The detailed contents of the GDF sample data are shown in Figure 33. This data consists of four *Point Features* and one *Relationship*.

Figure 32 – GDF 5.0 Use case site – AIST Tokyo Waterfront Annex



Point Feature instances must have point_feat_ID, feature_code, and coord properties: point_feat_ID means identifier of *Point Feature*, feature_code means the four-digit code of the Feature_Class to which the Feature in issue belongs, and coord means a position of Feature.

Relationship must have `rel_ID` and `rel_code`: `rel_ID` means identifier of *Relationship* and `rel_code` means (pre-defined or user-defined) relationship type code. And *Relationship* can have `num_feat` and `rel_feature`: `num_feat` means the number of features that belong to *Relationship* and `rel_feature` means the feature information belong to *Relationship*.

The mapped *ExternalAnchorState* result is shown in Figure 34.

Figure 33 – GDF 5.0 sample data (GDF_5_0_sample.xml)

```
<?xml version='1.0' encoding='UTF-8'?>
<gdf xmlns="http://www.udus.org"
      xmlns:core="http://www.opengis.net/2003/01/core"
      xmlns:gml="http://www.opengis.net/2003/01/gml"
      xmlns:feature="http://www.udus.org/2003/01/feature">
  <point_feature>
    <non-explicit_point_feat>
      <point_feat_ID>10</point_feat_ID>
      <feature_code>
        <feat_code_D>
          <value>n4110</value>
          <name>noRoad Element</name>
        </feat_code_D>
      </feature_code>
      <coord>
        <x>coord=1397779720</x>
        <y>coord=351869600</y>
      </coord>
    </non-explicit_point_feat>
  </point_feature>
  <point_feature>
    <non-explicit_point_feat>
      <point_feat_ID>11</point_feat_ID>
      <feature_code>
        <feat_code_D>
          <value>n4110</value>
          <name>noRoad Element</name>
        </feat_code_D>
      </feature_code>
      <coord>
        <x>coord=1397782400</x>
        <y>coord=351869400</y>
      </coord>
    </non-explicit_point_feat>
  </point_feature>
  <point_feature>
    <non-explicit_point_feat>
      <point_feat_ID>12</point_feat_ID>
      <feature_code>
        <feat_code_D>
          <value>n4110</value>
          <name>noRoad Element</name>
        </feat_code_D>
      </feature_code>
      <coord>
        <x>coord=1397784300</x>
        <y>coord=3518131000</y>
      </coord>
    </non-explicit_point_feat>
  </point_feature>
  <point_feature>
    <non-explicit_point_feat>
      <point_feat_ID>13</point_feat_ID>
      <feature_code>
        <feat_code_D>
          <value>n4110</value>
          <name>noBuilding</name>
        </feat_code_D>
      </feature_code>
      <coord>
        <x>coord=1397786430</x>
        <y>coord=3518131000</y>
      </coord>
    </non-explicit_point_feat>
  </point_feature>
  <relationship>
    <rel_ID>1000</rel_ID>
    <rel_code>
      <ref>1000</ref>
      <value>n1000</value>
      <name>noBuilding</name>
    </rel_code>
    <rel_feature>
      <feature_id>11</feature_id>
      <feature_name>FEATURE_ID_11</feature_name>
      <feature_code>
        <feat_code_D>
          <value>n4110</value>
          <name>noBuilding</name>
        </feat_code_D>
      </feature_code>
      <feature_cat>
        <category>
          <value>n1</value>
          <name>noBuilding</name>
        </category>
      </feature_cat>
      <feature_role>
        <role>feature_role_number</role>
        <value>feature_role_number1</value>
        <feature_role_number>1</feature_role_number>
      </feature_role>
      <rel_feature>
        <rel_feature>
          <feature_id>13</feature_id>
          <feature_name>FEATURE_ID_13</feature_name>
          <feature_code>
            <feat_code_D>
              <value>n4110</value>
              <name>noBuilding</name>
            </feat_code_D>
          </feature_code>
          <feature_cat>
            <category>
              <value>n1</value>
              <name>noBuilding</name>
            </category>
          </feature_cat>
          <feature_role>
            <role>feature_role_number</role>
            <value>feature_role_number2</value>
            <feature_role_number>2</feature_role_number>
          </feature_role>
        </rel_feature>
      </rel_feature>
    </rel_feature>
  </relationship>
</gdf>
```

Figure 34 – Part of sample data for ‘ExternalAnchorState’

```
<ExternalAnchorState gml:id="EA1">
  <externalNetworkReference>
    <core:informationSystem>GDF_5_0_sample.gml</core:informationSystem>
    <core:externalObject>
      <core:name>FEATURE_ID_11</core:name>
    </core:externalObject>
  </externalNetworkReference>
  <geometry>
    <gml:Point>
      <gml:pos>35.618624 139.778240</gml:pos>
    </gml:Point>
  </geometry>
  ...
</ExternalAnchorState>
```



A

ANNEX A (NORMATIVE) XML SCHEMA FOR INDOORGML SEAMLESS NAVIGATION MODULE

ANNEX A (NORMATIVE)

XML SCHEMA FOR INDOORGML SEAMLESSNAVIGATION MODULE

This annex provides the normative XML schema document for the SeamlessNavigation Module.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns="http://www.opengis.net/indoorgml/1.0/seamlessNavi"
  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"
  xmlns:IndoorNavi="http://www.opengis.net/indoorgml/1.0/navigation"
  targetNamespace="http://www.opengis.net/indoorgml/1.0/seamlessNavi"
  version="1.0"
  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/indoorgmlcore.xsd"/>
  <xs:import namespace="http://www.opengis.net/indoorgml/1.0/navigation"
    schemaLocation="http://schemas.opengis.net/indoorgmlnavi.xsd"/>
  <!-- ===== -->
  <xs:element name="AnchorState" type="AnchorStateType" substitutionGroup="IndoorCore:State">
    <xs:annotation>
      <xs:documentation>AnchorState
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <!-- ===== -->
  <xs:complexType name="AnchorStateType">
    <xs:complexContent>
      <xs:extension base="IndoorCore:StateType">
        <xs:sequence>
          <xs:element name="transformReferencePoint" type="ExternalPositionType"/>
          <xs:element name="rotationAngle" type="gml:VectorType" minOccurs="0"/>
          <xs:element name="rescalingFactor" type="gml:VectorType" minOccurs="0"/>
          <xs:element name="translationVector" type="gml:VectorType" minOccurs="0"/>
          <xs:element name="duality" type="AnchorSpacePropertyType" minOccurs="0"/>
          <xs:element name="connects" type="AnchorLinkPropertyType" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

```

</xs:complexType>
<!-- ===== -->
<xs:complexType name="AnchorState.PropertyType">
<xs:sequence minOccurs="0">
<xs:element ref="AnchorState"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!-- ===== -->
<xs:complexType name="AnchorSpace.PropertyType">
<xs:sequence minOccurs="0">
<xs:element ref="IndoorNavi:AnchorSpace"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!-- ===== -->
<xs:complexType name="ExternalPositionType">
<xs:sequence>
<xs:element name="geometry" type="gml:PointPropertyType"/>
</xs:sequence>
<xs:attribute name="srsName" type="xs:anyURI" use="required"/>
</xs:complexType>
<!-- ===== -->
<xs:element name="AnchorLink" type="AnchorLinkType" substitutionGroup="gml:AbstractFeature">
<xs:annotation>
<xs:documentation>AnchorLink</xs:documentation>
</xs:annotation>
</xs:element>
<!-- ===== -->
<xs:complexType name="AnchorLinkType">
<xs:complexContent>
<xs:extension base="gml:AbstractFeatureType">
<xs:sequence>
<xs:element name="connectToIndoor" type="AnchorState.PropertyType"/>
<xs:element name="connectToOutdoor" type="ExternalAnchorState.PropertyType"/>
<xs:element name="geometry" type="gml:CurvePropertyType"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!-- ===== -->
<xs:complexType name="AnchorLink.PropertyType">
<xs:sequence minOccurs="0">
<xs:element ref="AnchorLink"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!-- ===== -->
<xs:element name="ExternalAnchorState" type="ExternalAnchorStateType" substitutionGroup="gml:AbstractFeature">
<xs:annotation>
<xs:documentation>ExternalAnchorState</xs:documentation>

```

```
</xs:documentation>
</xs:annotation>
</xs:element>
<!-- ===== -->
<xs:complexType name="ExternalAnchorStateType">
<xs:complexContent>
<xs:extension base="gml:AbstractFeatureType">
<xs:sequence>
<xs:element name="externalNetworkReference" type="IndoorCore:ExternalReferenceType"/>
<xs:element name="geometry" type="gml:PointPropertyType"/>
<xs:element name="connects" type="AnchorLinkPropertyType" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<!-- ===== -->
<xs:complexType name="ExternalAnchorState.PropertyType">
<xs:sequence minOccurs="0">
<xs:element ref="ExternalAnchorState"/>
</xs:sequence>
<xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<!-- ===== -->
</xs:schema>
```



B

ANNEX B (NORMATIVE) SEAMLESS NAVIGATION MODULE EXAMPLE DOCUMENT

ANNEX B (NORMATIVE)

SEAMLESS NAVIGATION MODULE EXAMPLE DOCUMENT

The following are examples of the SeamlessNavigation module documents generated as described in Clause 6.2

```
<CityModel xmlns="http://www.opengis.net/citygml/2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:tran="http://www.opengis.net/citygml/transportation/2.0"
  xsi:schemaLocation="http://www.opengis.net/citygml/2.0 http://schemas.opengis.net/citygml/
  profiles/base/2.0/CityGML.xsd">
  <cityObjectMember>
    <tran:TransportationComplex gml:id="TC1">
      <tran:lod0Network>
        <gml:CompositeCurve srsName="EPSG:4326">
          <gml:curveMember>
            <gml:LineString>
              <gml:pos>35.618696 139.777972</gml:pos>
              <gml:pos>35.618624 139.778240</gml:pos>
              <gml:pos>35.618101 139.778643</gml:pos>
              <gml:pos>35.617625 139.779061</gml:pos>
              <gml:pos>35.617475 139.778820</gml:pos>
              <gml:pos>35.617320 139.778951</gml:pos>
            </gml:LineString>
          </gml:curveMember>
        </gml:CompositeCurve>
      </tran:lod0Network>
    </tran:TransportationComplex>
  </cityObjectMember>
</CityModel>
```

cityTransSample.gml

```
<gml:FeatureCollection
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns="http://www.opengis.net/indoorgml/1.0/seamlessNavi"
  xmlns:core="http://www.opengis.net/indoorgml/1.0/core"
  xmlns:navi="http://www.opengis.net/indoorgml/1.0/navigation"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/indoorgml/1.0/core http://schemas.opengis.net/
  indoorgml/1.0/indoorgmlcore.xsd"
```

```

http://www.opengis.net/indoorgml/1.0/navigation http://schemas.opengis.net/indoorgml/1.0/
indoorgmlnavi.xsd
http://www.opengis.net/indoorgml/1.0/seamlessNavi seamlessnavi.xsd">
<gml:featureMembers>
<core:IndoorFeatures gml:id="IFs">
<core:primalSpaceFeatures>
<core:PrimalSpaceFeatures gml:id="PSs">
<core:cellSpaceMember>
<core:CellSpace gml:id="CS1">
<gml:name>Main Hall</gml:name>
<core:cellSpaceGeometry>
<core:Geometry2D>
<gml:Polygon srsName="EPSG:4326">
<gml:exterior>
<gml:LinearRing>
<gml:posList> 35.6181996 139.7784147 35.6186619 139.778033 35.61851 139.7777545 35.6183772
139.7778641 35.6183119 139.7777444 35.618092 139.777926 35.6181573 139.7780457 35.6180476
139.7781362 35.6180561 139.7781516 35.6180339 139.77817 35.6181648 139.77841 35.618187 139.
7783916 35.6181996 139.7784147 </gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</core:Geometry2D>
</core:cellSpaceGeometry>
</core:CellSpace>
</core:cellSpaceMember>
<core:cellSpaceMember>
<navi:AnchorSpace gml:id="AS1">
<gml:name>Entrance</gml:name>
<core:cellSpaceGeometry>
<core:Geometry2D>
<gml:Polygon srsName="EPSG:4326">
<gml:exterior>
<gml:LinearRing>
<gml:posList>
35.6185243 139.7781466 35.61857293778 139.77810644557 35.61857882026 139.77811733809 35.
61852967618 139.77815783109 35.6185243 139.7781466
</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</core:Geometry2D>
</core:cellSpaceGeometry>
<navi:class> 1000 </navi:class>
<navi:function> 1000 </navi:function>
<navi:usage> 1000 </navi:usage>
</navi:AnchorSpace>
</core:cellSpaceMember>
</core:PrimalSpaceFeatures>
</core:primalSpaceFeatures>
<core:multiLayeredGraph>
<core:MultiLayeredGraph gml:id="MLG1">
<core:spaceLayers>
<core:spaceLayerMember>

```

```

<core:SpaceLayer gml:id="SL1">
<core:nodes>
<core:stateMember>
<core:State gml:id="S1">
<core:duality xlink:href="#CS1"/>
<core:geometry>
<gml:Point>
<gml:pos> 35.61833152465 139.77806794632 </gml:pos>
</gml:Point>
</core:geometry>
</core:State>
</core:stateMember>
<core:stateMember>
<AnchorState gml:id="A1">
<core:geometry>
<gml:Point srsName="EPSG:4326">
<gml:pos> 35.61855174466 139.77813125399 </gml:pos>
</gml:Point>
</core:geometry>
<transformReferencePoint srsName="EPSG:4326">
<geometry>
<gml:Point srsName="EPSG:4326">
<gml:pos> 35.61855174466 139.77813125399 </gml:pos>
</gml:Point>
</geometry>
</transformReferencePoint>
<duality xlink:href="#AS1"/>
<connects xlink:href="#AL1"/>

</AnchorState>
</core:stateMember>
</core:nodes>
</core:SpaceLayer>
</core:spaceLayerMember>
</core:spaceLayers>
</core:MultiLayeredGraph>
</core:multiLayeredGraph>
</core:IndoorFeatures>
<ExternalAnchorState gml:id="EA1">
<externalNetworkReference>
<core:informationSystem>cityTransSample.gml</core:informationSystem>
<core:externalObject>
<core:name>GMLID_T1</core:name>
</core:externalObject>
</externalNetworkReference>
<geometry>
<gml:Point>
<gml:pos>35.618624 139.778240</gml:pos>
</gml:Point>
</geometry>
<connects xlink:href="#AL1"/>
</ExternalAnchorState>
<AnchorLink gml:id="AL1">
<connectToIndoor xlink:href="#A1"/>

```

```
<connectToOutdoor xlink:href="#EA1"/>
<geometry>
<gml:LineString srsName="EPSG:4326">
<gml:posList> 35.61855174466 139.77813125399 35.618624 139.778240 </gml:posList>
</gml:LineString>
</geometry>
</AnchorLink>
</gml:featureMembers>
</gml:FeatureCollection>
```

seamlessNaviSample.gml



C

ANNEX C (NORMATIVE) REVISION HISTORY

C

ANNEX C (NORMATIVE) REVISION HISTORY

| DATE | RELEASE | EDITOR | PRIMARY CLAUSES MODIFIED | DESCRIPTION |
|------------|---------|---|--------------------------|-----------------|
| 2018-11-01 | 0.1 | Kyong-Sook Kim all Teahoon Kim Jiyeong Lee | | Initial version |
| 2018-12-03 | 0.5 | Kyong-Sook Kim 5, 6, Annex B Teahoon Kim Jiyeong Lee | | complemented |
| 2019-02-04 | 1.0 | Kyong-Sook Kim all Teahoon Kim Jiyeong Lee Hye-Young Kang | | final version |



D

ANNEX D (INFORMATIVE)

D

ANNEX D (INFORMATIVE)

D

BIBLIOGRAPHY
