3D pgRouting and Visualization in Cesium JS Using the Integrated Model of LADM and IndoorGML

Abdullah ALATTAS, Marian DE VRIES, Martijn MEIJERS, The Netherlands, and Sisi ZLATANOVA, Australia, Peter van OOSTEROM, The Netherland

Key words: LADM, IndoorGML, Indoor navigation, pgRouting, PostgreSQL

SUMMARY

A web-based application has been developed, exploiting the integrated model of LADM and IndoorGML to provide indoor navigation based on the user's access rights in an educational building. Different types of users (students, teachers, visitors, etc.) have different access rights, which also depend on the exact time (e.g. inside or outside office hours). A 3D BIM IFC file of a building has been geo-referenced and converted into a LADM complaint database in PostgreSQL/PostGIS and is enriched with information about access rights based on the relationship between users, time and indoor spaces. The PostgreSQL extension pgRouting has been used for the actual routing. To support the access rights-based routing, the database contains several tables to represent nodes, edges, parties (users), and rights. There is one overall network for the whole building, and database views are used to dynamically select the relevant nodes and edges based on the time and the user's rights. The Dijkstra algorithm is used to compute the shortest path. Finally, the 3D geospatial web-platform Cesium JS is used to create a client GUI allowing to specify start and destination, the user and time, and to visualize the navigation routes. As this GUI is web-based it can run on different platforms, such as desktops, laptops, tablets and mobile phones. This paper provides a complete description of all the steps to design, develop and test the integrated model of LADM and IndoorGML.

3D pgRouting and Visualization in Cesium JS Using the Integrated Model of LADM and IndoorGML

Abdullah ALATTAS, Marian DE VRIES, Martijn MEIJERS, The Netherlands, and Sisi ZLATANOVA, Australia, Peter van OOSTEROM, The Netherland

1 INTRODCTION

Current indoor environments can become enormously large and complex, affecting the users of the buildings during the navigation from one space to another. Therefore, many indoor applications have been developed to provide guidance while inside a building. Each indoor navigation application is based on a different approach and technology to provide the best routing experience.

Several researchers have studied and developed semantic data model for Indoor navigation application. A semantic model has been proposed in (Meijers et al., 2005); the model aims to simplify the evacuation routes' calculation during emergencies by providing exit possibilities based on the building characteristics such as available doors and windows. Becker et al 2009 proposed the multilayered space-event model for navigation in indoor spaces, which was included in the specifications of the international OGC standards IndoorGML (Lee et al 2014). The Indoor Navigation Space Model (INSM) has been proposed in (Liu, Zlatanova, 2012), is a semantic data model developed to derive the connectivity graph of a building in an automated way. According to the semantic of building, the model extends the categorization of the indoor spaces and can be utilized to define the nodes and edges of the connectivity graph. The OntoNav semantic model has been developed by (Tsetsos et a., 2005), and it is based on the building network geometry without considering the obstacles. The model is essentially user-centric, depending on the user's physical and perceptual capabilities and routing preferences. The research by (Puikkonen et al., 2009) has studied the UI requirements of the indoor navigation maps for mobile devices.

Instead of developing semantic models, some researchers have reported approaches for deriving sematic from semantically reach City Models and Building Information Models (BIM). Specific attention has been given to one of the BIM international standards, i.e. Industry Foundation Classes (IFC) (Liu et al 2021). A BIM model called BO-IDM has been developed by (Isikdag et al., 2013) to facilitate indoor navigation and orientation. The developed model applies the ISO, 19107 compliant data types for the transformation from the BIM models. The BO-IDM model produces precise semantic information for indoor navigation and describes the non-geo-referenced structure and complicated geometries of BIMs using ISO 19107 compliant descriptions.

Research has also concentrated on using only logical models, which omit geometrical references for nodes and edges. Navigation paths are then computed relying on sematic information. The Combinatorial Data Model (CDM) that has been proposed in (Lee 2004) describes building's adjacency, connectivity, and hierarchical relationships. It is a logical model that provides an indoor graph without the geometric features. An indoor navigation system based on a logical network, proposed in (Liu et al., 2019), relies on indoor spaces' semantics and characteristics instead of having notions for metrics. The system uses a space classification that shows spaces that are essential for navigation, such as horizontal and vertical orientations, doors, and windows.

Other researchers have focused on the subdivision of the indoor space. A conceptual model discussed in (Krūminaitė et al., 2014) provides an automated determination of the indoor space's function areas by utilizing the user behaviour and perception of the indoor environment and the subdivision of the space to create the navigation network. Diakite and Zlatanova 2018 presented a framework for classification of indoor spaces and identification of free of obstacles (static and dynamic) spaces to support automatic network derivation and user-tailored path computation.

Many indoor navigation systems have been developed which besides navigation path provide localization and user tracking. An Indoor navigation system that supports rescuers during emergencies by finding the shortest route has been developed by (Rueppel and Stuebbe, 2008). The system provides the rescuers with information related to their spatial context by utilizing wireless LAN, Ultra-wide Band (UWB), and Radio Frequency Identification (RFID). Additionally, the developed indoor navigation system presents floor plans on mobile devices from the existing 3D BIM models. A self-deployable solution for indoor navigation systems during an emergency has been proposed in (Renaudin et al., 2007). The indoor navigation system is based on Radio-frequency identification tags and inertial Micro Electromechanical Sensors to be estimated and matched with the building's pure inertial positioning system. An indoor navigation system called Near Field Communication (NFC) has been proposed in (Ozdenizci et al., 2011); the system aims to provide more reliable localization using NSF tags, placed inside the building. Such systems providing localization support have been focused on providing navigation to disabled users or users with special needs. A low-cost indoor navigation system for visually impaired, proposed by (Ivanov 2010), is based on mobile terminals, NFC and RFID tags. The application allows the users to imagine the space's map, such as the space dimensions and the relative location of interest points. Another low-cost system based on 3D models and UWB tags for localization and guidance of visually impaired people has been recently developed and tested by Benitez Sandoval et al 2020.

However, all of these applications do not consider the users' rights, restrictions, and responsibilities, and as a result they are providing the same routing options for all users. In reality, each user has a different relationship with the indoor environment. By not considering these differences, the indoor navigation application cannot provide the best route for the users based on their type. Therefore, the integrated model of Land Administration Domain Model

(LADM) and IndoorGML aims to define the users' rights, restrictions, and responsibilities at the time of the indoor navigation to provide an optimal route.

This paper presents a web-based application for 3D indoor navigation for an educational building based on the integrated model of ISO 19152 LADM and OGC IndoorGML. The International standards aims to ensure the presented solution works for any building in the world. The integrated model of LADM and IndoorGML allows to compute different routes with respect to the user's access rights such as student, visitor, maintenance, and teacher.

A 3D BIM/IFC file for an educational building has been used to derive the indoor network. The 3D model has been built in Revit software to extract the different types of information for each space, such as space name, space number, function, area, height, level, and space location (XYZ). A database has been created in PostgreSQL/PostGIS to store all the information and create the 3D graph using the pgRouting extension. Finally, the 3D geospatial web-platform 'Cesium JS' has been used to create the GUI and to visualize the 3D indoor navigation graph and the optimal path from start to destination. Using a web-based visualisation platform allows every user with a web browser on the laptop, tablet, or mobile to use the indoor routing application. The application provides a better understanding of the indoor environment during the navigation.

This paper is organized as follows: The integrated model based on ISO LADM and OGC IndoorGML standards is presented in section 2. Section 3 describes the creation of the 3D model. The development of the database is presented in Section 4. The client GUI and visualization of the routes in Cesium JS is represented in Section 5 and the paper ends with the conclusion.

2 THE INTEGRATED MODEL OF LADM AND INDOORGML

The conceptual model of LADM and IndoorGML defines the access rights of the indoor spaces for specific users. Each user creates an association with the indoor spaces based on the building's type and the space usage function. The indoor environment spaces have different functions, and that impact the accessibility rights of the users. The LADM ISO 19152 standard defines the user's access rights based on the relationship between the indoor spaces and the user. LADM defines the rights, restrictions, and responsibilities to all spaces to indicate the accessible spaces for each user type (Alattas et al., 2017).

The accessibility of indoor spaces of the building is influenced by assigning such rights, which provides additional information during computing the routes for the users of the indoor environment (Zlatanova et al 2016a). Therefore, the navigation process becomes more suitable and easier by avoiding all the non-accessible spaces based on the user's rights. Furthermore, the conceptual model provides additional information to the user's access rights by representing the access times to the building spaces. The user's motion in the indoor environment is affected by the accessibility Rights, Restrictions, and Responsibilities (RRR) of the spaces.

LADM and IndoorGML standards are linked by an association between the spatial features of indoor space (cell space) in the IndoorGML standard and the spatial unit in the LADM standard (Alattas et al 2017). By having the association, the space spatial information is collected from the cell space class in IndoorGML by the LA_SpatialUnit in LADM, and the LA_RRR/LA_BAUnit classes in the administrative package that contain the administrative (legal) information. The users' information is collected from an external database by the party package; see Figure 1.

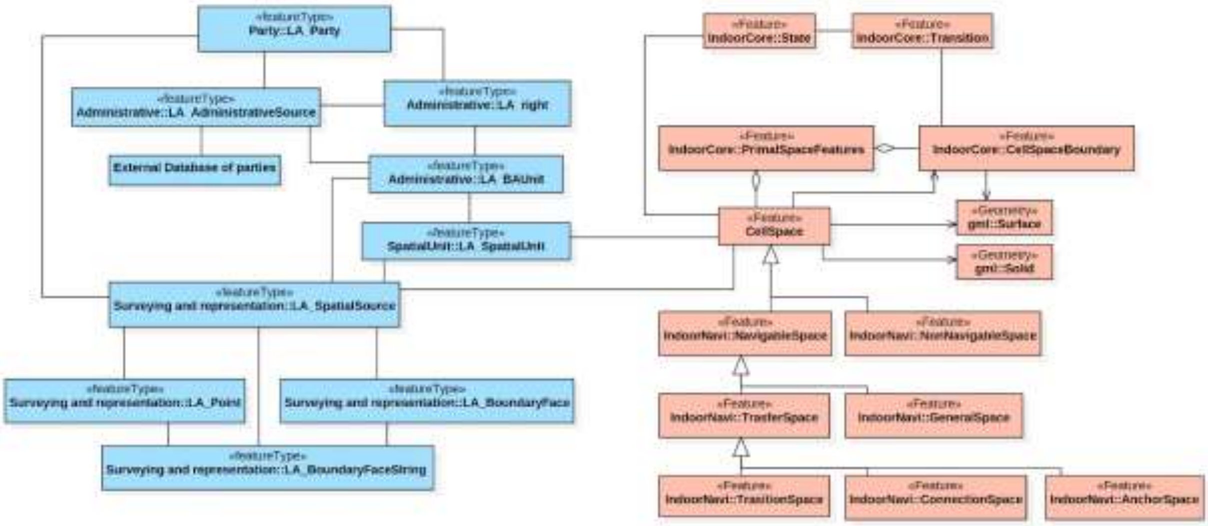


Figure 1. LADM-IndoorGML combined use model, the LADM classes are in blue and IndoorGML classes are in coral (Alattas et al., 2017)

The cell space class of IndoorGML represents all the types of spaces in the building. Each cell space has a unique ID and function type that is used during the navigation process. The unique ID and the type of function and cell geometry is provided by IndoorGML to LADM to define the user's navigable spaces according to the rights, restrictions, and responsibilities. Then, LADM classifies each GeneralSpaces and TransferSpaces into categories based on each cell's rights, restrictions, and responsibilities. Based on the LA_RightType attribute and the associated parties, LADM classifies the indoor spaces into different cell types with specific functional rights. There are two types of indoor spaces based on user access rights; (1) spaces that have private access rights and (2) spaces that have common access rights. The first category indicates that a particular individual or group of users has the right to access the cell, while the second category shows a common access right may exist between users.

IndoorGML is an OGC standard that provides a framework for Indoor Navigation by elaborating the indoor space and providing GML syntax for encoding the geoinformation (Lee et al, 2014). IndoorGML creates geometry, topology, and semantics models for the indoor spaces used for navigation network components. The standard provides the navigation network components based on the spatial required application for a specific indoor environment. IndoorGML is based on the concepts of primal and dual spaces.

The primal space subdivision is determined based on several characteristics of a building, such as (construction, operative use, security, or sensor coverage), or the user profile, such as walking or driving. The indoor space subdivision can be produced in a single space layer or multi-layered space model where the spaces from different layers could overlap, while the spaces from the same layer do not overlap. In addition, a particular theme space subdivision can either subdivide or combine several topographic units (rooms), such as security area, check-in area, or dangerous area, and the subdivision may change dynamically Figure 2.

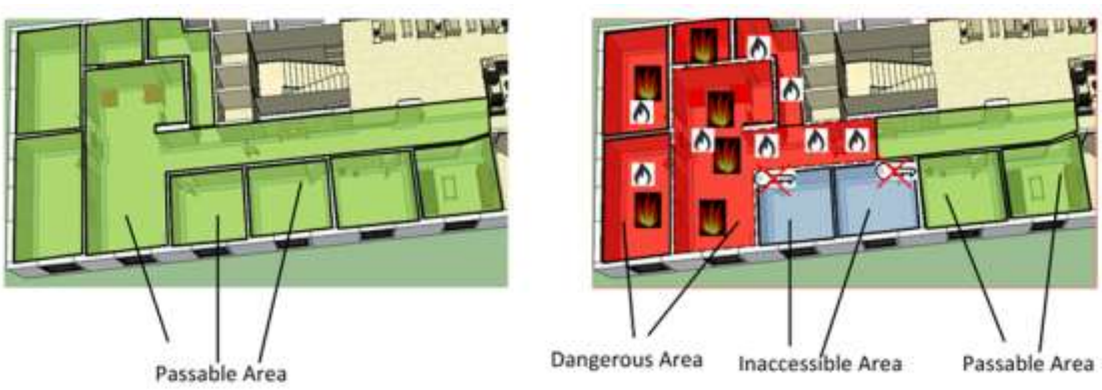


Figure 2: Union of topographic spaces to represent dangerous areas (Zlatanova et al 2016b)

The dual model is created from the primal model by applying the Poincaré duality (Munkres 1984). According to Poincaré duality, a k-dimensional object in N-dimensional (primal space) is transformed to (N – k) dimensional object in dual space. Therefore, the 3D spaces in the 3D primal space, such as rooms and corridors, are mapped into nodes (0D object) in dual space. At the same time, the 2D surfaces that are shared by the two solid objects are changed into an edge (1D) connecting the two nodes in the dual space (Munkres 1984). While the entire network is obtained from the primary space by utilizing only adjacency relationships, the connectivity between the spaces depends on semantics, such as the doors' notion. Figure 3 shows examples of dual spaces derived from the same primary space.

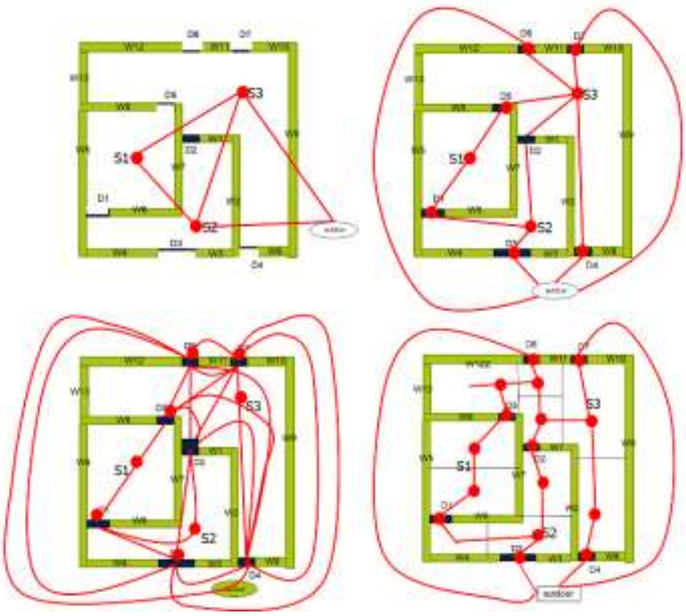


Figure 3: Examples of networks: (a) ‘thin’ doors which is a symbolically used; (b) ‘thick’ doors which are considered spaces itself; (c) only door spaces which are all the possible door-to-door connections; and (d) the subdivision of rooms in which each new space has only one door (Zlatanova et al 2016b)

3 THE CREATION AND PREPARATION OF THE 3D BIM/IFC MODEL

A 3D (BIM/IFC) model for the Faculty of Architecture and the Built Environment at TU Delft, Netherlands, is created to examine the integrated model of LADM and IndoorGML with real-world data. The Faculty of Architecture and the Built Environment consists of four floors and two sub-floors shared by four departments: Architecture (AR), Architecture Engineering and Technology (AET), Management in the Built Environment (MBE), and Urbanism (URB). Each department has supporting spaces, such as staff offices, secretary offices, meeting rooms, and copy and printing spaces. The first step to create the 3D model, is to obtain two types of information: the 2D floor plans of the building and the function information of each space. Therefore, the facility management department of TU Delft has been contacted to obtain the necessary information. By having the 2D floor plans of the building, the 3D model was created by using Autodesk Revit and Dinamo extension, as shown in Figure 4.

Figure 4: 3D model of the Faculty of Architecture and the Built Environment at TU Delft

3.1 Enriching with semantic/thematic data

After creating the 3D BIM/IFC model, the model was enriched by adding the semantic information to each space. The information received from the facility management department of TU Delft needed to be transformed and integrated into the model. This information was attached to the 3D spaces (ifcSpace) that differ from the physical elements of the building, such as walls, doors, or windows. Creating the 3D spaces for the entire building without having a table that contains the semantic information is difficult, mostly manual process. Therefore, a space table (with semantic information / thematic attributes for each room) was created by using the Dynamo extension (https://dynamobim.org). The table includes all the semantic information of the building, as shown in Figure 5. Then, the 3D spaces were created by using the Room function in Revit. During the creation of each 3D space, the correct information was attached to the space by selecting manually the space name from the table. Then, a table for the doors was created to provide information about the connectivity of the spaces (from space and to space), as shown in Figure 6. Figure 7 illustrates the difference between the physical model and the enriched model.

A	B	C	D	E	F	G
Number	Room Number	Building	Name	Space Name	Usage function	Type
1	08.01.01.010	08 Architecture and the Built Envir	01.Mid.010	BIBLIOTHEEK	A Education	A4 Study room / area
2	08.01.01.050	08 Architecture and the Built Envir	01.Mid.050		H Horizontal traffic	H1 Times
3	08.01.01.100	08 Architecture and the Built Envir	01.Mid.100	BIBLIOTHEEK	A Education	A4 Study room / area
4	08.01.01.801	08 Architecture and the Built Envir	01.Mid.801		H Horizontal traffic	H1 Times
5	08.01.01.802	08 Architecture and the Built Envir	01.Mid.802		H Horizontal traffic	H1 Times
6	08.01.01.803	08 Architecture and the Built Envir	01.Mid.803	BIBLIOTHEEK	A Education	A4 Study room / area
7	08.01.01.804	08 Architecture and the Built Envir	01.Mid.804		H Horizontal traffic	H1 Times
8	08.01.01.851	08 Architecture and the Built Envir	01.Mid.851		V Vertical traffic	V1 Stairs
9	08.01.01.853	08 Architecture and the Built Envir	01.Mid.853		V Vertical traffic	V2 Lift
10	08.01.01.854	08 Architecture and the Built Envir	01.Mid.854		V Vertical traffic	V1 Stairs

Figure 5: Space schedule for the 3D model

A	B	C	D	E	F
Mark	Level	From Room: Name	From Room: Room	To Room: Name	To Room: Room Number
3	1st floor	01.Oost.812	08.02.01.812	01.Oost.808	08.02.01.808
6	BG	BG.Mid.803	08.01.00.803		
8	BG	BG.Oost.600	08.02.00.600	BG.Oost.620	08.02.00.620
10	BG	BG.Oost.560	08.02.00.560	BG.Oost.600	08.02.00.600
11	BG	BG.Oost.808	08.02.00.808	BG.Oost.500	08.02.00.500
12	BG	BG.Oost.859	08.02.00.859	BG.Oost.859	08.02.00.859
13	BG	BG.Oost.808	08.02.00.808	BG.Oost.530	08.02.00.530
15	BG	BG.Oost.807	08.02.00.807	BG.Oost.808	08.02.00.808

Figure 6: Door table for the 3D model

Figure 7: The difference between the physical and the enriched BIM/IFC model

3.2 Extracting the location information (xyz) of the spaces and the doors

After enriching the 3D model with semantic information, the location (xyz) of each 3D space and door had to be extracted to create the nodes and edges of the dual space of IndoorGML. The location (xyz) of the 3D space was defined according to the position of the space during the creation. while the location of the door was selected to be in the center/middle of the door, as shown in Figure 8.

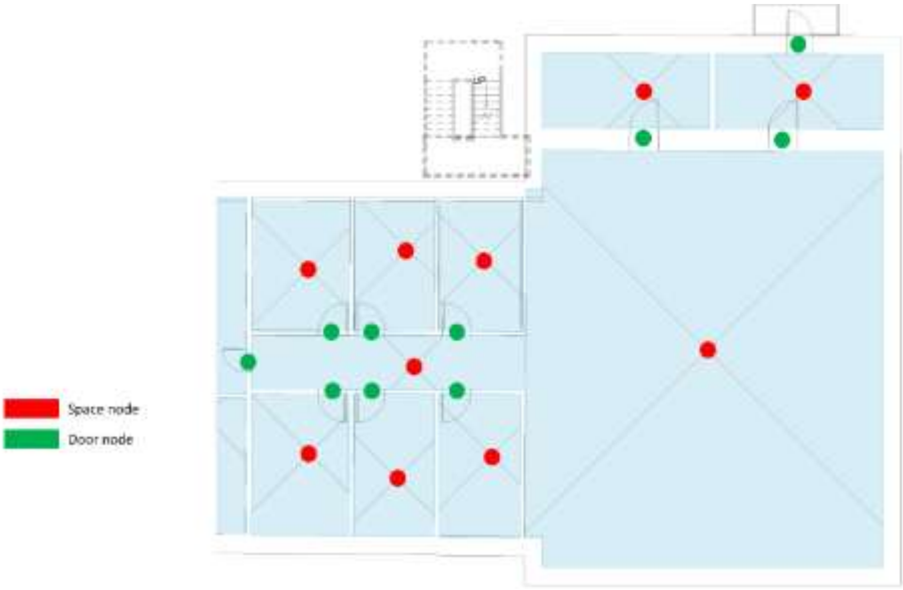


Figure 8: The location (xyz) of the spaces and doors

Extracting the location of any elements from the 3D model is not directly possible in Revit. Therefore, a process flow has been developed in Dynamo to extract the location (xyz). The process flow consists of several steps, where each step is responsible for a specific task. The first step of the process flow defines the target element, such as space or door. The second step is dedicated to the coordinate system's extraction. In this step, the project coordinate system is used to define the location of the element. By identifying the coordinate system and getting the element location, Dynamo transforms the geometry into x, y, z information according to the element location from the project coordinate point.

The third step provides information about the 3D space or the door, such as name and ID. The last step takes care of extracting the information to an Excel sheet. Figure 9 shows an example of the Dynamo process flow that has been used to extract the location (xyz) of the spaces. Finally, after extracting the location (xyz) of each space and door, the excel sheet contains all the required information to create the indoor navigation graph as shown in Figure 10.

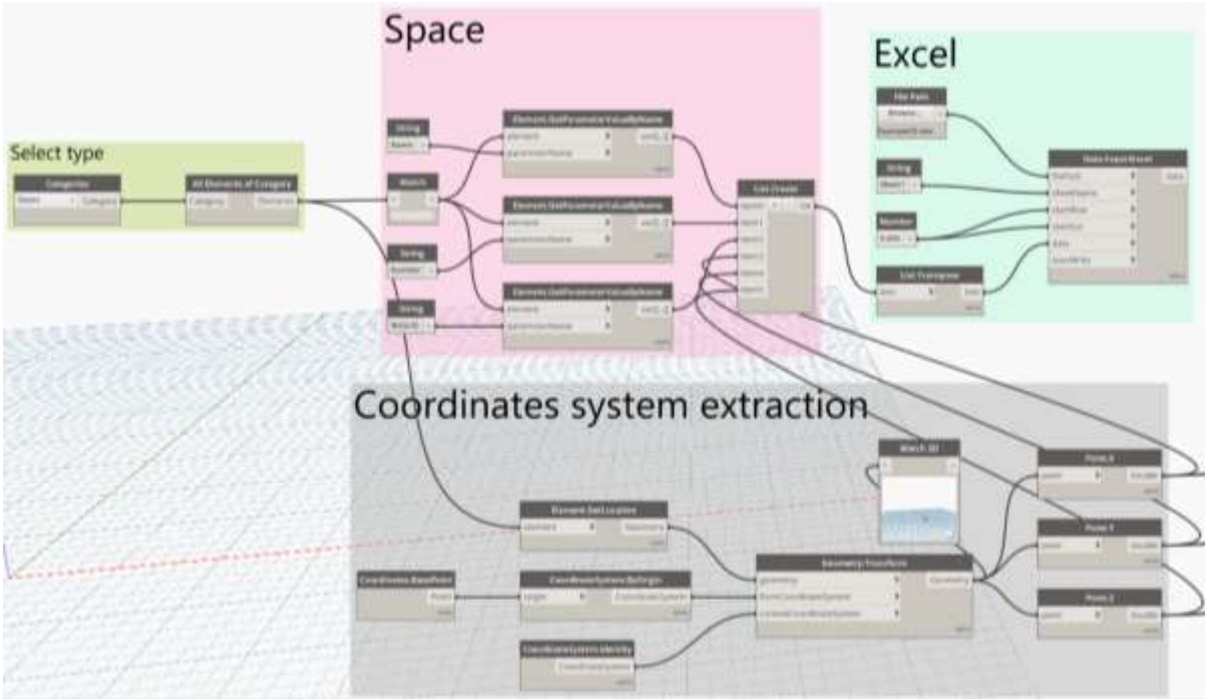


Figure 9: The Dynamo process flow that has been used to extract the location (xyz) of the spaces

	A	B	C	D	E	F	G	H	I	J
1	Id	RoomName	Usagefunction	Type	Area	Level	Volume	X	Y	Z
2	1	01.Mid.010	A Education	A4 Study room / area	190 m^2	1st floor	578.12 m^3	116.1477	32.92158	9.4
3	3	01.Mid.100	A Education	A4 Study room / area	196 m^2	1st floor	1124.14 m^3	156.0664	33.49046	9.4
4	4	01.Mid.801	H Horizontal traffic	H1 Times	74 m^2	1st floor	424.13 m^3	120.2766	39.22927	9.4
5	5	01.Mid.802	H Horizontal traffic	H1 Times	61 m^2	1st floor	187.21 m^3	133.5149	39.85491	9.4
6	6	01.Mid.803	A Education	A4 Study room / area	161 m^2	1st floor	694.34 m^3	139.1046	34.1417	9.4
7	7	01.Mid.804	H Horizontal traffic	H1 Times	74 m^2	1st floor	424.15 m^3	155.7065	39.10685	9.4
8	8	01.Mid.851	V Vertical traffic	V1 Stairs	7 m^2	1st floor	37.03 m^3	106.5713	35.98489	9.4
9	9	01.Mid.853	V Vertical traffic	V2 Lift	2 m^2	1st floor	6.70 m^3	129.5513	30.87287	9.4
10	10	01.Mid.854	V Vertical traffic	V1 Stairs	11 m^2	1st floor	32.04 m^3	130.1163	28.37873	9.4
11	11	01.Mid.855	V Vertical traffic	V1 Stairs	3 m^2	1st floor	19.11 m^3	143.3339	41.41551	9.4
12	12	01.Oost.010	A Education	A2-4 drawing room	103 m^2	1st floor	594.31 m^3	100.4978	82.25135	9.4
13	13	01.Oost.050	A Education	A4 Study room / area	83 m^2	1st floor	477.11 m^3	101.4705	72.1554	9.4
14	14	01.Oost.110	A Education	A4 Study room / area	81 m^2	1st floor	467.89 m^3	100.8792	60.05702	9.4
15	15	01.Oost.150	S Sanitary	S1 Toilet room	4 m^2	1st floor	25.05 m^3	97.93495	54.39457	9.4
16	16	01.Oost.170	S Sanitary	S1 Toilet room	33 m^2	1st floor	191.71 m^3	101.9991	53.23953	9.4
17	17	01.Oost.180	S Sanitary	S1 Toilet room	23 m^2	1st floor	69.55 m^3	102.2508	49.16784	9.4

Figure 10: The required information to create the indoor navigation graph

4 DEVELOPING THE DATABASE USING POSTGRESQL

Currently, the PostgreSQL database is considered one of the most advanced open-source DBMS. Therefore, PostgreSQL has been used to create a database for the integrated model of LADM and IndoorGML.

4.1 Creating the two tables Node (room) and Edge (network):

A database named BKNetwork was created within PostGIS, the spatial extension of PostgreSQL. Two tables to represent the navigation networks, Node and Edge, were created. Figure 11 shows the UML classes for the node and edge tables. The table Node has the following columns: nid, roomname, usagefunction, area, level, volume, x, y, z. The node table was populated by using the CSV file that was extracted from Revit. The table was cleaned by removing any rows that have the value of NULL in columns x, y, and z, as shown in Figure 12. The Edge table has the following columns: ID, nodefrom, tonode, as shown in Figure 13. The edge table was populated by using the CSV file that was extracted from Revit.

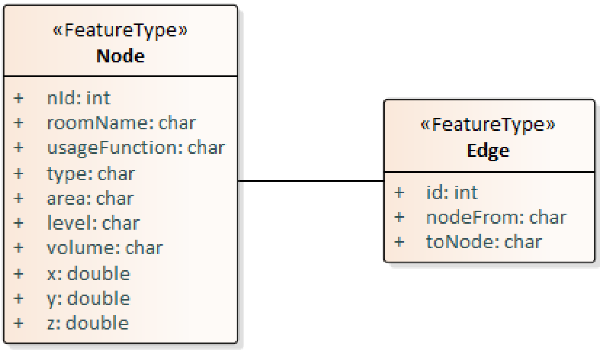


Figure 11: The UML classes for the node and edge tables

	nid integer	roomname character varying	usagefunction character varying	type character varying	area character varying	level character varying	volume character varying	x double precision	y double precision	z double prec
1	50	01.Oost.812	H Horizontal traffic	H1 Times	32 m^2	1st floor	97.71 m^3	53.2716301	27.31427367	9.4
2	51	01.Oost.841	H Horizontal traffic	H1 Times	2 m^2	1st floor	6.93 m^3	15.43888744	31.54224556	9.4
3	140	01.West.854	V Vertical traffic	V1 Stairs	36 m^2	1st floor	205.86 m^3	171.8130161	43.54197619	9.4
4	141	01.West.856	V Vertical traffic	V1 Stairs	8 m^2	1st floor	43.97 m^3	191.8416078	51.40384239	9.4
5	142	01.West.857	V Vertical traffic	V1 Stairs	8 m^2	1st floor	43.72 m^3	191.8949443	38.14777315	9.4
6	143	01.West.858	V Vertical traffic	V1 Stairs	13 m^2	1st floor	77.34 m^3	196.9903548	34.78826502	9.4
7	144	01.West.859	V Vertical traffic	V2 Lift	7 m^2	1st floor	42.07 m^3	169.8503513	35.69532569	9.4

Figure 12: Part of the **node** table

	id integer	nodefrom character varying	tonode character varying
1	1	01.Oost.812	D1
2	132	Room 6	D132
3	317	01.Oost.812	D317
4	474	01.West.806	D474
5	790	D34	Room 3
6	791	D35	Room 3
7	1262	D506	01+.West.410

Figure 13: The edge table

4.2 Creating the geometry of the nodes and edges

In this phase, the table Node contained only alpha numerical information (including the separate x, y, and z attributes). Therefore, an additional column had to be created to store the node's geometry for each space. The ‘geom’ column was added and populated by using the function ST_MakePoint(x,y,z) for each space.

By having the geometry of the nodes in the node table, the edge_geom_view was created to store the geometry of the edges between the two spaces *nodefrom* and *tonode*. Figure 14 shows the concept of creating the edges between the nodes of the spaces and doors. The edge_geom_view was created by using the following function:

```
Create view edge_geom_vw as
select e.*, ST_Makeline (fr.geom, tt.geom) as geom
from edge e, node fr, node tt
where fr.roomname = e.nodefrom and tt.roomname = e.tonode;
```

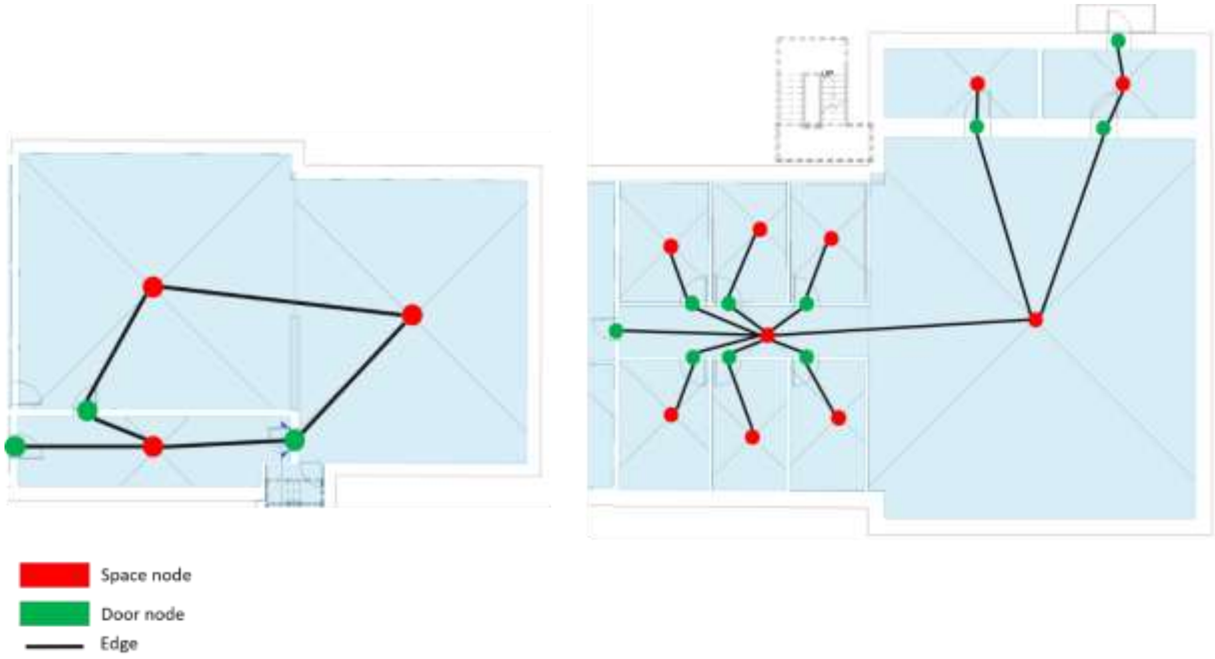


Figure 14: The concept of creating the edges between the nodes

4.3 Using pgRouting extension to create the indoor navigation routes

The pgRouting is an extension for the PostgreSQL database that provides routing functionality for indoor and outdoor. However, pgRouting does not support creating the network topology for 3D geometries (nodes and edges). It is only creating the topology for 2D geometries. Therefore, by having the nodefrom and tonode and geom columns in the edge_geom_view, there is no need to use pgRouting to create the topology. The edge_geom_view already contains the network between the spaces.

4.4 The cost, source, and target columns

In this step, the pgRouting extension must be added to the database to include additional columns to the edge_geom_view to compute the navigation routes. The first column is the cost, and we use the length of the edge by using the function ST_3DLength(geom). Then, two additional columns were added to the edge_geom_view: *source* and *target*. The two columns have the value of the node ID of the *nodefrom* and *tonode* columns, as shown in Figure 15.

Data Output Explain Messages Notifications

	id integer	nodefrom character varying	tonode character varying	geom text	cost double precision	source integer	target integer
1	1	01.Oost.812	D1	LINESTRING Z (53.2716301 27.31427367 9.4,52.65579435 15.35370002 9.28)	11.9764174822045	50	728
2	2	BG.Mid.803	D2	LINESTRING Z (137.4976237 30.88644468 3.7,136.5424089 28.34497224 3.7)	2.71505386271038	303	729
3	3	BG.Oost.600	D3	LINESTRING Z (9.193643505 8.50922468 3.7,5.644498332 7.123743106 3.7)	3.80998564970037	342	730
4	4	BG.Oost.560	D4	LINESTRING Z (26.15657016 6.573066581 3.7,12.14231164 11.75213829 3.7)	14.9406233347995	341	731
5	5	BG.Oost.808	D5	LINESTRING Z (37.74012175 10.05853868 3.7,36.27501858 8.426110472 3.7)	2.19347877879366	353	732

Figure 15: The cost, source, and target columns for the edge view (edge_wv)

4.5 Creating the users and rights tables

Based on the integrated model of LADM and IndoorGML, the LADM defines the user access rights based on the user relationship to the indoor environment. Therefore, the LA_Party and LA_Right tables were created. Figure 16 shows the UML classes for the LA_Party and LA_Right tables. The LA_Party table includes the user ID (pID), name and the type of the user, as shown in Figure 17.

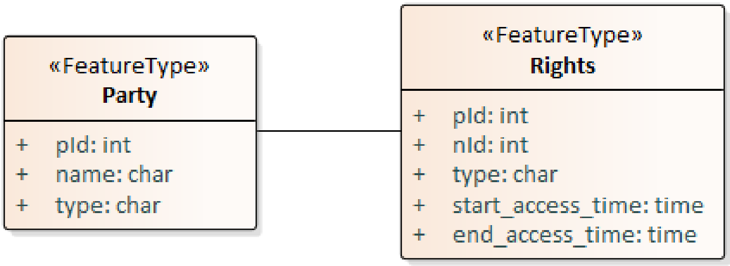


Figure 16: Shows the UML classes for the LA_Party and LA_Rights tables

	pid integer	name character varying	type character varying
1	1	Liam	student
2	2	Noah	student
3	3	Oliver	student
4	4	William	teacher
5	5	Elijah	teacher
6	6	James	teacher
7	7	Benjamin	maintenance
8	8	Lucas	maintenance
9	9	Mason	visitor
10	10	Ethan	student

Figure 17: The LA_Party table

According to the user access rights, the right table was created. The LA_Right table contains user ID (pID), node ID (nID), access type (type), start_access_time, and end_access_time. The LA_Right table has all the nodes ID where the users can access, as shown in Figure 18.

Data Output | Explain | Messages | Notifications

	pid integer	nid integer	type character varying	start_access_time time without time zone	end_access_time time without time zone
1	1	50	access	07:00:00	19:00:00
2	1	51	access	07:00:00	19:00:00
3	1	140	access	07:00:00	19:00:00
4	1	141	access	07:00:00	19:00:00
5	1	142	access	07:00:00	19:00:00
6	1	143	access	07:00:00	19:00:00

Figure 18: The LA_Right table

4.6 Creating node and edge views based on the rights of user at given time

After creating the party table and LA_Right table, dynamic views for the node and edge tables can be created and they use the accessible spaces for a specific user (and at a specific time). Table 1 shows the total number of nodes and edge according to user's access rights. It is clearly visible that the teachers have access to most of the spaces. While the maintenance staff members have access rights to the basement and the corridors of the building only.

Table 1: Number of accessible spaces for each type of the users according to their rights.

Users type	Node	Edge
Total	1394	2992
Student	860	1646
Teacher	906	1720
Maintenance	528	696
Visitor	324	442

The following queries have been used to create the node view and the edge view for a student **Liam** at **14:00:00** hours. Note that in the actual scripts to create the views we use parameters, and the values are obtained from the client GUI. Figure 19 shows the content node_vw view, and Figure 20 shows the edge_vw views for the student, together forming the (sub)network, which is accessible for Liam at 14:00:00 hours.

```
create view node_vw as
select node.nid, node.geom
from node n, LA_rights r, LA_party p
```

```
where p.name= 'Liam' and r.pid=p.pid and r.type= 'access' and r.nid=node.nid
    and    r.start_access_time    <=    '14:00:00'    and    '14:00:00'    <=
r.end_access_time;

create view edge_vw as
select edge_geom_cost_vw.*
from edge_geom_cost_vw, node_vw nf, node_vw nt
where nf.nid=edge_geom_cost_vw.source and nt.nid=edge_geom_cost_vw.target;
```

	nid integer	geom geometry
1	50	01010000A0110F0000FB0D6EC6C4A2...
2	51	01010000A0110F0000AFD9C2DAB5E...
3	140	01010000A0110F0000E313573A047A...
4	141	01010000A0110F0000DF217B73EEFA...
5	142	01010000A0110F0000BB873A62A3FC...
6	143	01010000A0110F0000F9AD8CFCBD9F...
7	144	01010000A0110F00008EF3ED13363B...

Figure 19: The node view for student

Data Output Explain Messages Notifications

	id integer	fromroomname character varying	toroomname character varying	nodefrom geometry	nodeto geometry	geom geometry	cost double precision	source integer	target integer
1	1377	D621	01.Oost.812	01010000A0110...	01010000A0110...	01020000A0110...	9.64461422973641	1348	50
2	1378	D622	01.Oost.812	01010000A0110...	01010000A0110...	01020000A0110...	5.82067507391789	1349	50
3	1379	D623	01.Oost.812	01010000A0110...	01010000A0110...	01020000A0110...	1.20696761190692	1350	50
4	1071	D315	01.Oost.812	01010000A0110...	01010000A0110...	01020000A0110...	6.46242262927356	1042	50
5	2584	D316	01.Oost.812	01010000A0110...	01010000A0110...	01020000A0110...	5.62824864258055	1043	50

Figure 20: The edge view for a student

4.7 Computing the shortest path

The Dijkstra algorithm has been used to compute the shortest path for the users of the building in pgRouting. The node and edge views of the user have been used to compute the indoor navigation routes as follows:

```
SELECT X.seq, Y.nid, Z.geom, X.Path_seq, X.edge, X.cost, X.agg_cost
FROM
    pgr_dijkstra(
       'select id, source, target, cost from edge_vw',
      (select nid from node_vw where roomname ='BG.Mid.802' limit 1),
      (select nid from node_vw where roomname ='BG+.West.310' limit 1),
        FALSE
    ) AS X
      INNER JOIN
    node_vw AS Y ON X.node = Y.nid
      LEFT JOIN
    edge_vw AS Z ON X.edge = Z.id
ORDER BY seq;
```

Figure 21 shows the path segments from the starting point (the main entrance of the building) to the destination point (office 'BG+.West.310'). Note, start and destination are parameters in the actual SQL scripts, as well as values, which are set via the client GUI.

Data Output | Explain | Messages | Notifications

	seq integer	nid integer	geom geometry	path_seq integer	edge bigint	cost double precision	agg_cost double precision
1	1	302	01020000A0110...	1	1428	7.45921426018157	0
2	2	1399	01020000A0110...	2	2940	13.9334386831176	7.45921426018157
3	3	304	01020000A0110...	3	2339	10.0036920210965	21.3926529432991
4	4	798	01020000A0110...	4	827	5.88636732429947	31.3963449643957
5	5	436	01020000A0110...	5	2939	2.38394091739378	37.2827122886951
6	6	1398	01020000A0110...	6	1427	8.77123133838234	39.6666532060889
7	7	435	01020000A0110...	7	2413	6.0657029080525	48.4378845444713
8	8	872	01020000A0110...	8	901	10.5276410667169	54.5035874525237
9	9	434	01020000A0110...	9	2910	9.97536212444123	65.0312285192406
10	10	1369	01020000A0110...	10	1398	2.13231369360411	75.0065906436819
11	11	449	01020000A0110...	11	1416	1.92293623085794	77.138904337286
12	12	1387	01020000A0110...	12	2928	2.32544799273016	79.0618405681439
13	13	530	01020000A0110...	13	62	1.98794009252385	81.3872885608741
14	14	789	01020000A0110...	14	1574	1.80718695918744	83.3752286533979
15	15	532		15	-1	0	85.1824156125854

Figure 21: The path segments for a student

5 VISUALIZATION OF THE ROUTES IN CESIUM JS

The routing with taking into account the access rights inside buildings of the different users (at different times) is exposed via a web-based 3D GUI. In this way end-users can use the functionality via their laptop, tablet or mobile phone web-browser via an easy to use interface where the start and destination are specified together with user (and time). At the server-side there is the PostgreSQL/PostGIS database (with pgRouting functionality), GeoServer implementing industry standard OGC protocols such as Web Feature Service (WFS), Web Map Service (WMS), and Web Coverage Service (WCS), and Apache Tomcat as webserver. Figure 22 shows the system architecture of the system at http://pakhuis.tudelft.nl:8080/edu/cesium74/Apps/routing/.

The actual background images are in jpg form retrieved from another server (virtualearth.net) with HTTP requests such as:

```
GET http://ecn.t1.tiles.virtualearth.net/tiles/a1.jpeg?n=z&g=10184
```

The 3D building model is represented in a number of b3dm tiles with the geometry of the spaces. They are obtained by the client by the HTTP GET request to our pakhuis server:

```
http://pakhuis.tudelft.nl:8080/edu/cesium74/Apps/SampleData/cadastral3/tile
set7/data/data0.b3dm
```

Instead of using local coordinates, the building geometry has been transformed to geographic coordinates, which enables combinations with other (background) geographic information. As part of the geo-reference there is a coordinate transformation from our RD+NAP (EPSG 7415) to ECEF (Earth Centered, Earth Fixed, which s EPSG:4978) needed for Cesium JS.

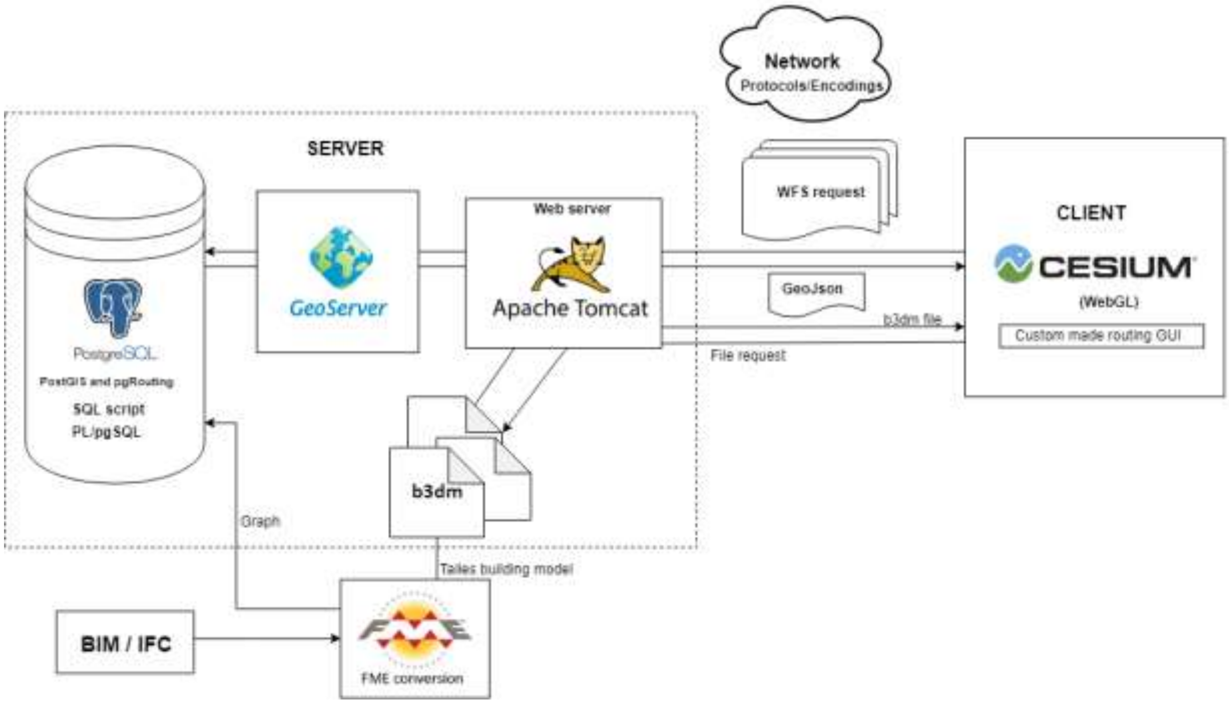


Figure 22: The system architecture

5.1 Visualize & interact with the 3D building model

The Cesium JS based 3D web-client allows basic visualization and interaction functionality out of the box: A complete globe with earth imagery, zooming, panning (rotating over the Earth globe), adding additional map layers, hover-over (showing name), selecting a specific object (and showing more attributes), searching for specific locations via gazetteer (and fly to that location), etc. Cesium JS is using WebGL for high performance rendering by using the GPU at client side (when available).

JavaScript at the client side may be added to provide additional functionality. In our case a way to specify the user, the start and the target rooms, and to initiate the path computation and visualization. As stated, Cesium JS allows retrieving addition (3D) map layers and the best format for this is b3dm tiles, though other formats are supported, such as KML, but in our practice often leading to performance issues and/or bugs. “Batched 3D Model allows offline batching of heterogeneous 3D models, such as different buildings in a city, for efficient streaming to a web client for rendering and interaction. Efficiency comes from transferring multiple models in a single request and rendering them in the least number of WebGL draw calls necessary” (3D Tiles Specification, 2019).

A Cesium tile set consists of a tileset.json and b3dm with the geometry and other attributes. The tileset.json contains bounding boxes and other meta information, and has the urls of the 'tiles', in our case: b3dm files. B3dm files use gltf encoding for the geometry plus a 'batch table' that makes it possible to include non-geo attributes in the files. The 3D model of the Bouwkunde building is stored in b3dm files (these are the 'tiles' of the tileset) that are on the file system of the server.

5.2 Connect GUI/Cesium JS client to DBMS/web-server, call shortest path

Computing the route is done in the database, by sending to the GeoServer a HTTP GET (WFS) request. For producing the response, the 'SQL View' possibility of GeoServer (extension of standard WFS) is used to define not the normal or typical database table as WFS layer, but a parameterized 'SQL View'; see Figure 23. When the maplayer **`bkpath_party`** is requested via WFS to GeoServer at the serverside, this results in the execution of the PL/pgSQL function **`frontoname_party`** with three parameters (party_name, source_room, target_room), which defines the proper SQL views, calls the dijkstra pgRouting and returns the resulting database table as new/dynamic map layer.

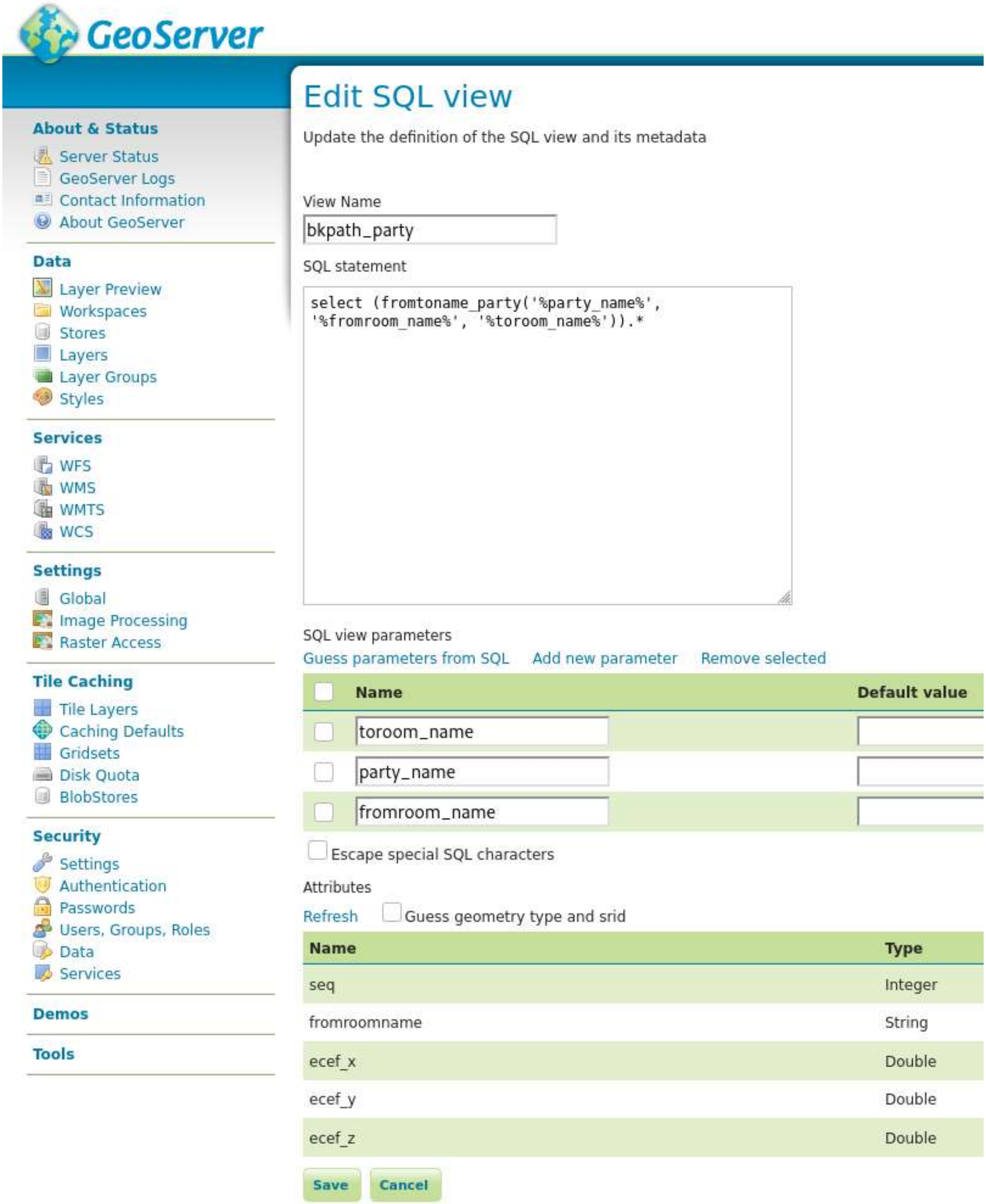


Figure 23: Definition of SQL view 'bkpath_party' (with three parameters: party_name, source_room, target_room) based on call of PL/pgSQL function 'fromtoname_party'

Below the PL/pgSQL code is shown that drops the old views (previous users), created news views (based on rights of current users), and starts the Dijkstra routing. The returned table contains the computed route: room identifiers and the node geometry.

```
CREATE OR REPLACE FUNCTION frontoname_party(
    party_name character varying,
    source_room character varying,
    target_room character varying)
RETURNS
    TABLE(room_id integer, roomname character varying,
          geom geometry, seq integer)
as
$$
declare
     source_id int;
     target_id int;
begin
    execute 'drop view if exists node_vw cascade';
    execute 'drop view if exists edge_vw cascade';

    execute 'create view node_vw as
    select room.id, room.roomname from room, rights, party where party.name
= ' || quote_literal(party_name) || ' and rights.pid = party.pid and
rights.type= ' || quote_literal('access') || ' and rights.nid=room.id';

    execute 'create view edge_vw as select edge_geom_cost_vw.*  from
edge_geom_cost_vw, node_vw nf, node_vw nt
where nf.nid=edge_geom_cost_vw.source and nt.nid=edge_geom_cost_vw.target';

    execute 'select id from node_vw where node_vw.roomname = '||
quote_literal(source_room) into source_id;
    execute 'select id from node_vw where node_vw.roomname = '||
quote_literal(target_room) into target_id;

    return query
        SELECT
            X.Path_seq, X.edge, X.seq, Y.nid, Z.geom
        FROM
            pgr_dijkstra(
               'select id, source, target, cost from edge_vw',
         source_id,
         target_id,
               FALSE
            ) AS X
         INNER JOIN
            node_vw AS Y ON X.node = Y.nid
         LEFT JOIN
            edge_vw AS Z ON X.edge = Z.id
        ORDER BY seq;
end;
$$ language plpgsql;
```

5.3 The navigation GUI

Using at the client-side JavaScript code the user is provided with a GUI sidebar added to Cesium JS (Figure 24), that contains setting transparency of walls, elevating the building, switching on/off the different floors (currently hardcoded for the example building, in the future a generic list of floors), specifying the user (group); e.g. staff of student, entering the names of the start and the target rooms, and finding the rout. The result shows the shortest path as line (animated with movement) and highlights the relevant spaces.

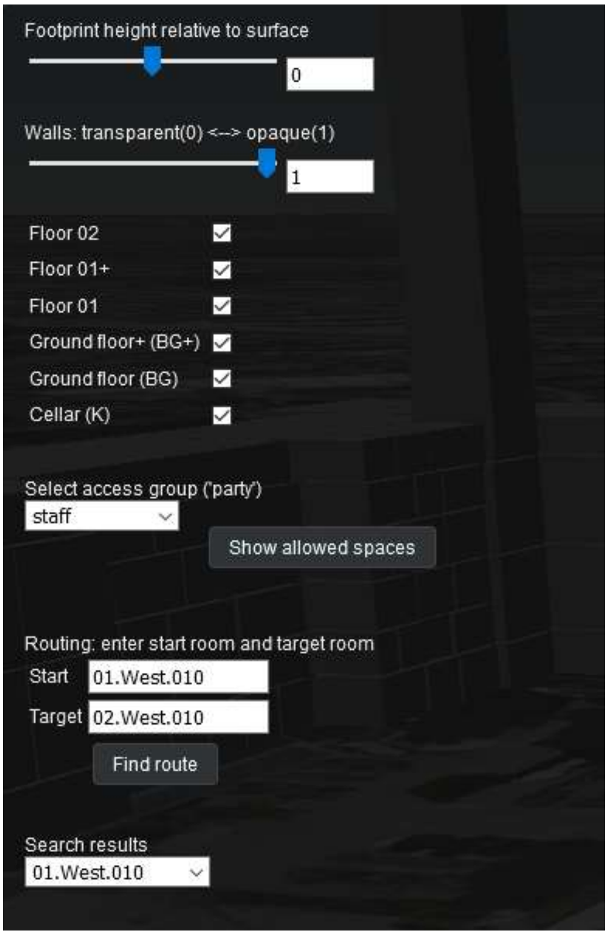


Figure 24: The sidebar that can be modified for the navigation routes visualization

Before starting the actual routing, it is possible to check and visualize which rooms/spaces are accessible by a specific user (e.g. a student) with the following HTTP GET request:

```
http://pakhuis.tudelft.nl:8088/geoserver/nl/ows?service=WFS&version=1.0.0&request=GetFeature&outputFormat=application/json&typeName=nl:allowed_rooms&cql_filter=partyname='student'”
```

This request is constructed and sent to a WFS service (in our case GeoServer WFS implementation) that passes on the query to the database (in our case PostgreSQL). The response contains the map layer **nl:allowed_rooms** encoded in GeoJSON (in total 859 rooms/spaces):


```
{
      "type": "FeatureCollection",
      "totalFeatures": 859,
      "features": [
            {
                  "type": "Feature",
                  "id": "allowed_rooms.fid--7290fc9f_177e56c1fc7_7535",
                  "geometry": {
                        "type": "Point",
                        "coordinates": [
                              3922993.57039708,
                              299849.38535691,
                              5003272.91050721
                        ]
                  },
                  "geometry_name": "geom_ecef",
                  "properties": {
                        "id": 740,
                        "roomname": "D13",
                        "ecef_x": 3922993.570397083,
                        "ecef_y": 299849.3853569102,
                        "ecef_z": 5003272.910507206,
                        "partyname": "student"
                  }
            },
…..
            {
                  "type": "Feature",
                  "id": "allowed_rooms.fid--7290fc9f_177e56c1fc7_788f",
                  "geometry": {
                        "type": "Point",
                        "coordinates": [
                              3923049.68503034,
                              299841.28404792,
                              5003236.92225268
                        ]
                  },
                  "geometry_name": "geom_ecef",
                  "properties": {
                        "id": 1457,
                        "roomname": "D730",
                        "ecef_x": 3923049.685030338,
                        "ecef_y": 299841.2840479159,
                        "ecef_z": 5003236.922252683,
                        "partyname": "student"
                  }
            }
      ],
      "crs": {
```

```
                "type": "name",
                "properties": {
                        "name": "urn:ogc:def:crs:EPSG::404000"
                }
        }
}
```


The response is processed in the Cesium JS client. In this case to show the rooms allowed for a student. The 'match' between what comes back from the database via WFS and the units (rooms, parcels) in the b3dm model is done by looping over the batch tables of the b3dm tiles and finding the match between identifier attribute in the b3dm batch tables ('LongName') and the identifier attribute 'roomname' in the database response. These spaces are highlighted.

The routing is started by sending the HTTP GET WFS request of the map layer nl:bkpath_party as follows:

```
http://pakhuis.tudelft.nl:8088/geoserver/nl/ows?service=WFS&version=1.0.0&request=GetFeature&outputFormat=application/json&typeName=nl:bkpath_party&viewparams=party_name:student;fromroom_name:01.West.010;toroom_name:02.West.010
```

When this routing fails, as no path exists, we get back a GeoJSON response without features:


```
{
        "type": "FeatureCollection",
        "totalFeatures": 0,
        "features": [],
        "crs": null
}
```


indicating that there is no allowed route for a student to get from room 01.West.010 to room 02.West.010.

The same HTTP GET WFS request but now for a staff member:

```
http://pakhuis.tudelft.nl:8088/geoserver/nl/ows?service=WFS&version=1.0.0&request=GetFeature&outputFormat=application/json&typeName=nl:bkpath_party&viewparams=party_name:staff;fromroom_name:01.West.010;toroom_name:02.West.010
```

This gives a map layer containing the route encoded in the following GeoJSON result (with 28 nodes in total):

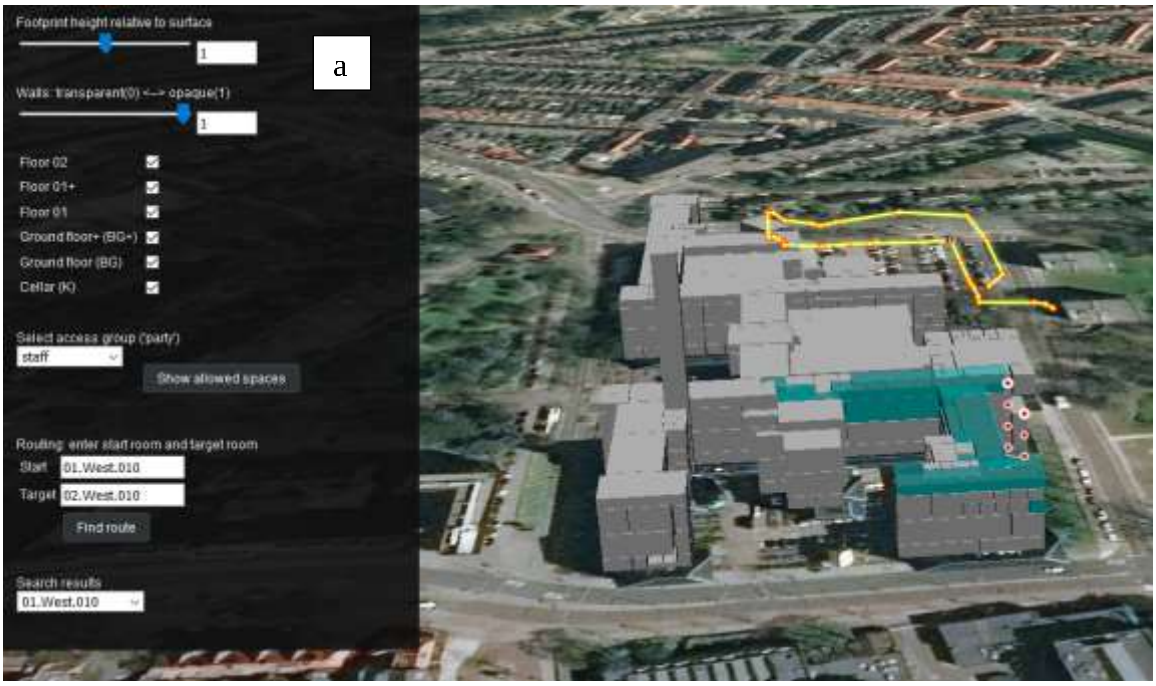

```
{
        "type": "FeatureCollection",
        "totalFeatures": 28,
        "features": [
                {
                        "type": "Feature",
                        "id": "bkpath_party.fid--7290fc9f_177e56c1fc7_74fd",
                        "geometry": null,
                        "properties": {
                                "seq": 0,
                                "fromroomname": "01.West.010",
                                "ecef_x": 3923139.967637715,
                                "ecef_y": 299826.58231767156,
                                "ecef_z": 5003167.481330292
                        }
                },
                {
```

```
                "type": "Feature",
                "id": "bkpath_party.fid--7290fc9f_177e56c1fc7_74fe",
                "geometry": null,
                "properties": {
                      "seq": 1,
                      "fromroomname": "D446",
                      "ecef_x": 3923137.5399089674,
                      "ecef_y": 299827.24866635184,
                      "ecef_z": 5003169.33252207
                }
          },
          {
                "type": "Feature",
                "id": "bkpath_party.fid--7290fc9f_177e56c1fc7_74ff",
                "geometry": null,
                "properties": {
                      "seq": 2,
                      "fromroomname": "01.West.801",
                      "ecef_x": 3923135.6947698826,
                      "ecef_y": 299826.22960566,
                      "ecef_z": 5003170.830278884
                }
          },
          ......
          {
                "type": "Feature",
                "id": "bkpath_party.fid--7290fc9f_177e56c1fc7_7517",
                "geometry": null,
                "properties": {
                      "seq": 26,
                      "fromroomname": "D567",
                      "ecef_x": 3923135.3362143203,
                      "ecef_y": 299822.50364835275,
                      "ecef_z": 5003178.817982247
                }
          },
          {
                "type": "Feature",
                "id": "bkpath_party.fid--7290fc9f_177e56c1fc7_7518",
                "geometry": null,
                "properties": {
                      "seq": 27,
                      "fromroomname": "02.West.010",
                      "ecef_x": 3923138.395770645,
                      "ecef_y": 299815.0269162718,
                      "ecef_z": 5003176.880054299
                }
          }
    ],
    "crs": null
}
```

The returned route is illustrated by highlighting the spaces that are passed and by drawing a path which can be visualized with the Cesium JS animation options. This can be (re)started by pusing the 'play' button, in the lower left corner of the GUI.

5.4 Routing example with visualization

In this subsection a routing example with visualization is given: same time, same source, same target, *different person* (example from section 5.3). The route in Cesium JS is not drawn by using the edge geometry, but by using a Cesium JS functionality to interpolate positions between given points (in our case the network nodes). This works much faster than adding linestring entities to Cesium (then Cesium uses the Entity API). The interpolation functionality also makes the animation possible: letting a symbol 'walk' along the interpolated route. The following example in Figure 25 shows the route for a staff member and a student for the same start and destination. While, the staff member has a route as shown in Figure 25 top, the student, who does not have access rights for the same destination, obtains no route as shown in Figure 25 bottom (note the ‘No results’ waring in the interface).



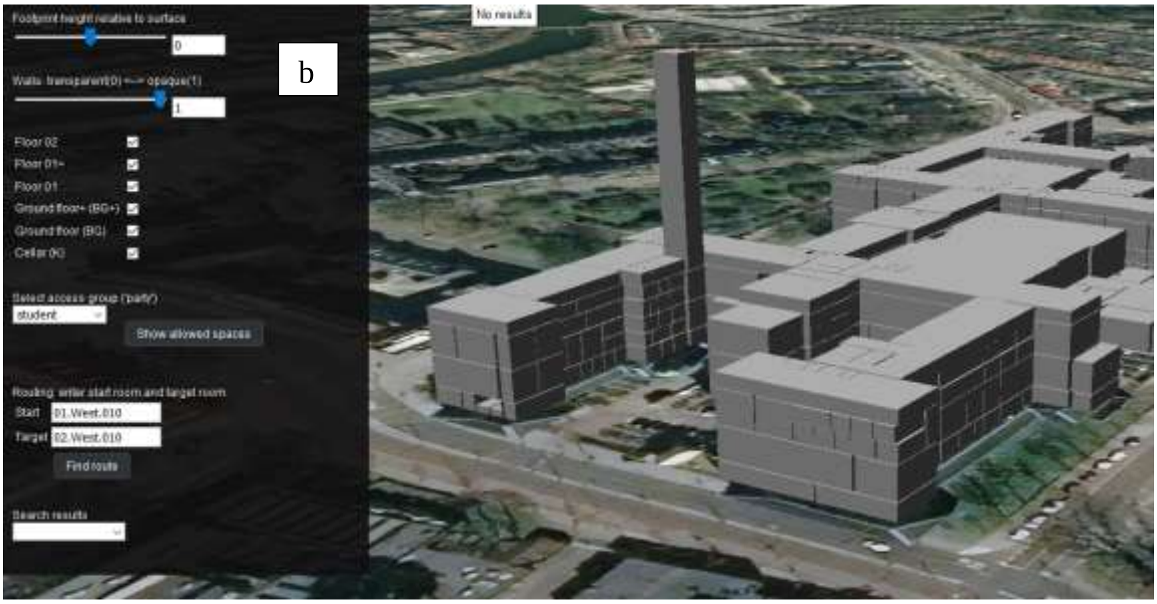


Figure 25: The route for a staff member (above) and a student (below) that have different access rights for the same destination

6 CONLIUSION

Many researchers have studied and developed indoor navigation applications for different purposes. Some of these applications have focused on the indoor environment's semantics, while other applications have focused on using different technology to support the navigation process. However, these applications did not consider the user accessibility rights during indoor navigation. Therefore, the integrated model of LADM and IndoorGML has been developed to define the access rights based on the relationship between the user and the indoor environment. By using international standards, we hope that in the future all complex buildings will share this information with their users, who could each use their own clients/front-end (on laptop, tablet, mobile phone) to use the services.

Each indoor environment has different access rights for the users based on the user type and individual. These access rights depend on categorizing the indoor environments' users into groups based on their relationship to the indoor environment. The access rights are also time dependent. Each user can access several spaces based on these rights. Having information about the access rights, the indoor navigation will be more accurate, because it reflects the relationship between the users and the indoor environment during the navigation. The user will have a navigation route based on these access rights.

This paper presented the development of a 3D web-application for indoor navigation based on user access rights. The development of the application started by creating a 3D BIM/IFC model for an educational building, consisting of several floors and different types of spaces. By creating the 3D BIM/IFC model, several types of information were attached to enrich the 3D model. The second step was extracting the semantic information and the location (xyz) of the spaces and doors to create the indoor environment's connectivity graph. By having all the previous information, a database was created in PostgreSQL with the extensions of PostGIS and pgRouting to create the nodes and edges of the graph and compute the navigation route for the users based on the access rights. Finally, the web-client GUI based on Cesium JS was used to specify the parameters (user, time, start, destination) and to visualize the indoor navigation routes.

The future work will focus on the following:

- Extending the views and GUI to visualize the access time rights for all the users.
- Testing the same approach in different indoor environments such as hospitals, shopping malls, office buildings, airports, etc.
- Using the integrated model of LADM and IndoorGML to develop indoor navigation applications for different indoor environments conditions such as emergencies.
- Developed mobile-based application based on the integrated model that uses the real data of the users.

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.

Becker, T., Nagel, C., and Kolbe, T. H.2009. A multilayered space-event model for navigation in indoor spaces. In 3D Geo-Information Sciences. Springer Berlin Heidelberg, 61-77

Benitez Sandoval, E; Li, B; Diakite, A; Zhao, K; Oliver, N; Bednarz, T; Zlatanova, S, 2020, A Visually Impaired User Experience using a 3D-Enhanced Facility Management System for Indoors Navigation, ICMI’20, International Conference of Multimodal Interaction, 5p https://dl.acm.org/doi/10.1145/3395035.3425247

Brown, G., C. Nagel, S. Zlatanova and T.H. Kolbe, 2013, Modelling 3D Topographic Space Against Indoor Navigation Requirements, Progress and New Trends in 3D Geoinformation Science, LNG&C, Springer, Heidelberg, New York, Dordrecht, London, pp. 1-22.

Cozzi, P., Lilley, S., & Getz, G. (2019). 3D Tiles Specification 1.0. *Open Geospatial Consortium: Wayland, MA, USA.*

Diakité A. A. and S. Zlatanova, 2018, Spatial subdivision of complex indoor environments for 3D indoor navigation, International Journal of Geographical Information Science, 32(2), pp. 213-235

Isikdag, U., Zlatanova, S., & Underwood, J. (2013). A BIM-Oriented Model for supporting indoor navigation requirements. *Computers, Environment and Urban Systems, 41*, 112-123.

ISO. ISO 19152:2012, Geographic Information-Land Administration Domain Model, 1st ed.; ISO: Geneva, Switzerland; 118p.

Ivanov, R. (2010, June). Indoor navigation system for visually impaired. In Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies (pp. 143-149).

Krūminaitė, M., & Zlatanova, S. (2014, November). Indoor space subdivision for indoor navigation. In *Proceedings of the sixth ACM SIGSPATIAL international workshop on indoor spatial awareness* (pp. 25-31).

Lee, J. (2004). A spatial access-oriented implementation of a 3-D GIS topological data model for urban entities. *GeoInformatica*, *8*(3), 237-264.

Lee, J.; Li, K.-J.; Zlatanova, S.; Kolbe, T.H.; Nagel, C.; Becker, T. OGC IndoorGML, OGC 14-0051r1. 2014. Available online: http://www.opengeospatial.org/standards/indoorgml#downloads (accessed on 8 March 2021).

Lemmen, C.H.J.; van Oosterom, P.J.M.; Thompson, R.; Hespanha, J.P.; Uitermark, H. The Modelling of Spatial Units (Parcels) in the Land Administration Domain Model (LADM). In Proceedings of the XXIV FIG International Congress, Sydney, Australia, 11–16 April 2010.

Liu, L. B. Li, S. Zlatanova, P. van Oosterom, 2021, Indoor navigation supported by the Industry Foundation Classes (IFC): A survey, Automation in Construction, Vol 121, January 2021, 10436

Liu, L., & Zlatanova, S. (2012, November). A semantic data model for indoor navigation. In *Proceedings of the Fourth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness* (pp. 1-8).

Liu, L., Zlatanova, S., Li, B., van Oosterom, P., Liu, H., & Barton, J. (2019). Indoor routing on logical network using space semantics. *ISPRS International Journal of Geo-Information*, *8*(3), 126.

Meijers, M., Zlatanova, S., & Pfeifer, N. (2005). 3D geo-information indoors: structuring for evacuation. *Proceedings of next generation 3D city models, Bonn, Germany, 6*, 11-16.

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

Ozdenizci, B., Ok, K., Coskun, V., & Aydin, M. N. (2011, April). Development of an indoor navigation system using NFC technology. In *2011 Fourth International Conference on Information and Computing* (pp. 11-14). IEEE.

Puikkonen, A., Sarjanoja, A. H., Haveri, M., Huhtala, J., & Häkkilä, J. (2009, November). Towards designing better maps for indoor navigation: experiences from a case study.

In *Proceedings of the 8th international conference on mobile and ubiquitous multimedia* (pp. 1-4).

Renaudin, V., Yalak, O., Tomé, P., & Merminod, B. (2007). Indoor navigation of emergency agents. *European Journal of Navigation, 5*(ARTICLE), 36-45.

Rueppel, U., & Stuebbe, K. M. (2008). BIM-based indoor-emergency-navigation-system for complex buildings. *Tsinghua science and technology, 13*(S1), 362-367.

Tsetsos, V., Anagnostopoulos, C., Kikiras, P., Hasiotis, P., & Hadjiefthymiades, S. (2005, July). A human-centered semantic navigation system for indoor environments. In *ICPS'05. Proceedings. International Conference on Pervasive Services, 2005.* (pp. 146-155). IEEE.

Visual scripting environment for designers: Dynamo. (2020, June 15). Retrieved March 17, 2021, from https://dynamobim.org/

Zlatanova, S.; Li, K.J.; Lemmen, C.; Oosterom, P. Indoor Abstract Spaces, 2016a, Linking IndoorGML and LADM. In Proceedings of the 5th International FIG 3D Cadastre Workshop, Athens, Greece, 18–20 October 2016; pp. 317–328.

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

Zlatanova, S., L. Liu, G. Sithole, J. Zhao and F. Mortari, 2014, Space subdivision for indoor applications, GISt Report, nr. 66, Delft, 2014 50 p.

BIOGRAPHICAL NOTES

Abdullah Alattas is a PhD candidate at the section ‘GIS Technology’, 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: Technische Universität München (TUM), Department of Cartography, Technische Universität Wien (TU Vienna), Research Group Cartography, and Technische Universität Dresden (TU Dresden), Institute for Cartography. In 2008, he received a bachelor’s degree in architecture from Faculty of Environmental Design, King Abdulaziz University in Jeddah, Saudi Arabia.

Marian de Vries holds an MSc in Economic and Social History from the Free University Amsterdam, The Netherlands (VU). Since 2001 she works as researcher at the Section GIS Technology, Faculty of Architecture and the Built Environment, Delft University of Technology (TU Delft). Focus of her research is on distributed geo-information systems. She participated in a number of projects for large data providers in the Netherlands such as

Rijkswaterstaat and the Dutch Cadastre, and in the EU projects HUMBOLDT (Data harmonisation and service integration) and ELF (European Location Framework).

Martijn Meijers (1981) is assitent professor at the chair of GIS technology, Digital Technologies, AE+T, faculty of Architecture and the Built Environment, Delft University of Technology. Martijn started his studies in Geodesy and Cartography at Utrecht University of Professional Education (Hogeschool van Utrecht), where he specialised in Geographic Information Systems. He continued his studies at Delft University of Technology, where he obtained a Master of Science degree in Geomatics in 2006. In 2011, he successfully defended his PhD thesis at this university on the topic of Variable-scale Geo-information. Martijns research interests include geo-database management systems, map generalisation, cartography and geo-visualization, (applied) computational geometry for GIS, handling large datasets and topological consistency.

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'.

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, head of the section 'GIS Technology'. He is the current chair of the FIG working group on '3D-Cadastres'.

CONTACTS

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

Marian de Vries

Delft University of Technology
Section GIS-technology,
Faculty of Architecture and the Built Environment
P.O. Box 5030, 2600 GA Delft
THE NETHERLANDS
Tel. +31 15 2784268
E-mail: M.E.deVries@tudelft.nl
website http://www.gdmc.nl

Martijn Meijers
Delft University of Technology,
Section GIS-technology,
Faculty of Architecture and the Built Environment
P.O. Box 5030, 2600 GA Delft
THE NETHERLANDS
Tel. +31 15 2785642
E-mail: B.M.Meijers@tudelft.nl
website http://www.gdmc.nl

Sisi Zlatanova
UNSW Built Environment
Kensington Campus
Sydney, NSW 2052 Australia
Tel: +61 2 93856847
E-mail: s.zlatanova@unsw.edu.au
website http://www.be.unsw.edu.au

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