

## Deriving the Technical Model for the Indoor Navigation Prototype based on the Integration of IndoorGML and LADM Conceptual Model

Alattas, Abdullah; Oosterom, Peter van; Zlatanova, Sisi

**Publication date**

2018

**Document Version**

Final published version

**Published in**

Proceedings of the 7th Land Administration Domain Model Workshop

**Citation (APA)**

Alattas, A., Oosterom, P. V., & Zlatanova, S. (2018). Deriving the Technical Model for the Indoor Navigation Prototype based on the Integration of IndoorGML and LADM Conceptual Model. In C. Lemmen, P. van Oosterom, & E. Fendel (Eds.), *Proceedings of the 7th Land Administration Domain Model Workshop* (pp. 245-268). International Federation of Surveyors (FIG).

**Important note**

To cite this publication, please use the final published version (if applicable). Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

# **Deriving the Technical Model for the Indoor Navigation Prototype based on the Integration of IndoorGML and LADM Conceptual Model**

**Abdullah ALATTAS, Peter van OOSTEROM, the Netherlands and  
Sisi ZLATANOVA, Australia**

**Key words:** Indoor, Navigation, LADM, Database, Prototype

## **SUMMARY**

This paper shows the conversion of LADM-IndoorGML conceptual model to technical model. The aim of this research is assessing the conceptual model and discovering the possibilities and the shortcomings of the conceptual model via the conversion to technical model. There are three steps to convert the conceptual model:

- Prepare the LADM-IndoorGML UML model;
- Transform the class diagram to table diagram;
- Generate SQL DDL code from the table diagram.

During the work from step to step several issues did appear and they are addressed in this paper to enable more automated transformation possibilities from the conceptual model to technical mode. Most of the issues are quite generic and also applicable when converting other conceptual models into technical models. There are a few issues related to our specific conceptual models (IndoorGML and LADM packages) and the Enterprise Architect software (which is the used tool in our case), but most issues are generic: the primary keys that have been created to all tables by the software even if there are ID attributes, foreign keys, the association multiplicity, the attributes multiplicity, data type, spatial data type, index, spatial index, constraints, and inheritance. The research shows that there is a need to develop a complete UML diagram for IndoorGML that contains all the attributes and their datatypes. The current class structure of the code list in LADM standard has been developed for better semantic meaning during the implementation by creating an intermediate conceptual model class to support the transformation. Many critical decisions have been taken during the derivation of the technical model to solve these issues.

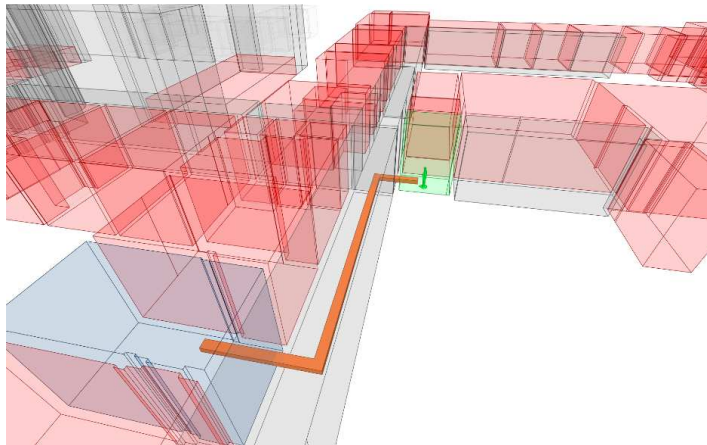
# Deriving the Technical Model for the Indoor Navigation Prototype based on the Integration of IndoorGML and LADM Conceptual Model

Abdullah ALATTAS, Peter van OOSTEROM, the Netherlands and  
Sisi ZLATANOVA, Australia

## 1. INTRODUCTION

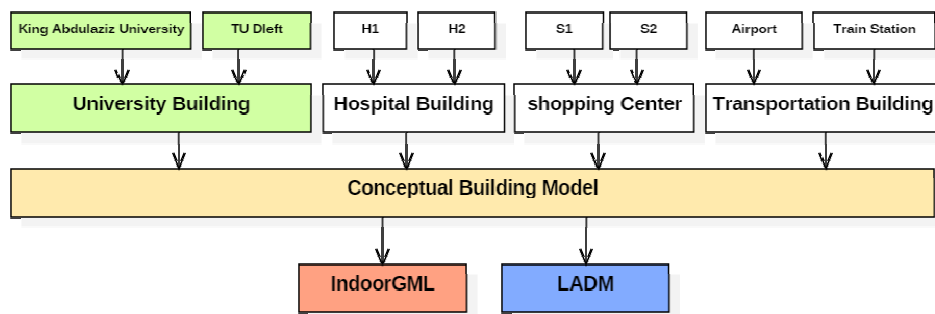
The indoor environment of our cities is huge and complicated and that affects the movement of users. Because of the complexity of the indoor environments, users require assistance to reach their destinations. For this propose many indoor applications have been developed to guide the user by applying its own semantic and geometric description of space. The current indoor navigation applications do not provide the indoor navigation path based on the access rights, and that allow the users to reach spaces that are forbidden for their type. For example, an educational building has several categories of users such as students related to different programs, employees related to different departments (academic staff, administrative staff, and maintenance staff), visitors related or not related to the university. Each user of these categories has individual indoor activities that differ over time and from the rest of users, and the location of these activities are affected by the user profile. To distinguish between the activities of the users of the indoor space, many types of information have to be addressed from several sources such as facility management (room number, title, function, and access), university database (for students schedules and employees tasks descriptions, visitors appointment information), a standard for rights, restrictions, and responsibilities and spatial database (that contain geometry and semantics information of the indoor spaces). Currently, no standard or commercial application contains all information that is required for such advanced indoor navigation task based on the user rights. Therefore, it is critical to uniquely specify the type of objects and their properties and identify the spaces which are available for navigation for a specific user profile based on access rights. A conceptual model proposed by (Alattas et al, 2017) has introduced an approach taking into consideration the integration of IndoorGML and Land Administration Domain Model (LADM) to determine the accessibility of the indoor spaces based on the ownership or the functional right of use for educational buildings.

IndoorGML is an Open Geospatial Consortium (OGC) standard that presents a framework for an indoor navigation system which explains the indoor space and Geography Markup Language (GML) syntax for encoding geoinformation (Zlatanova et al, 2016). Land Administration Domain Model (LADM) is an ISO standard (19152:2012) that concentrates on rights, responsibilities, and restrictions (RRR) that influence lands and space/construction elements (Lemmen et al, 2015). The LADM-IndoorGML conceptual model defines the user-space access rights over time to provide efficient navigation for individual or group of users of the indoor environments. The LADM-IndoorGML conceptual model applies LADM to establish a relationship between the indoor spaces and the users, by assigning rights, restrictions, and responsibilities to each indoor space to indicate the accessible spaces for each type of user.



**Figure 1. Indoor navigation based on LADM Rights, Restrictions and Responsibilities; red spaces are not accessible, blue spaces are accessible.**

After determining the user-space access right for each type of user of the indoor space, IndoorGML provides navigation path based on the current access rights for the user as shown in Figure 1. The conceptual model is part of a broad hierarchical framework of models for space subdivision based on the right, restriction, and responsibility of the users as shown in Figure 2.



**Figure 2. Hierarchical framework of conceptual models**

In order to assess the conceptual model, it is tested with real-world data applications. This paper presents the development of the technical model and draws conclusions with respect to the possibilities and the drawbacks of the conceptual model. For the purpose a model-driven approach is used (Hespanha et al, 2008) applying model transformation patterns (Jacob et al, 2008) as provided by a commercial software. During the development of the prototype, several decisions about the design of the conceptual model have been taken to convert the conceptual model into the technical model. A modeling tool is used such as Enterprise Architect (EA) to convert the UML diagrams into a database. The work was completed in in the following five steps:

- First, using the ISO TC211 EAP (EA Project) file, the LADM class diagram was modified by adding the new classes of LADM-IndoorGML conceptual model.

- Second, IndoorGML UML model with attributes was created. They were taken from the XML/GML schema of IndoorGML standard and the complete UML model was built by using the code reverse engineering of Enterprise Architect.
- The third step was to include the new classes of LADM-IndoorGML model to the LADM class diagram.
- Then the diagram was converted to table diagram.
- The last step was to generate the SQL DDL (Data Definition Language) from the table diagram.

During each step, all issues have been considered to find a solution to have a fully automated converting tool from conceptual to technical model.

## 2. RELATED WORK

### 2.1 LADM-IndoorGML conceptual model

The integrated model of LADM and IndoorGML that has been proposed by (Alattas et al, 2017) shows that the link between the two standards is through the CellSpace class in IndoorGML and the LA\_SpatialUnit in LADM to define the accessibility for the user based on their right. The right, restriction, and responsibilities are based on the classes of LADM. Each indoor environment can be used and reached by different types of users. For each type of user, there are specific spaces and times that they have the right access to use. So, by regulating the access of the user according to their right, the indoor navigation will be more efficient. The main idea of the integration is that the LADM associates the subdivision of the indoor space based on the right of the user. The association between the LA\_SpatialUnit and the CellSpace provides the space ID and the function of the space as shown in Figure 3. The LA\_RRR collects the function of the space to assign a specific right access for each type of users. Based on the right of the subdivision of the spaces by LADM, an IndoorGML based application can provide the indoor path for the users.

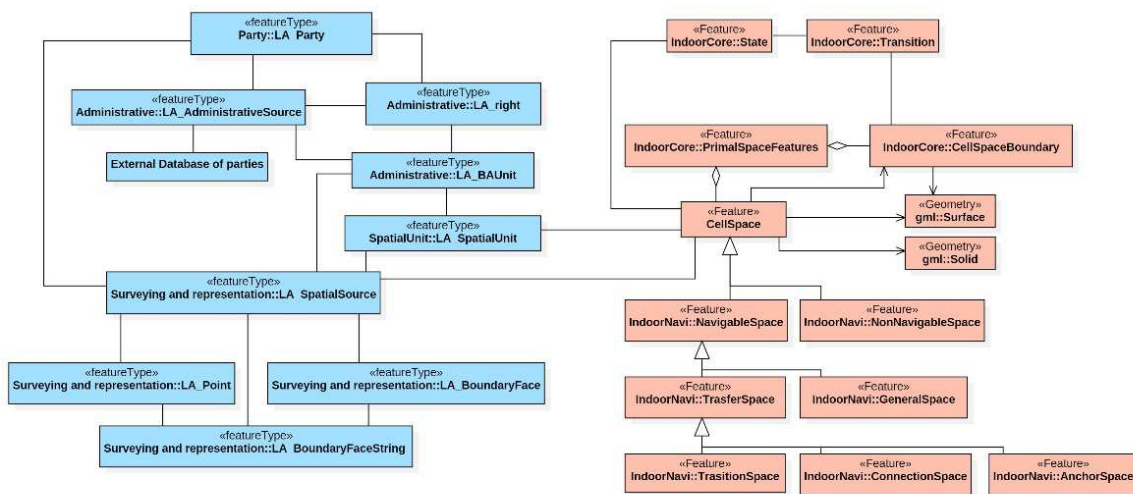


Figure 3. The integrated model of LADM (blue) and IndoorGML (red) (Alattas et al, 2017)

There are two different spaces defined by LADM, common right spaces and private right spaces. According to the right type of the user, the subdivision of the indoor space will be generated.

The LADM Party package represents the user information, and it consists of LA\_Party class, subclass LA\_GroupParty, and LA\_PartyMember which is an association class between LA\_Party and LA\_GroupParty (Figure 4). For better representation of the user, the Party Package has been further refined to include three additional classes and their code list classes as shown in Figure 4. The LA\_GroupParty has two additional subclasses to represent specific groups of users in a different class: LA\_GroupStudent and LA\_GroupEmployee. Each subclass of the LA\_GroupParty has attributes that will be used for the representation of the subclass. The attributes of the LA\_Party and the LA\_GroupParty will be inherited to the subclass according to the generalization association that connects the classes.

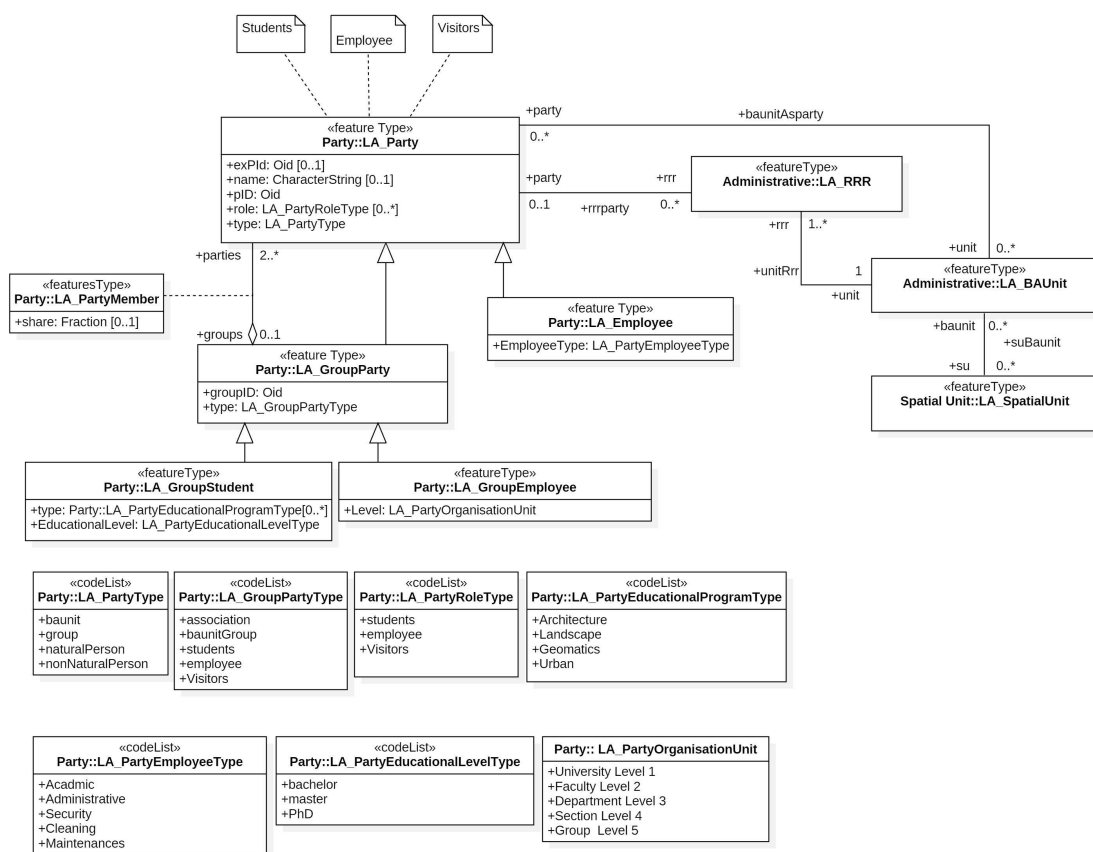


Figure 4. The party package of the LADM-IndoorGML conceptual model (Alattas et al, 2017)

## 2.2 Deriving the technical from the conceptual model in the case of Malaysian LADM country profile

In related work, (Zulkifli et al, 2014) have described the development of a prototype for the assessment of Malaysian LADM country profile. The Malaysian LADM country profile consists of two parts: administrative and spatial. As mentioned above, Enterprise Architect provided an automated transformation from the conceptual UML model into a technical model. By having the technical model, they could convert a sample data from JUPEM (spatial) and land office (legal and administrative) into the database. They have started by using UML diagram of the conceptual model that has been developed by Zulkifli in 2014 to create a database schema in Oracle Spatial. Each UML class refers to a table that has the same name in the database schema. There are views (names end with View), code lists tables (names end with Type), and additional tables to resolve many-to-many relation between two classes (names end with Relationship). Based on LADM standard, the Oid data type has two different parts: the namespace and a local ID. However, in their system JUPEM, they do not need to add namespace because it is the same for all objects in their system. Thus, the local ID is sufficient for the different identifier values. The main characteristic is that the ID must be unique for each object. However, for some objects, there could be several versions with the same ID and could be distinguished by their beginDateTime attributes that related to the VersionedObject class (Van Oosterom 1997). For each table of the database, there is a Primary Key (PK) with a set of one or more attributes. Each attribute value of the PK has to be unique for each entity. However, there can be several versions of an object (with the same ID) but with by some changed attribute values, and that means for one object there are several records in the same table with different begin date/time. To locate a unique record in the table, the ID and beginDateTime should be included to PK.

The authors have implemented each code list with its own table. The name of code list table has the extension Type which is added after the code list name. Because the code list could be updated, they have beginDateTime and endDateTime attributes. From other table a code list value can be referred to by using the ID (cID) attribute of the code list as FK. The development of the prototype that they have proposed consist of four steps as shown in Figure 5:

- Develop the LADM country profile for Malaysia (conceptual model).
- Obtain the database schema (technical model).
- Populate the Oracle Spatial database (converting and inserting the data).
- View, query and edit the data by employing Bentley MicroStation.

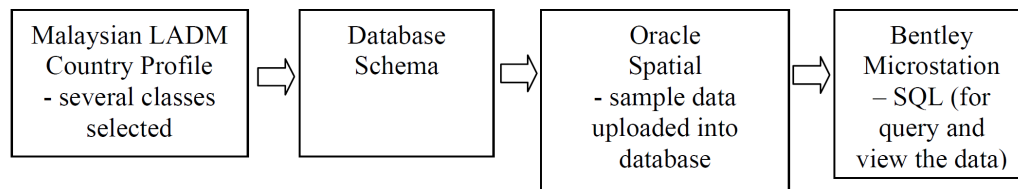


Figure 5. The four steps of the prototype development (Zulkifli et al, 2014)

### 3. TRANSFORMATION A CONCEPTUAL MODEL INTO A TECHNICAL MODEL

This section investigates the transformation of the LADM-IndoorGML conceptual model that has been proposed by (Alattas et al, 2017) into the technical model. This work aims to investigate the issues that may occur during the transformation from class diagram to table diagram and then to SQL DDL. Some more examples of mapping conceptual UML model to technical model can be seen in van Bennekom-Minnema 2008, Hespanha 2012 and Dilo et al, 2011. There are three steps we did take to convert class diagram into SQL DDL:

- As mentioned above, the first step is to create the LADM-IndoorGML class diagram (Subsection 3.1).
- The next step is to generate the LADM-IndoorGML tables diagram by using the transformation tool in Enterprise Architect software (Subsection 3.2).
- The last step is to convert the LADM-IndoorGML table diagram into SQL by using the code engineering tool (Subsection 3.3).

#### 3.1 Preparing LADM and IndoorGML UML model

##### 3.1.1 LADM UML model

The LADM model consists of several packages such as Party package, Administrative package, Spatial Unit package. Each of these packages contains the feature classes and the code lists for each package. The first step is to modify all packages by adding the new feature classes from the LADM-IndoorGML conceptual model. For example, the Party package in LADM standard does not contain LA\_Employee, LA\_GroupEmployee, and LA\_GroupStudent which they are part of the LADM-IndoorGML model. Therefore, these classes and their associations have to be included in the LADM UML model in the original file as shown in Figure 6 below.

##### 3.1.2 IndoorGML UML model

The IndoorGML standard (OGC GML 3.2.1) does not have a complete UML diagram (including attributes, their data types, and multiplicities) yet. Therefore, the XML/GML schema and the underdevelopment Java classes of IndoorGML has been used to derive the attributes into UML model (class diagram) and to make the conceptual model complete. The code engineering tool in Enterprise Architect has been used to generate the UML classes from the Java classes (via reverse engineering) as shown in Figure 7 below.

The result was a UML class for the core model of IndoorGML, and it represents the Java implementation. For example, the UML diagram included the edge and the node as classes; however, they are defined as the aggregations of SpaceLayer in the IndoorGML standard. Many attributes have been created in the UML model, which are not real attributes of classes in the standard but are used for the specific Java implementation to represent the associations between the classes. There are also many GetGeometry/SetGeometry/etc. operations to access these 'private' attributes. Therefore, the automatically generated UML class diagram for IndoorGML had to be corrected: making true attributes public, removing 'association implementation' attributes and removing the unneeded Get/Set operations.



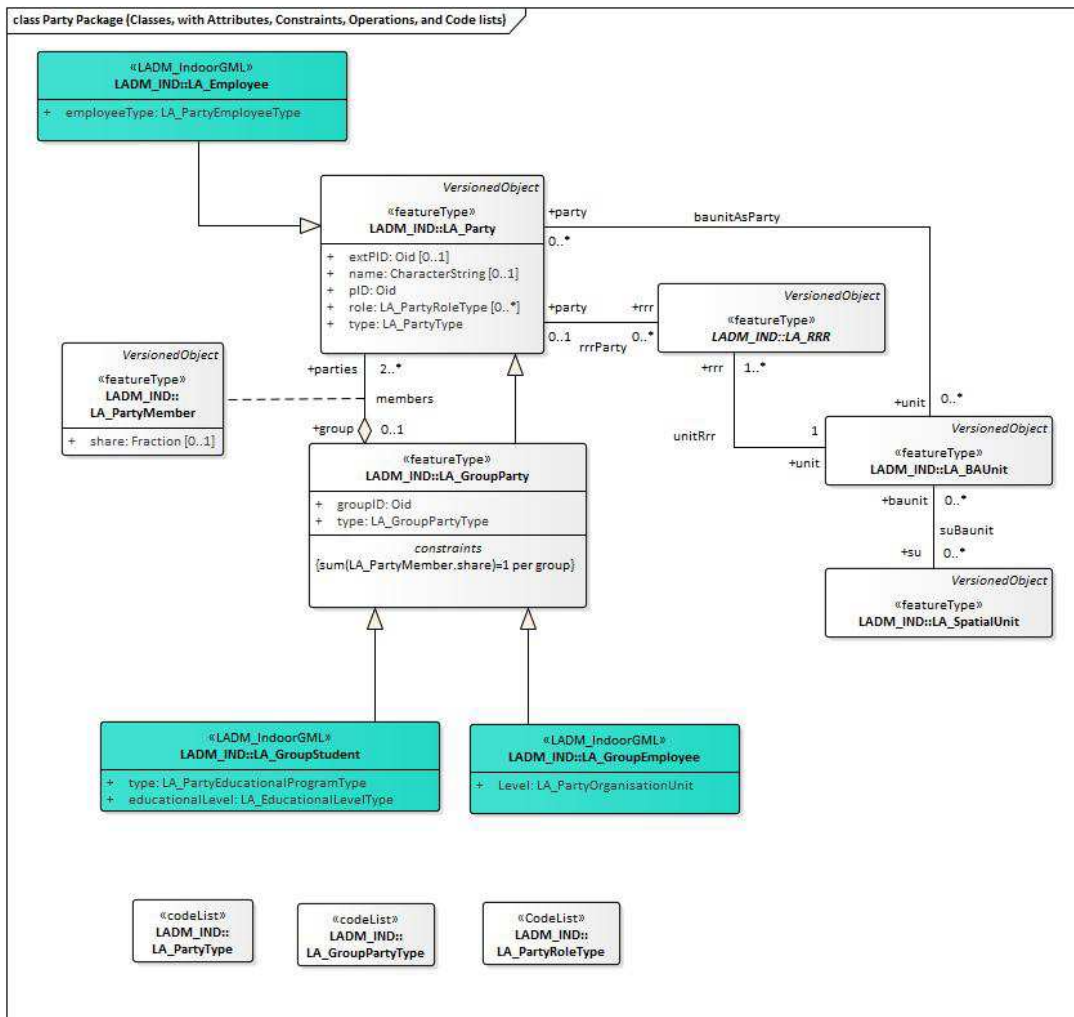


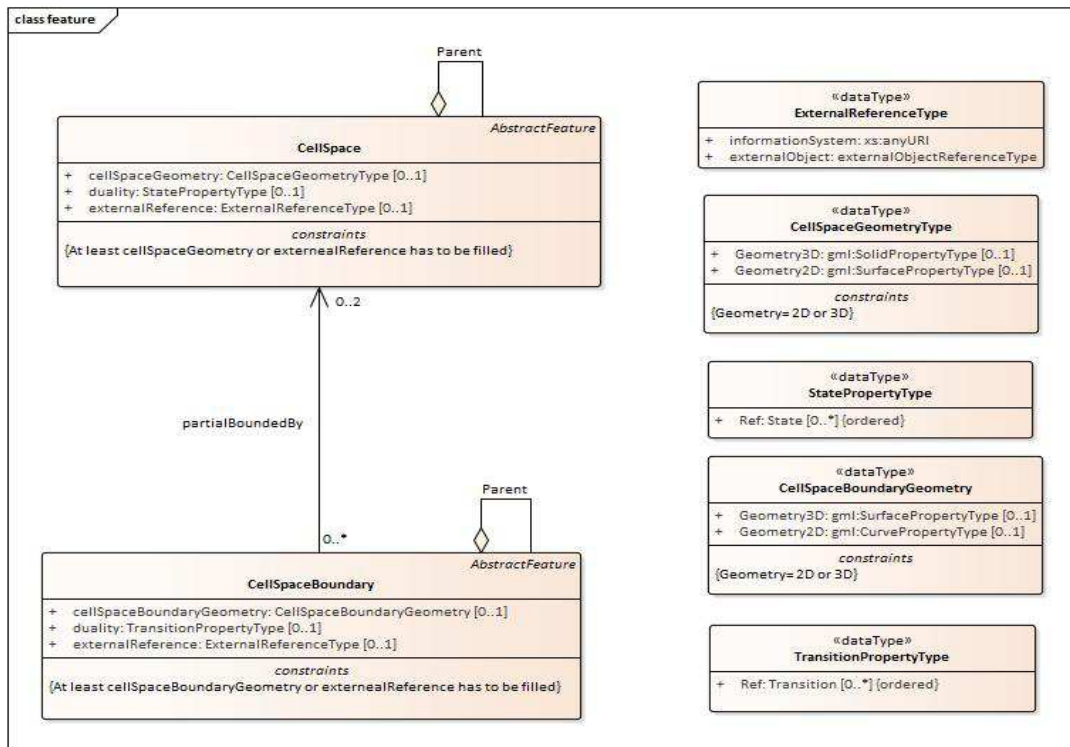
Figure 6. Party class diagram contain the LADM-IndoorGML new classes (in blue the assed classes)

The correction of the model has been done manually at this stage. According to the XML/GML schema of IndoorGML standard (Annex B), the CellSpace class has three attributes as shown in Figure 8 below:

- cellSpaceGeometry;
- duality;
- externalReference.

The data type of cellSpaceGeometry includes 2D (surface/area in 2D space), or 3D (solid/volume in 3D space) geometry and only one data type have to be chosen for each object. The duality attribute has StatePropertyType as a data type and it refers to zero or more states. Note that 'state' is the IndoorGML term for a node in the dual representation (OD dual as the counterpart of the CellSpace). The externalReferenceType is a data type for the externalReference attribute and has a two-part data type: part 1 is the informationSystem





**Figure 8. The corrected attributes for CellSpace class and cellSpaceBoundary**

The CellSpaceBoundary also has three attributes: cellSpaceBoundaryGeometry, duality, and externalReference which have more or less the same data types as the CellSpace class, with the small difference that duality attribute has TransitionPropertyType as a data type. Note that ‘transition’ is the IndoorGML term for an edge in the dual representation (1D dual as the counterpart of the CellSpaceBoundary). Also, it has a cellSpaceBoundaryGeometry attribute that has 2D (curve/line in 2D space) or 3D geometry (surface/area in 3D space) as a data type. The multiplicity related to the Parent associations for both CellSpace and CellSpaceBoundary are 0..1 at one side: CellSpace may have at most 1 parent, and at other side 0..\*, a CellSpace may have 0 or more sub-CellSpaces. The multiplicity of the partialBoundedBy association at side of CellSpace is 0..2 (as a boundary may have at both sides a CellSpace). Perhaps this could even be more if they are also allowed to bound sub-CellSpaces (and as we can have a deep hierarchy, then the 0.2 multiplicity actually becomes 0..\*, a bit losing the original meaning). Further, there is a constraint for both CellSpace and CellSpaceBoundary that at least the attribute cellSpaceGeometry or externalReference is filled. In summary two of the three 'attributes' of CellSpace and CellSpaceBoundary are actually references (duality and externalReference).

### 3.1.3 LADM-IndoorGML conceptual UML model

After building the new classes and their code list classes for all LADM packages and generated the UML model of IndoorGML, the next step was to create a new diagram for LADM-IndoorGML UML model under a new package. However, by inserting all the related

classes from different packages of LADM model into the new diagram, the Enterprise Architect software will not perform the transformation to generate DDL for the new diagram and by copying the files from their original source to the new LADM-IndoorGML file the software will perform the transformation but with incorrect or missing classes or associations. Thus, the only solution was to move all related classes to the conceptual model of LADM-IndoorGML from their sources into the new package after modifying them. Next step was to insert all the classes into the diagram to perform the transformation. The association and the multiplicity between the classes were created in automated way based on the original information that is stored in each package. Figure 9 shows the complete LADM-IndoorGML UML diagram.

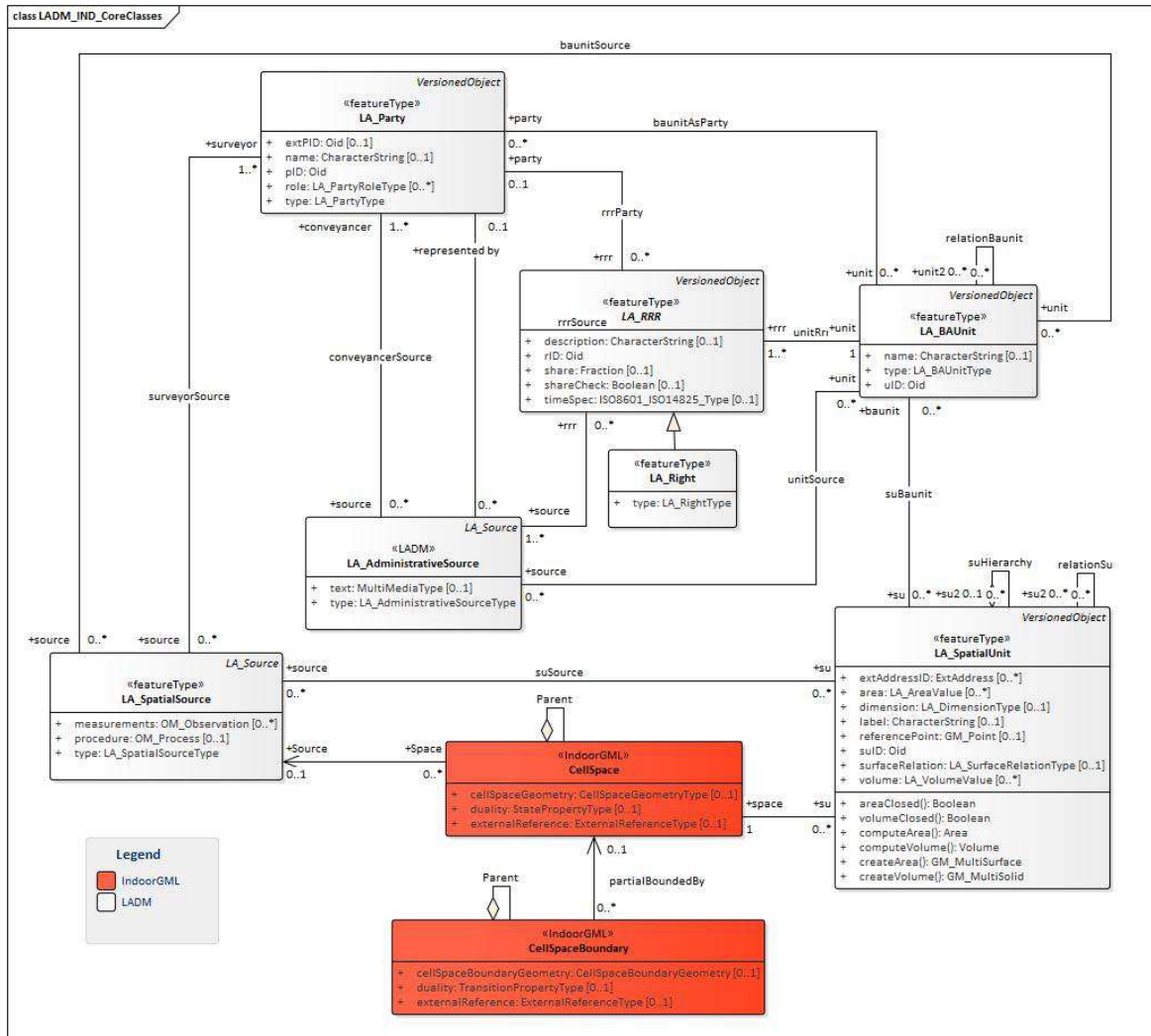


Figure 9. LADM-IndoorGML UML diagram

### 3.2 Generating Tables Diagram from the UML Diagram

By using the transformation tool model in Enterprise Architect, the UML diagram of LADM-IndoorGML is converting into table diagram. This step starts by defining the PostgreSQL as our default database in the software under the settings menu (there are many options for other databases). The transformation model conducts the transformation for the current package, and by including all the classes in the same package as described in the previous section, the transformation will create a table for each class and association between the classes as shown in Figure 10 below.

This subsection describes the different issues that are related to the transformation step as follows: inheritance (3.2.1), primary key and a foreign key (3.2.2), association class (3.2.3), multiplicity (3.2.4), constraints (3.2.5), data type (3.2.6), spatial data type (3.2.7), code list classes (3.2.8), and indexing (3.2.9).

#### 3.2.1 Inheritance

When the association between two tables is a generalization as in Figure 11, there are three options to inherit the attributes:

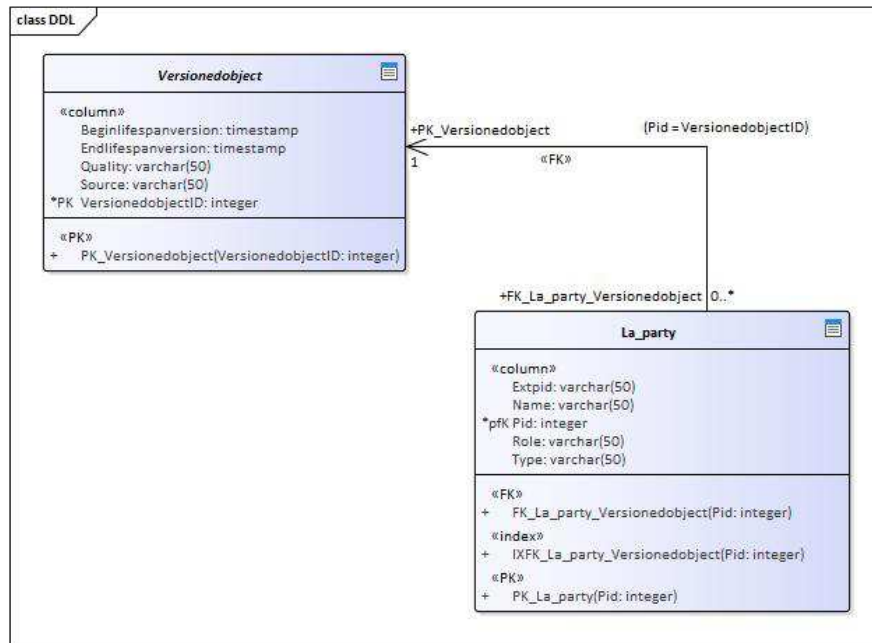
- By using a flat model which mean all the attributes of the parents are copied to the children classes. This option makes the model to contain the same attributes over and over again in all the children classes
- The second option is to create a super class that is used as a reference class and the children classes refer to the super class by using PK and FK. This option has been used in this case to avoid the duplication of the attributes all over the model.
- The third option is to define the children class as a subclass of the parent class by defining the parent class as object which will allow the children classes to store the attributes of the parent class.

#### 3.2.2 Primary Key and Foreign Key

The transformation tool creates a PK and a FK based on the relation between the tables. However, the software has generated a new unique attribute for each table even if the UML class has an ID attribute. The name of the new Primary key is “TableName + ID”. As shown in Figure 12 below, the LA\_Party table already has Pid attribute which is considered as the ID of the table, however, there is a new attribute called La\_partyID that has been created automatically by the software and has been used as PK for this table. To correct the PK for the tables that have an ID attribute, the correction has done manually by eliminate the association between the tables and then select the correct attribute.







**Figure 11. A generalization association**

### 3.2.3 Association Class

The LA\_PartyMember is an optional association class between LA\_party class and LA\_GroupParty. The transformation model created a table for LA\_PartyMember that has a separate connector to LA\_Party table and LA\_GroupParty as shown in Figure 12. The connector has used the PK to define the relationship between the tables.

### 3.2.4 Multiplicity of Association and Attributes

The transformation model does not transform the multiplicity of the attributes to tables, and there is no solution to fix this issue. Moreover, the multiplicity of the association (connectors) between the tables is different from the class diagram and that need to correct manually after the transformation. For example, the multiplicity for LA\_GroupParty in the UML model is 2..\*; however, the transformation replaced it with 1, and this has to be modified manually as shown in Figure 13 below.

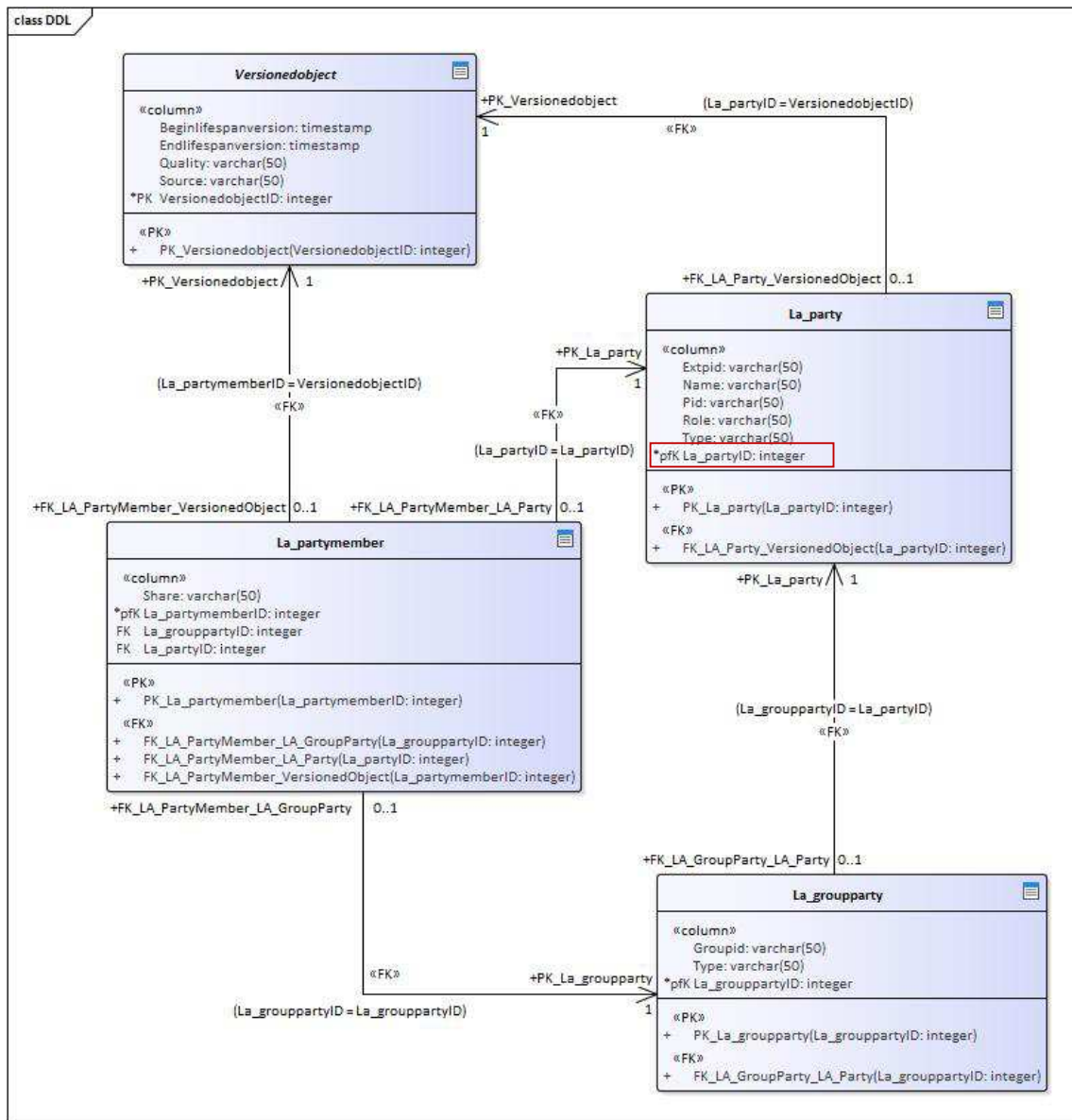
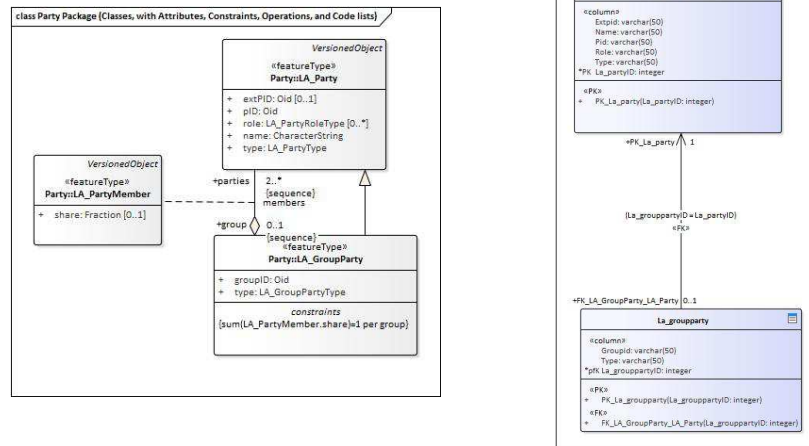


Figure 12. Additional Primary key has been created for each table





**Figure 13. The multiplicity for the association between the tables, left figure shows class diagram, right figure shows the tables diagram**

### 3.2.5 Constraints

The constraint of the Class diagram is modeled during the transformation to table diagram. For example, the LA\_group party has a constraint that the sum of the share of LA\_PartyMember association class is equal to one per group. However, the table diagram of the LA\_GroupParty does not contain the constraint even when changing the appearing feature of the table.

### 3.2.6 Data Type

The data type that the LADM-IndoorGML offers such as Oid and Fraction have a very important meaning. The Oid is a generic data type that used as an identifier for the objects. It consists of local identifier and nameSpace that used as an identifier for the data source. The transformation model does not consider the Oid and Fraction as data type and it replaces them with "varchar" in the table diagram. There is no solution to modify the data type in the table diagram. This issues could be solved by editing the SQL code. The rest of the attributes of the conceptual model have the same issue. They have a specific data type, however, the software during the transformation assign "varchar" as a data type for all of them.

### 3.2.7 Spatial Data Type

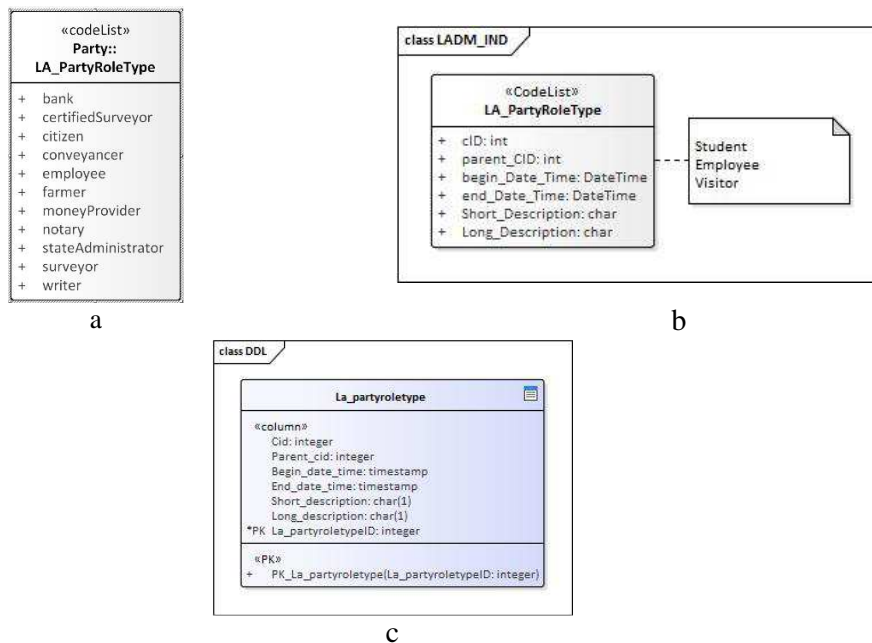
The spatial classes of the conceptual model such as LA\_SpatialUnit, LA\_Point, LA\_BoundaryFace, and LA\_BoundaryFaceString have a spatial data type for some of their attributes. For example GM\_Point, GM\_MultiSurface, and GM\_MultiCurve are spatial data types that are already predefined in the ISO standard; however, the software does not realize their type and select a "varchar" as a data type in table diagram. After the transformation, the software offers a list of spatial data types that could be used such as geometry, geometry collection, linestring, multilinestring, point, multipoint, polygon, and multipolygon. If the spatial data type does not include in the list, the only way is to define the type manually in the SQL code.

### 3.2.8 Code List Classes

The code List classes in the UML model contains the possible code list values for the attributes. The current class structure of the code list provides a better understanding of the conceptual model as shown in Figure 14a. However, for better semantic meaning during the transformation of the code list class into table diagram, the structure of the code list class has to be modified. Each code list should have a `cid`, `parent_cid`, `begin_date_Time`, `end_Date_Time` and `description` to allow for updating the value of the code list class (Kaloginni et al, 2017 and Kaloginni, 2016). (Zulkifile et al, 2015) has proposed that the implementation of each code list class will result in a single table. Also, the name of the class of the code list should have a “Type” after the name of the class. So, there are three steps to convert the class diagram of the code list into SQL:

- Convert the class diagram of the code list to intermediate conceptual model that contains the new structure as shown in Figure 14b.
- Transform the intermediate conceptual model of the code list class diagram into table diagram as shown in Figure 14c.
- And then, Generating the SQL code from the table diagram.

Additional to the attributes that proposed by (Kaloginni et al, 2017), the code list class of LADM-IndoorGML has `short_Description` and `long_description` attributes as shown in Figure 14c.



**Figure 14. The code list class in conceptual model, left shows the normal class structure, right shows the implementation structure for the code list.**

### 3.2.9 Indexing

The transformation from the UML diagram into DDL provide in an automated way the index for the PK and the FK. Each table will list all the PKs and FKs that related to the table or related to the association between the table and other tables. The transformation creates a B-

tree index for the FK as shown in Figure 15. Each table contains an index to store the FK that related to the table. Enterprise Architect software support only a spatial index (R-tree) that related to ArcGIS toolbox which in this it does not fit to this work. The solution is to define the spatial index in the SQL code manually.

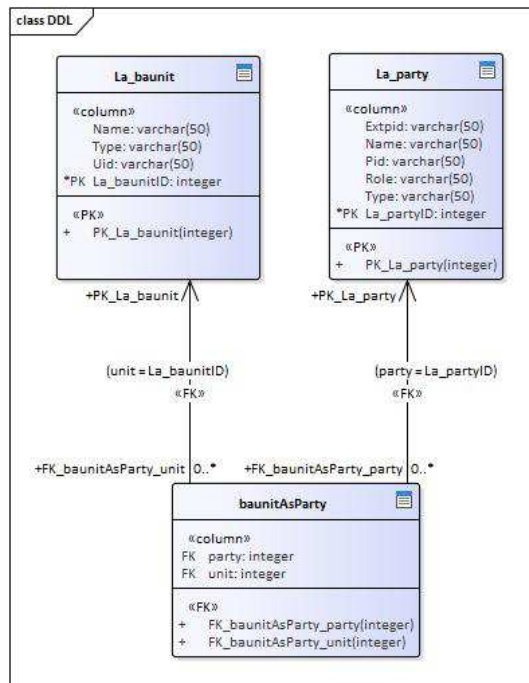


Figure 15. The index that the transformation provides in automated way

### 3.3 Generating SQL from UML Table Diagram

The code engineering in Enterprise Architect software is a tool that generates a SQL DDL from the UML table diagram in automated way. The code engineering converts all the tables for LADM-IndoorGML as shown in Figure 16. The SQL code contains the comment that the UML class includes next to each attribute.

```

DROP TABLE IF EXISTS La_party CASCADE
;
CREATE TABLE La_party
(
  Extpid varchar(50) NULL,      -- the identifier of the party in an external
  registration
  Name varchar(50) NULL,      -- the name of the party
  Pid integer NOT NULL,      -- the identifier of the party
  Role varchar(50) NULL,      -- the role of a party in the data update and maintenance
  process
  Type varchar(50) NULL      -- the type of the party
)
;

```

Figure 16. SQL code for creating LA\_Party table

Furthermore, the code engineering generates tables for Primary Keys and Indexes as shown in Figure 17 and Figure 18 show the SQL code for the code list tables.

```
CREATE INDEX IXFK_AddressLA_SpatialUnit_La_spatialunit ON AddressLA_SpatialUnit
(landParcel ASC)
;
CREATE INDEX IXFK_baunitAsParty_La_baunit ON baunitAsParty (unit ASC)
;
CREATE INDEX IXFK_baunitAsParty_La_party ON baunitAsParty (party ASC)
;
ALTER TABLE La_availabilitystatustype ADD CONSTRAINT PK_La_availabilitystatustype
PRIMARY KEY (La_availabilitystatustypeID)
;
ALTER TABLE La_baunit ADD CONSTRAINT PK_La_baunit
PRIMARY KEY (UId)
;
```

**Figure 17. SQL code for Primary Keys, and Indexes**

```
CREATE TABLE La_partyroletype
(
    Cid integer NULL,
    Parent_cid integer NULL,
    Begin_date_time timestamp without time zone NULL,
    End_date_time timestamp without time zone NULL,
    Short_description char(1) NULL,
    Long_description char(1) NULL,
    La_partyroletypeID integer NOT NULL
)
;
```

**Figure 18. SQL code for codelist table**

## 4 CONCLUSION

### 4.2 Main results

This paper has presented the derivation of the conceptual model of LADM-IndoorGML to technical model. The aim of the research is assessing the conceptual model to discover the possibilities and the shortcoming/limitations of the conceptual model. This paper shows that by using Enterprise Architect software, there are still many issues that need attention and manual adjustment. The UML model of the ISO TC211 has been used to prepare the LADM-IndoorGML class diagram which has to be created under new package. All the classes from different packages have to be included in the new package to ensure all the classes, and their associations are correctly transformed to table diagram. The transformation creates a new unique attribute that assigned as PK to all tables even if the class diagram has a unique ID attribute that was modeled in the UML class diagram. The solution of this issue is by correcting the PK for all tables manually. The additional PK also affect the associations between the tables and that also has to be modified manually. The multiplicity of the classes diagram has been changed in the table diagram, and in some cases, there are no solution to modify that. Also, the transformation does not cover the multiplicity that related to the attributes. The transformation creates a B-tree index in an automated way and to have a

spatial index need more research to find a solution. The spatial index has to be defined manually in the SQL code. The research shows that to convert the table diagram into SQL could be done automatically by using the code engineering tool of Enterprise Architect. The SQL code reflects the result of the UML table diagram, so, by avoiding all the limitations of the transformation the SQL DDL file will be correct to load into PostgreSQL. Considering the importance of these developments, we propose to include an annex to the new version of LADM standard to present the integrated model of IndoorGML and LADM which support the indoor navigation based on rights, restrictions, and responsibilities.

### 4.3 Future work

The future work will concentrate on running SQL code in PostgreSQL with PostGIS spatial extension to create the database and populating the database with data. Then a web user interface will be created by using WebGL technology that allows a complex interactive three-dimensional computer graphics visualisations from within web browsers. The development of the prototype will be used to refine the conceptual model. Different types of indoor environments and their users will be analysed in two iterations steps. The first iteration step starts with exploring the relationship between the indoor space and the user by examining an educational building to define the rights of use for the indoor spaces. According to the previous step, the Party package will be developed to produce an initial conceptual model. Based on the outcome of the user investigations, space subdivision (e.g. Diakit  and Zlatanova, 2018) will be performed to assess the accessibility of indoor spaces based on the RRRs. Next, the research will define the type of spaces according to the influence of RRRs. The second iteration will concentrate on different types of buildings to define the integration requirements for a generic conceptual model. Further, the conceptual model will be explored for additional applications such as crisis management and maintenance applications to define the access right of use based on the impact of these applications. Also, the impact of LOD will be studied to estimate the complexity of indoor subdivisions. Based on the new outcome the model will be converted from a conceptual model to technical model, and the web user interface will be developed to manage/edit the outcome data. It is essential to perform the first iteration step to ensure that the methodology is correct and to avoid any lack of the results.

## REFERENCES

- Alattas, A., Zlatanova, S., Van Oosterom, P., Chatzinikolaou, E., Lemmen, C., Li, K.-J. (2017). Supporting Indoor Navigation Using Access Rights to Spaces Based on Combined Use of IndoorGML and LADM Models. *ISPRS International Journal of Geo-Information*, 6(12), 384. Doi:10.3390/ijgi6120384
- Diakit  A. A. and S. Zlatanova (2018). Spatial subdivision of complex indoor environments for 3D indoor navigation, *International Journal of Geographical Information Science*, 32(2), 213-235.
- Dilo, A., Zlatanova, S. (2011). A data model for operational and situational information in emergency response: The Dutch case. *Applied geomatics*, 3(4), 207-218.

Hespanha J.P. (2012). Development Methodology for an Integrated Legal Cadastre (Deriving Portugal Country Model from the Land Administration Domain Model), PhD thesis, Delft University of Technology, pp. 343, 2012.

Hespanha J.P., Van Bennekom-Minnema, J., Van Oosterom, P.J.M., Lemmen, C.H.J. (2008). The model driven architecture applied to the Land Administration Domain Model version 1.1 with focus on constraints specified in the object constraint language, in Proceedings from FIG Working Week - Integrating Generations, Stockholm, Sweden. Federation International des Géomètres, June 2008, pp. 1-19.

Iacob, M-E., Steen, M.W., L. Heerink, L. (2008). Reusable model transformation patterns, in 12th Enterprise Distributed Object Computing Workshop, Munich, Germany, September 2008, p. 16.

Kalogianni, E., Dimopoulou, E., Quak, W., Germann, M., Jenni, L., Van Oosterom, P. (2017). INTERLIS Language for Modelling Legal 3D Spaces and Physical 3D Objects by Including Formalized Implementable Constraints and Meaningful Code Lists. *ISPRS International Journal of Geo-Information*, 6(12), 319. doi:10.3390/ijgi6100319

Kalogianni, E. (2016). Linking the Legal with the Physical Reality of 3D Objects in the Context of Land Administration Domain Model (LADM). Master's Thesis, Delft University of Technology, Delft, the Netherlands, 2016.

Lemmen, C.H.J., van Oosterom, P.J.M., Bennett, R. (2015). The Land Administration Domain Model. *Land Use Policy*; 2015, 49, pp. 535-545.

Van Bennekom-Minnema, J. (2008). The Land Administration Domain Model 'Survey Package' and Model Driven Architecture, Master's thesis, GIMA (TUD, UT-ITC, UU, WUR), pp. 199, 2008.

Zlatanova, S., Van Oosterom, P.J.M., Lee, J., Li, K.-J., Lemmen, C.H.J. (2016). LADM and IndoorGML for Support of Indoor Space Identification. In Proceedings of the 11<sup>th</sup> 3D Geoinfo Conference on ISPRS Annals of the photogrammetry, Remote Sensing and Spatial Information Science, Athens, Greece, 20–21 October 2016.

Zulkifli, N. A., Rahman, A. A., Jamil, H., Hua, T. C. (2014). Development of a Prototype for the Assessment of the Malaysian LADM Country Profile. In Abstract submitted to FIG Congress.

Zulkifli, N., Rahman, A., Jamil, H., Hua, T.C., Choon, T.L., Seng, L.K., Lim, C.K., Van Oosterom, P.J.M. (2015). Towards Malaysian LADM Country Profile for 2D and 3D Cadastral Registration System. In Proceedings of the 10<sup>th</sup> 3D GeoInfo Conference, Kuala Lumpur, Malaysia, 28–30 October 2015.

## BIOGRAPHICAL NOTES

**Abdullah Alattas** is a PhD candidate at the Section GIS-technology, Department of Research for the Built Environment (OTB), Faculty of Architecture and the Built Environment, Delft University of Technology (TU Delft). He is a lecturer at the Geomatics Department at the Faculty of Environmental Design, King Abdulaziz University in Jeddah, Saudi Arabia. In 2014 he obtained a master's degree in Cartography from the international Master program that is a cooperation of: the Technische Universität München (TUM), Department of Cartography, the Technische Universität Wien (TU Vienna), Research Group Cartography, and the Technische Universität Dresden (TU Dresden), Institute for Cartography. In 2008 he received a bachelor's degree in Architecture from the Faculty of Environmental Design, King Abdulaziz University in Jeddah, Saudi Arabia.

**Peter van Oosterom** obtained an MSc in Technical Computer Science in 1985 from Delft University of Technology, the Netherlands. In 1990 he received a PhD from Leiden University. From 1985 until 1995 he worked at the TNO-FEL laboratory in The Hague. From 1995 until 2000 he was senior information manager at the Dutch Cadastre, where he was involved in the renewal of the Cadastral (Geographic) database. Since 2000, he is professor at the Delft University of Technology (OTB institute) and head of the section 'GIS Technology'. He is the current chair of the FIG working group on '3D-Cadastrals'.

**Sisi Zlatanova** obtained her MSc in Geodesy, Photogrammetry and Cartography at the University of Architecture, Civil Engineering and Geodesy, Sofia, Bulgaria in 1984 and specialised Applied Mathematics at Technical University Sofia. She has received her PhD degree from Graz University of Technology, Austria in 2000. She worked as a software developer at Bulgarian Central Cadastre (1985 -1989), assistant professor at University of Architecture and Civil Engineering, Sofia (1989-1999) and associate professor at the Delft University of Technology (2000-2017). Since 2018 she is a professor at the University of New South Wales, Faculty of Built Environment, Sydney, Australia. She is the current president of ISPRS Technical Commission IV 'Spatial Information Science'.

## CONTACTS

Abdullah Alattas  
Delft University of Technology  
Faculty of Architecture and the Built Environment  
P.O. Box 5030  
2600 GA Delft  
THE NETHERLANDS  
Phone: +31 639898691  
E-mail: a.f.m.alattas@tudelft.nl

Peter van Oosterom  
Delft University of Technology  
Faculty of Architecture and the Built Environment  
P.O. Box 5030  
2600 GA Delft  
THE NETHERLANDS  
Phone: +31 15 2786950  
E-mail: P.J.M.vanOosterom@tudelft.nl  
Website: <http://www.gdmc.nl>

Sisi Zlatanova  
UNSW Built Environment  
Kensington Campus  
Sydney, NSW 2052  
AUSTRALIA  
Phone: +61 2 93856847  
E-mail: [s.zlatanova@unsw.edu.au](mailto:s.zlatanova@unsw.edu.au)  
Website <http://www.be.unsw.edu.au>



Abdullah Alattas, Peter van Oosterom and Sisi Zlatanova  
Deriving the Technical Model for the Indoor Navigation Prototype based on the Integration of IndoorGML and  
LADM Conceptual Model

7<sup>th</sup> International FIG Workshop on the Land Administration Domain Model  
11-13 April 2018, Zagreb, Croatia