How to Extend IndoorGML for Seamless Navigation Between Indoor and Outdoor Space

Taehoon Kim[1], Kyoung-Sook Kim[1(✉)], and Jiyeong Lee[2]

[1] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
{kim.taehoon,ks.kim}@aist.go.jp
[2] The University of Seoul, Seoul, Republic of Korea
jlee@uos.ac.kr

Abstract. Navigation and tracking systems became a traditional tool of positioning determination and route guidance for moving objects such as vehicles and pedestrians. Position techniques like Global Positioning Systems (GPS) and map information about navigation space are fundamental elements in these systems. As matching a position into the map, a route plan is determined by navigational knowledge of space constraints. In this paper, we propose an extension model of Open Geospatial Consortium (OGC) IndoorGML for representing a seamless navigation space, called Seamless Navigation Model (SNM). In particular, we link the existing spatial models for seamless navigation between indoor and outdoor environments for not only humans but also robots instead of developing a unified model of indoor and outdoor spaces. This study presents how to realize the concept of Anchor node in IndoorGML and define characteristics to connect different geospatial models of indoor and outdoor spaces. Finally, we provide a use-case of SNM with a road network represented by OGC CityGML, which is a standard format for the storage and exchange of virtual three-dimensional city models. Our model can reduce the integration cost of different geospatial models and encourage the reuse of the current map information for various tasks not only seamless navigation.

Keywords: Anchor node · Seamless navigation space · OGC IndoorGML · OGC CityGML

1 Introduction

With the advances in positioning and wireless communication technologies, navigation systems have become one of the most widely used types of location-based services in not only outdoor but also indoor space. An accurate positioning and map information are essential for the route guidance of objects moving from one

Y. Kawai et al. (Eds.): W2GIS 2019, LNCS 11474, pp. 46–62, 2019.
https://doi.org/10.1007/978-3-030-17246-6_5

place to another. As matching a position into the map, a route plan is determined by navigational knowledge of space constraints. Comparing to a long history of outdoor navigation, indoor navigation has recently witnessed an increase in interest with new positioning systems (e.g., WiFi, Bluetooth, and inertial navigation system) and indoor maps (e.g., Google Indoor[1], OpenStreetMap (OSM) Indoor[2], and Apple IMDF[3]). However, these indoor maps are based on two-dimensional floor plans for the position mapping like outdoor maps and bring issues of interoperability because of their different spatial models.

There are three major standards that provide spatial data models and formats for the representation and exchange of indoor space: IFC [3] of buildingSMART, OGC CityGML [11] and OGC IndoorGML [12]. In particular, these standards focus on three-dimensional (3D) geometric information of buildings and defines semantic features such as rooms, walls, roofs, and corridors [4]. However, IFC and OGC CityGML are suitable for general purpose in a wide range of indoor applications, but they lack for the purpose of indoor navigation, especially autonomous navigation of mobile robots, due to the absence of topological constructs. In other words, they need additional cost to compute a path from one position to another and guide a route. OGC IndoorGML defines a framework of indoor spatial information of building components and a network model for their topological relationships, e.g., adjacency and connectivity, for indoor navigation. However, it has a limitation to integrate navigation in indoor and outdoor environments. Still, most navigation systems are focused on only outdoor or indoor environment.

In this paper, we consider a spatial model to integrate the indoor and outdoor navigation spaces for seamless path planning. There are several studies for making unified models for indoor-outdoor seamless navigation [8,13,15,16]. Wang and Niu [16] proposed a data model based on the OSM basic data structure for mapping the connections between inner building parts; e.g., rooms, doors, stairs, elevators, etc. Also, they provide a routing algorithm to calculate the indoor-outdoor route using the proposed data model. However, these unified models have a common disadvantage: the cost to convert the existing models into a unified model. Instead of the unified model, we propose a reference model as an extension of IndoorGML to combine the indoor and outdoor features for seamless navigation in this study. In particular, we take into account the interoperability and compatibility of standardized data models and formats. This paper consists of four parts as follow: In Sect. 2, we describe the existing studies about indoorGML based navigation models and spatial data fusion methods. In the next section, we briefly introduce IndoorGML. In Sect. 4, we present the proposed seamless indoor-outdoor navigation model based on IndoorGML. In the next section, an experiment on the proposed model with CityGML is described, and the final section summarizes of this study and areas of further research.

[1] https://www.google.com/maps/about/partners/indoormaps.
[2] https://wiki.openstreetmap.org/wiki/Indoor_Mapping.
[3] https://register.apple.com/resources/imdf.

2 Related Work

Since IndoorGML has been published, various studies have been carried out to support navigation service [1,5,6,14]. Ryu et al. [14] presented Voice-based Indoor Maps (VIM) for visually impaired people. VIM is developed based on IndoorGML with extension model for landmark information of braille block. However, since it is based on IndoorGML, there is a limitation that it can be used only for indoor braille blocks. This is a common limitation of extending models based on IndoorGML.

Also, there are several studies on how to create an integrated model with IndoorGML and other standards [7,10]. Kim et al. [7] presented the methods for integrating IndoorGML and CityGML using the mapping between feature types of those standards. Also, they presented methods and guidelines of the automatic derivation of IndoorGML instance from CityGML LoD dataset. However, this study only considers connections between elements that represent the indoor space of two standard documents.

Lastly, there are a few studies about the connection between indoor and outdoor spaces based on IndoorGML [9,13]. Park et al. [13] presented the Topological Relation-based Data Fusion Model (TRDFM) using topological relations among spatial objects. Similar to our study, TRDFM was developed based on the concept of Anchor node in the IndoorGML document. However, this study extended the representation of outdoor spaces based on elements of IndoorGML. This method can not take advantage of existing standard data for outdoor space, so there is a restriction that TRDFM must rewrite outdoor spatial data even if data is representing the same area. To overcome these issues, we propose an extension model that making the connection between indoor and outdoor spaces, to integrating IndoorGML and other standards.

3 IndoorGML

The OGC IndoorGML was established as the OGC international standard for indoor navigation applications and XML-based formats to represent indoor spatial information. In this section, we briefly introduce IndoorGML and suggest the core concept of the model for indoor-outdoor seamless navigation service.

3.1 Basic Concept

IndoorGML expresses indoor space as two spatial models: Euclidean space represents the shape of cell spaces that is an element of the indoor space; Topology space represents the connectivity and adjacency between cell spaces. An indoor network can be derived from Topology Space using the Poincaré duality. Therefore, IndoorGML utilizes a network model for navigation and expresses the connectivity relationships among cell spaces. The nodes of the indoor network represent rooms, corridors, doors, elevators, and staircases. The edges of the indoor network represent the topological relationships among indoor spatial entities and can indicate the paths of pedestrian movement between nodes within a building. Therefore, one edge should be represented by two nodes. The network model in IndoorGML is represented by nodes (as called **State**) and edges (as called **Transition**) feature, as shown in Fig. 1.

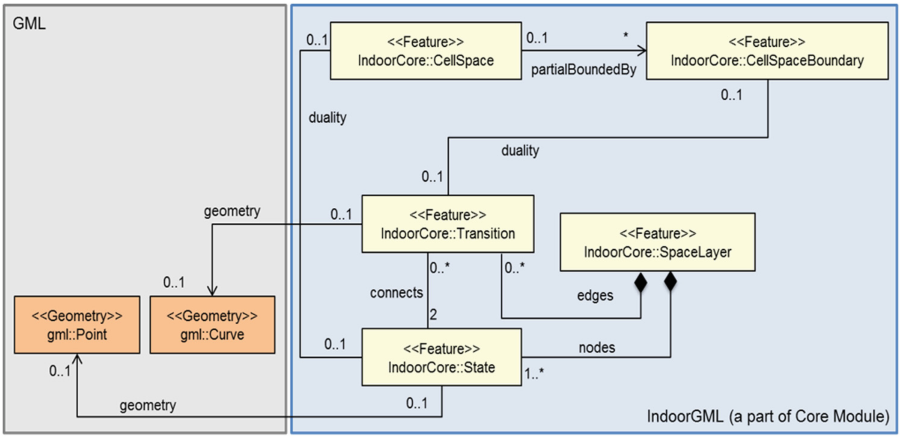

Fig. 1. Part of IndoorGML Core module UML diagram (OGC, IndoorGML [12])

3.2 Thin and Thick Model

IndoorGML is divided into two models depending on the representation of wall (and door) thickness. Thick wall model represents walls (and doors) has a certain thickness. In the case of thick wall model, walls and doors should be represent **CellSpace** (and **State**). On the other hand, thin wall model represents walls (and doors) has no thickness, like paper. In the case of thin wall model, walls and doors should be represent **CellSpaceBoundary** (and **Transition**). We need to consider both models for connection between indoor space and outdoor space. However, when expressing an entrance with Thin wall model, one **State** is required in the outdoor space according to the definition of transition. However, since **State** has a duality relation with **CellSpace** as shown in Fig. 1, it is necessary to express the outdoor space as **CellSpace** in order to create a **State** in outdoor space. However, this is not semantically equivalent to **CellSpace** defined in IndoorGML. In conclusion, the entrance should always be expressed as **State** like the door of the thick wall model.

3.3 Concept of Anchor Node

IndoorGML contains naive ideas for representing indoor and outdoor connections. In IndoorGML, “Entrance” is represented as a special node of the

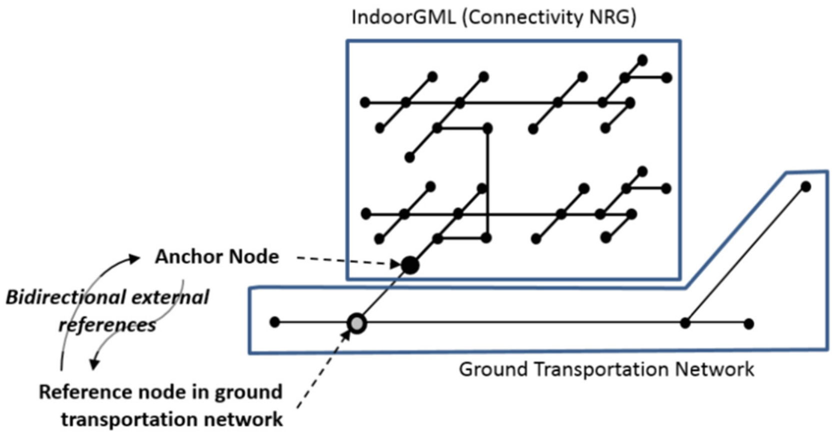

Fig. 2. Anchor node connecting indoor and outdoor networks (OGC, IndoorGML [12])

topological graph in an indoor space, connecting indoor and outdoor, called Anchor node, as shown in Fig. 2.

However, there is no object in the IndoorGML Core (and Navigation) model to represent connection information, and there are no specific guidelines for expressing indoor and outdoor connection information using only existing objects in the IndoorGML schema. To solve this problem, in [13], indoor and outdoor connections are expressed by extending **State** and **Transition** to **SpecialState** (Anchor node) and **SpecialTransition** (Anchor edge) for making an indoor-outdoor unified model. However, this unified model has the problem that all indoor and outdoor data must be described in the document of IndoorGML, even if outdoor data already exists with another format. Also, expressing the outside network using only the elements of IndoorGML do not fit from the characteristic of the elements of IndoorGML. Therefore, we need to extend IndoorGML by expressing entrance elements and making a link for the outside network, for representing indoor-outdoor connection information.

4 Seamless Navigation Model for IndoorGML

This section describes our Seamless Navigation Module (SNM) as an extension of IndoorGML based on the concept of Anchor node, which is the special node that represents the entrance of the building. SNM focuses on reusing other standards that represent outdoor spaces and making connections with them, as shown in Fig. 3. In order to connect two different spaces, we define the following attributes for an instance of Anchor nodes:

- Parameters for CRS conversion.
- External reference to the outdoor transportation network.

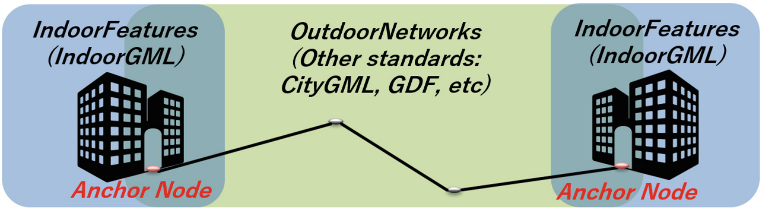

Fig. 3. Concept of Seamless Navigation Model

4.1 Conversion Method of CRS

For seamless navigation, conversion of the Coordinate Reference System (CRS) is necessary to integrate indoor and outdoor coordinate systems. In cases where the global CRS is used for indoor space, the conversion parameters are not necessary. However, in case of using the local CRS, four parameters are required to Cartesian coordinate system conversion:

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

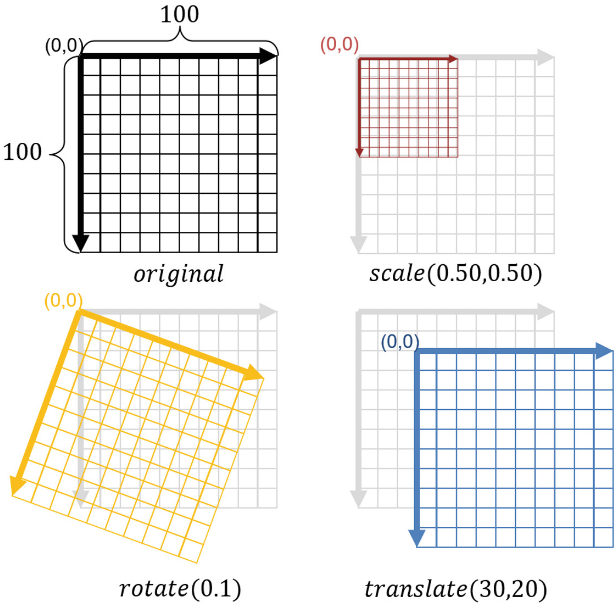

Fig. 4. Example of transform methods (2-dimensional case)

Firstly, we need origin to perform the transformation. Next, a scale value between the local coordinate system and the global coordinate system is required. Thirdly, the rotation angle of each axis is required for the rotation movement between the coordinate systems. Finally, we need a translation vector for parallel movement between coordinate systems. Examples of the conversion methods are shown in Fig. 4.

Unlike scaling and translation, the rotation is affected by the order in which the parameters (rotation angles) are applied. We usually can use Euler angle for 3D rotation, described in [2]. Euler angles are described as a sequence of rotations about three mutually orthogonal coordinate axes fixed in R^3 space. In this paper, we used yaw, pitch, and roll rotation, one of the sequences of Euler angles.

Yaw is a counterclockwise rotation of α around the z-axis. The rotation matrix is given by as follow:

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

Similarly, a pitch is a counterclockwise rotation of β around the y-axis, and a roll is a counterclockwise rotation of γ around the x-axis. The rotation matrix of pitch and roll are given by as follow:

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

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

$$R(\alpha, \beta, \gamma) = R_z(\alpha) R_y(\beta) R_x(\gamma)$$

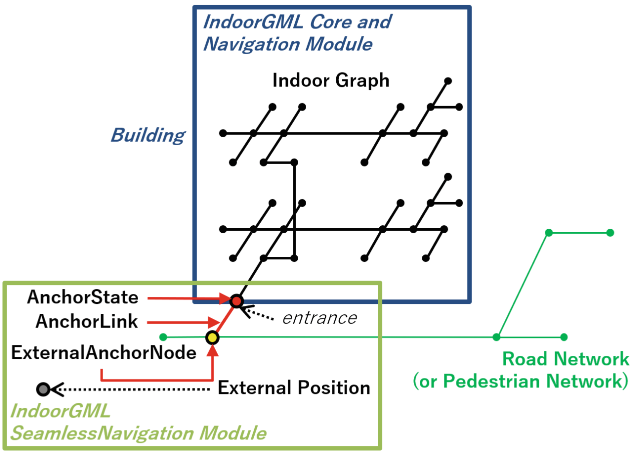

Fig. 5. Concept of IndoorGML Seamless Navigation Model

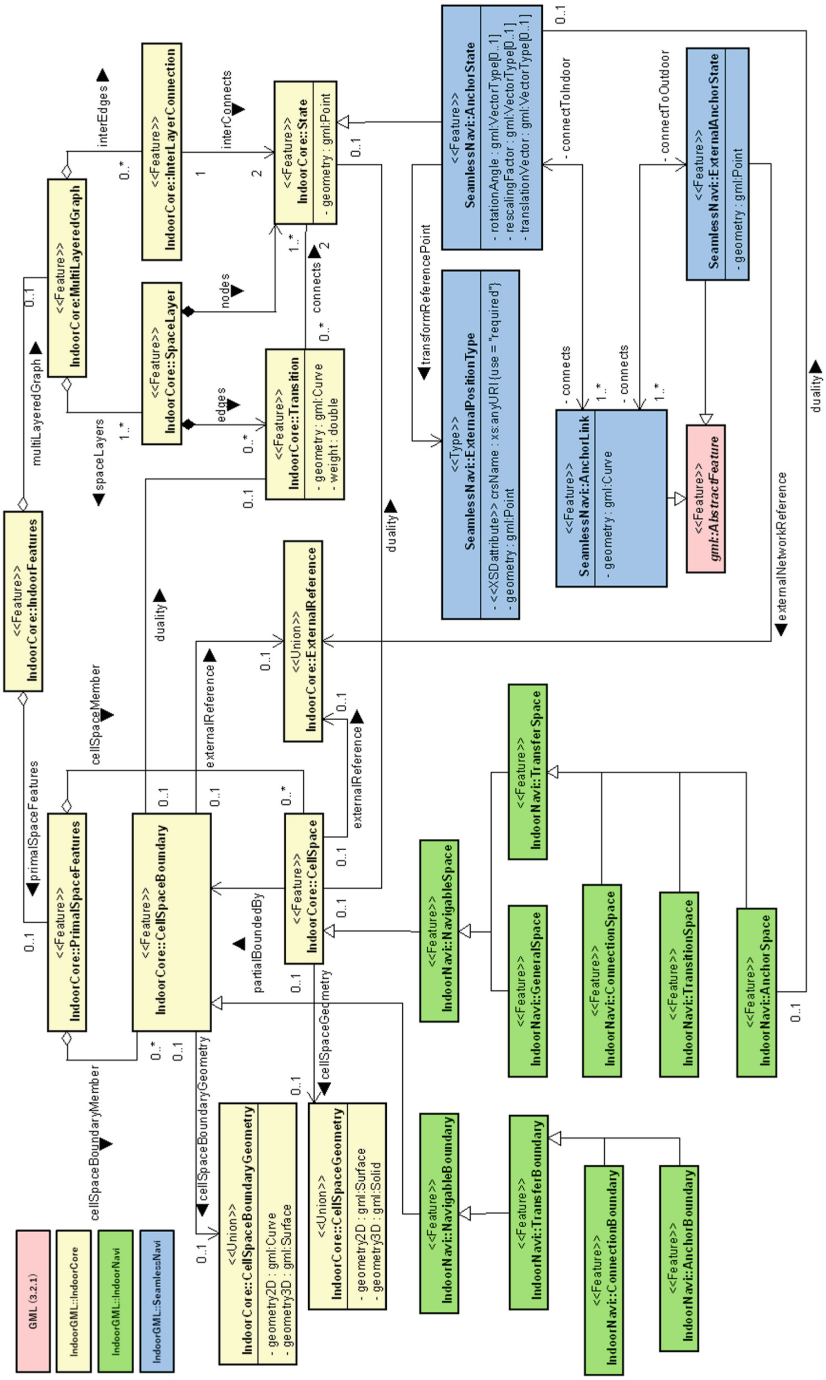

Fig. 6. UML diagram for Seamless Navigation Model

4.2 Seamless Navigation Model

The proposed Seamless Navigation Model is shown in Fig. 5. The SNM consists of three elements: **AnchorState**, **AnchorLink**, and **ExternalAnchorNode**. The UML diagram depicted in Fig. 6 shows the SNM data model based on the IndoorGML core and navigation module.

⟨AnchorState⟩
AnchorState represents a node that provides the connection between indoor space and outdoor space. It refers to entrance doors, and it can be used as a control point for indoor-outdoor integration. The XML schema for **AnchorState** is shown in Fig. 7. Each element in **AnchorState** are described as follow:

The *duality* element represents an association with the corresponding **AnchorSpace** class which represents a special opening space. **AnchorState** has a geometry that derived from **State** class, and it is one of the end points of curve geometry of **AnchorLink**. The *connects* element represents an association with the corresponding **AnchorLink** class. Lastly, **AnchorState** contains the conversion parameters for converting the coordinate of each point of indoor geometry according to the global CRS, if the local CRS is applied to the indoor space. In cases where the global CRS is used for indoor space, the conversion

```
<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="rescailingFactor" 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="AnchorStatePropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="AnchorState"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
<xs:complexType name="AnchorSpacePropertyType">
    <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>
```

Fig. 7. XML schema for **AnchorState**

parameters are not necessary. The *TransformReferencePoint* is **ExternalPosition** type element and a reference point that using for conversion. **ExternalPosition** is a point in the global CRS. Therefore, **ExternalPosition** must have an attribute *crsName* to represent the used outdoor network's CRS.

AnchorState have elements for CRS conversion: *transformReferencePoint* $p_o(x_0, y_0, z_0)$, *rotationAngle* $R(s_x, s_y, s_z)$, *rescailingFactor* $A(\alpha, \beta, \gamma)$, *translationVector* $T(t_x, t_y, t_z)$. Conversions using these parameters will produce results in the order in which they are applied. This paper assumes that the transformation is performed in the order as shown in Fig. 8: *Rotation* $\rightarrow$ *Scaling* $\rightarrow$ $Translation$.

In the case of rotation, the rotation is 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,

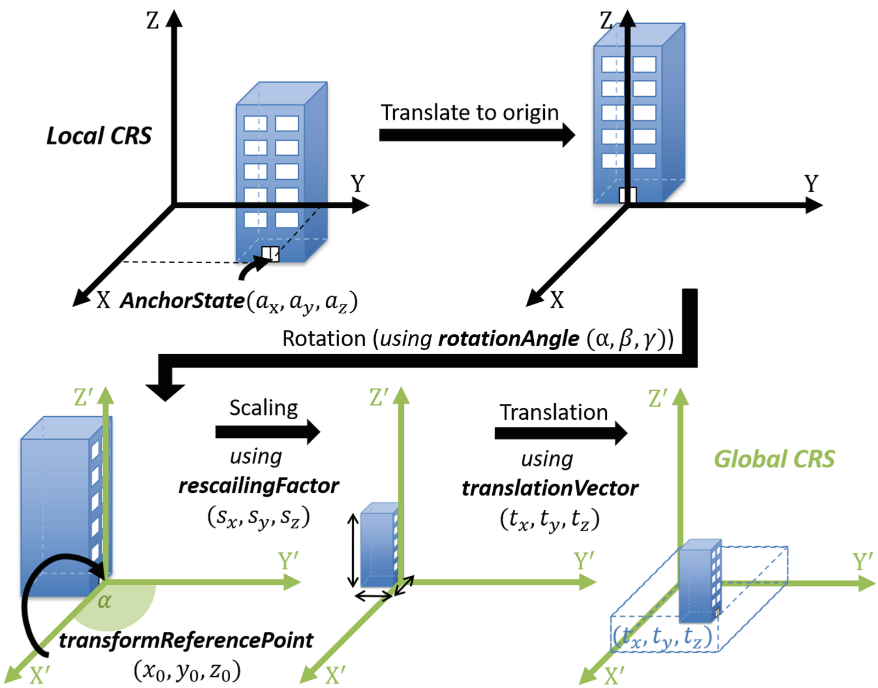

Fig. 8. Process of CRS conversion

$Convert(x, y, z, p_a, p_o, R, S, T)$ using the given parameters is as follows:

$$
\begin{aligned}
&Convert(x, y, z, p_a, p_o, R, S, T)\\
=&R_z(\alpha)R_y(\beta)R_x(\gamma)S(x-a_x, y-a_y, z-a_z)+p_o+T\\
=&\begin{bmatrix} \cos(\alpha)\cos(\beta) & \cos(\alpha)\sin(\beta)\sin(\gamma)-\sin(\alpha)\cos(\gamma) & \cos(\alpha)\sin(\beta)\cos(\gamma)+\sin(\alpha)\sin(\gamma)\\ \sin(\alpha)\cos(\beta) & \sin(\alpha)\sin(\beta)\sin(\gamma)+\cos(\alpha)\cos(\gamma) & \sin(\alpha)\sin(\beta)\cos(\gamma)-\cos(\alpha)\sin(\gamma)\\ -\sin(\beta) & \cos(\beta)\sin(\gamma) & \cos(\beta)\cos(\gamma) \end{bmatrix}\\
&*\begin{bmatrix} s_x*(x-a_x)\\ s_y*(y-a_y)\\ s_z*(z-a_z) \end{bmatrix} + \begin{bmatrix} x_0+t_x\\ y_0+t_y\\ z_0+t_z \end{bmatrix}
\end{aligned}
$$

⟨ExternalAnchorNode⟩

ExternalAnchorNode represents a node that represents the position on the outdoor network. The XML schema for **ExternalAnchorNode** as shown in Fig. 9. The *externalNetworkReference* element references to outdoor network in other standards; e.g., CityGML, GDF, etc. It is represented geometrically as **Point** in GML. It is one of the end points of curve geometry of **AnchorLink**. The *connects* element represents an association with the corresponding **AnchorLink** class.

```
<xs:element name="ExternalAnchorNode" type="ExternalAnchorNodeType"
substitutionGroup="gml:AbstractFeature"/>
<xs:complexType name="ExternalAnchorNodeType">
    <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="ExternalAnchorNodePropertyType">
    <xs:sequence minOccurs="0">
        <xs:element ref="ExternalAnchorNode"/>
    </xs:sequence>
    <xs:attributeGroup ref="gml:AssociationAttributeGroup"/>
</xs:complexType>
```

Fig. 9. XML schema for **ExternalAnchorNode**

⟨AnchorLink⟩

AnchorLink represents an edge between the indoor network and outdoor networks. The XML schema for **AnchorLink** as shown in Fig. 10. **AnchorLink** must have connection **AnchorState** and **ExternalAnchorNode**. For the geometrical representation of an **AnchorLink**, a **Curve** geometric primitive object from the GML is used. This geometry is derived from the geometry of **AnchorState** and **ExternalAnchorNode**.

```xml
<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="ExternalAnchorNodePropertyType"/>
                <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>
```

Fig. 10. XML schema for **AnchorLink**

5 Experimental Examples

For better understanding, this paper provides a use-case of the SNM for the Transportation model of CityGML 2.0 [11]. This section briefly introduces the CityGML Transportation model and suggests some guidelines for using the sample data to create the SNM data.

5.1 Use-Case with CityGML 2.0

CityGML is an official international standard data model and exchange format to store digital 3D models of cities and landscapes. CityGML support transportation model that focuses on thematic as well as on geometrical/topological aspects, as shown in Fig. 11. The main class is **TransportationComplex**, which is composed of the parts **TrafficArea** and **AuxiliaryTrafficArea**. In the Level of Detail (LOD) 0, the transportation complexes are modeled by line objects establishing a linear network. On this level, pathfinding algorithms or similar analyses can be executed.

Therefore, the geometry of **ExternalAnchorNode** 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. Also, the geometry of **AnchorLink** can be derived from two points which are the geometries of **AnchorState** and **ExternalAnchorNode**.

5.2 Case: AIST Tokyo Waterfront Annex

The sample data for the use-case have been conducted at a real site: AIST Tokyo Waterfront Annex in Japan as shown in Fig. 12. This sample data shows the basic structure of the SNM data and how the SNM and CityGML Transportation model datasets are linked via external references. For simplicity, the

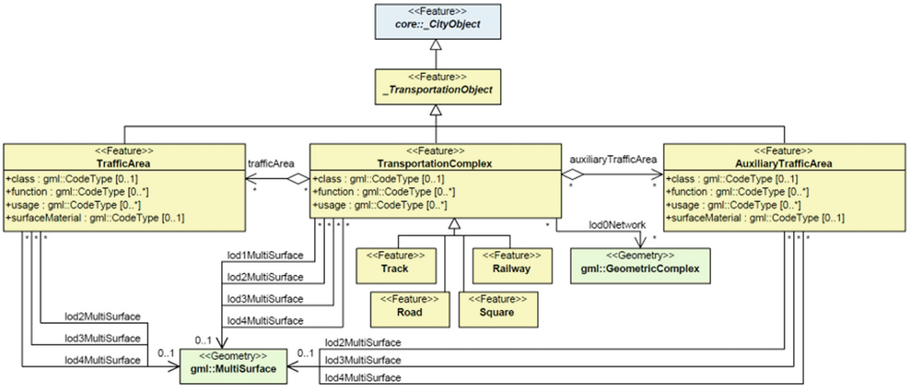

Fig. 11. UML diagram of transportation model in CityGML 2.0 (OGC, CityGML [11])

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) 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.

The detailed contents of sample data for the **AnchorState** class is shown in Fig. 13. This data consists of IndoorGML Core and Navigation modules. **AnchorState** can have a geometry derived from **State** class. This geometry is used to create the geometry of **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*. *TransformReferencePoint* must have a CRS information as shown in the yellow part of Fig. 13. *connects* is represented by xlink for GML id of **AnchorLink** class instance, as shown in the green part of Figs. 13 and 16. **AnchorState** can have a *duality* association with the **AnchorSpace** class instance, as shown in the blue part of Fig. 13.

Figure 14 shows **ExternalAnchorNode** 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 Figs. 14 and 15. The geometry of **ExternalAnchorNode** is derived from one of the points on a *lod0network*, as shown in the yellow part of Figs. 14 and 15. *connects* is represented by xlink for GML id of **AnchorLink** class instance, as shown in the green part of Figs. 14 and 16.

AnchorLink sample data is shown in Fig. 16. 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* instance.

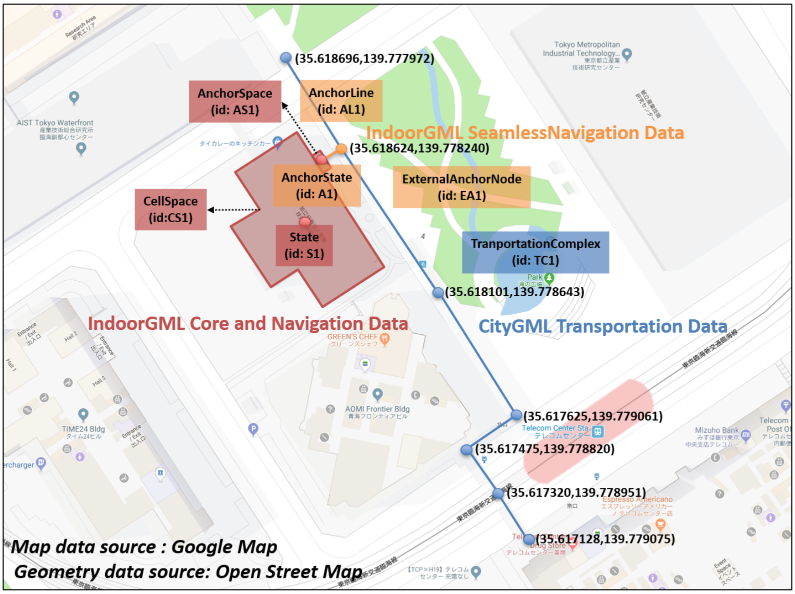

Fig. 12. Use-case site - AIST Tokyo Waterfornt Annex

```
<core:IndoorFeatures gml:id="IFs">
   ...
  <core:cellSpaceMember>
    <navi:AnchorSpace gml:id="AS1">
      <gml:name>Entrance</gml:name>
        ...
    </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>
```

Fig. 13. Part of sample data for **AnchorState** (seamlessNaviSample.gml) (Color figure online)

```xml
<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>
```


Fig. 14. Part of sample data for **ExternalAnchorNode** (seamlessNaviSample.gml) (Color figure online)


```xml
<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>
```


Fig. 15. Part of sample data for **TransportationComplex** (cityTransSample.gml) (Color figure online)


```xml
<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>
```


Fig. 16. Part of sample data for **AnchorLink** (seamlessNaviSample.gml) (Color figure online)

6 Conclusion

Indoor and outdoor spatial integration is one of the traditional topics in GIS, particularly location-based services such as navigation for pedestrians (and robots). Several researchers have developed several data fusion methods to integrate the data sets generated based on various data models. This study presented the OGC IndoorGML extension module for seamless navigation between indoor and outdoor space, called Seamless Navigation Model (SNM). The SNM is developed based on the concept of Anchor node. To realize the connectivity of different spatial data based on the topological relations using the SNM, we provided a use-case about connecting with OGC CityGML.

However, further researches are required to implement the purposed extension model to handle the various GIS formats. Further research will include additional use-case with international standards, for example, Geographic Data Files (GDF). And then, finally, we plan to submit the SNM to OGC as a discussion paper with use-cases.

Acknowledgment. This research was partially supported by the New Energy and Industrial Technology Development Organization (NEDO) and a grant (19NSIP-B135746-03) from National Spatial Information Research Program (NSIP) funded by Ministry of Land, Infrastructure and Transport of Korean government.

References

1. Alattas, A., Zlatanova, S., Van Oosterom, P., Chatzinikolaou, E., Lemmen, C., Li, K.J.: Supporting indoor navigation using access rights to spaces based on combined use of IndoorGML and LADM models. ISPRS Int. J. Geo-Inf. **6**(12), 384 (2017)
2. Arfken, G.B., Weber, H.J.: Mathematical methods for physicists. AAPT (1999)
3. BuildingSMART: Industry Foundation Classes (IFC) Version 4.1, IFC4X1 (2017)
4. Chen, J., Clarke, K.C.: Modeling standards and file formats for indoor mapping. In: Procedings of the 3rd International Conference on Geographical Information Systems Theory, Applications and Management, pp. 268–275 (2017)
5. Iida, H., Hiroi, K., Kaji, K., Kawaguchi, N.: A proposal of IndoorGML extended data model for pedestrian-oriented voice navigation system. In: Proceedings of the Seventh ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness, ISA 2015, pp. 2:1–2:6. ACM, New York (2015)
6. Jung, H.J., Lee, J.: Development of an omnidirectional-image-based data model through extending the IndoorGML concept to an indoor patrol service. J. Sens. **2017** (2017). https://www.hindawi.com/journals/js/2017/5379106/cta/
7. Kim, J.-S., Yoo, S.-J., Li, K.-J.: Integrating IndoorGML and CityGML for indoor space. In: Pfoser, D., Li, K.-J. (eds.) W2GIS 2013. LNCS, vol. 8470, pp. 184–196. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55334-9_12
8. Kim, Y.J., Kang, H.Y., Lee, J.: Development of indoor spatial data model using CityGML ADE. ISPRS-Int. Arch. Photogram. Remote Sens. Spat. Inf. Sci. **1**(2), 41–45 (2013)
9. Lee, J., Kang, H.Y., Kim, Y.J.: Developing data fusion method for indoor space modeling based on IndoorGML core module. J. Korea Spat. Inf. Soc. **22**(2), 31–44 (2014)

10. Liu, L., Zlatanova, S., Zhu, Q., Li, K.: Towards the integration of IndoorGML and IndoorlocationGML for indoor applications. ISPRS Ann. Photogram. Remote Sens. Spat. Inf. Sci. **4**, 343 (2017)
11. Open Geospatial Consortium: OGC City Geography Markup Language (CityGML) Encoding Standard, OGC 12-019 (2012)
12. Open Geospatial Consortium: OGC Indoor Geographic Markup Language (IndoorGML), OGC 14-005r5 (2018)
13. Park, J., Ahn, D., Lee, J.: Development of data fusion method based on topological relationships using IndoorGML core module. J. Sens. **2018** (2018). https://www.hindawi.com/journals/js/2018/4094235/cta/
14. Ryu, H.G., Kim, T., Li, K.J.: Indoor navigation map for visually impaired people. In: Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness, ISA 2014, pp. 32–35. ACM, New York (2014)
15. Teo, T.A., Cho, K.H.: BIM-oriented indoor network model for indoor and outdoor combined route planning. Adv. Eng. Inform. **30**(3), 268–282 (2016)
16. Wang, Z., Niu, L.: A data model for using OpenStreetMap to integrate indoor and outdoor route planning. Sensors **18**(7), 1–19 (2018)