



Development of a QGIS plugin for the CityGML 3D City Database

Konstantinos Pantelios
P5 - 29-06-2022

TU Delft Supervisors:

Camilo León Sánchez
Giorgio Agugiaro

External supervisors:

Nagel Claus
Zhihang Yao

Co-reader:

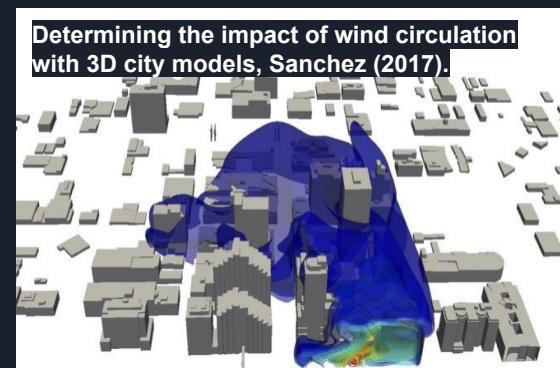
Martijn Meijers

3D City Models

- **Functionality**
 - Visualisation
 - Analysis and more...
- **Applications**
 - Urban planning
 - Energy modeling and more...
- **Data Management**
 - CityGML → OGC standard
 - Storage
 - Exchange
 - File encoding
 - E.g. XML, JSON (CityJSON)
 - Database encoding
 - E.g. SQL (3D City Database)



CityGML



Database encoding



3D City Models = Large amounts of data (usually)

Database functionalities:

- Dedicated resources (storage, processing)
- Easier queries of large data-sets
- Search, update, insert and delete operations
- Data structures (“native” index support)
- Organize and access data directly or via APIs and more...

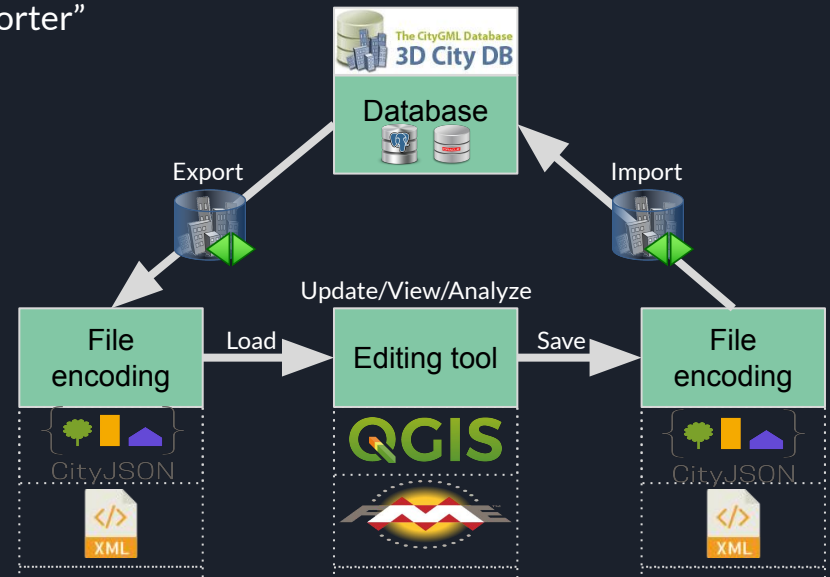
“3D City Database” (1/5)



- Open source software
- PostgreSQL (and Oracle)
- Supported by the “3DCityDB Importer/Exporter” tool and others

However!

- **Indirect** data usage



“3D City Database” (2/5)



Complicated direct data usage:

- **66 tables for 1 scenario (“citydb” schema) → Possibly multiple scenarios in the same database**
- Attributes are split over multiple tables
- Complex geometry table structure
- Features do not conform to the “Simple Feature Model” (SFM)

Resulting in:

- Requiring complex queries,
- Overall difficult interaction with the database

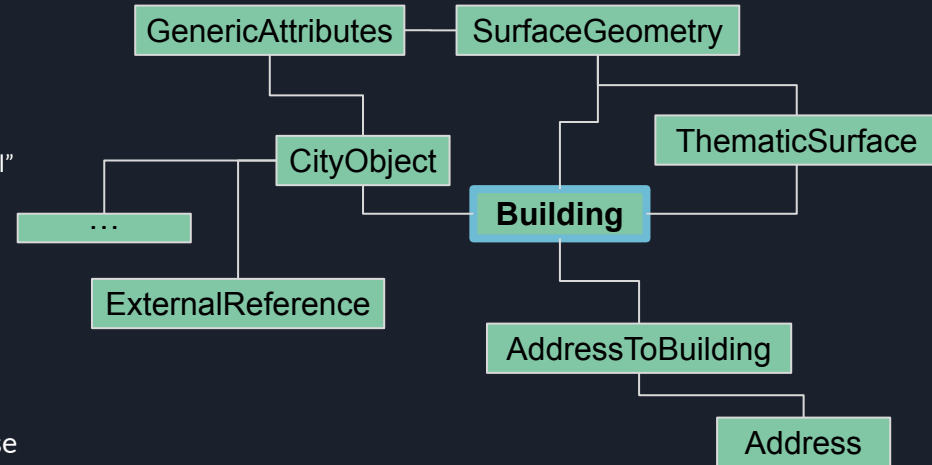
Tables (66)	cityobject_genericattrib	schema
address	cityobject_member	schema_referencing
address_to_bridge	cityobjectgroup	schema_to_objectclass
address_to_building	database_srs	solitary_vegetat_object
ade	external_reference	surface_data
aggregation_info	generalization	surface_geometry
appear_to_surface_data	generic_cityobject	tex_image
appearance	grid_coverage	textureparam
breakline_relief	group_to_cityobject	thematic_surface
bridge	implicit_geometry	tin_relief
bridge_constr_element	index_table	traffic_area
bridge_furniture	land_use	transportation_complex
bridge_installation	masspoint_relief	tunnel
bridge_open_to_them_srf	objectclass	tunnel_furniture
bridge_opening	opening	tunnel_hollow_space
bridge_room	opening_to_them_surface	tunnel_installation
bridge_thematic_surface	plant_cover	tunnel_open_to_them_srf
building	raster_relief	tunnel_opening
building_furniture	relief_component	tunnel_thematic_surface
building_installation	relief_feat_to_rel_comp	waterbod_to_waterbnd_srf
city_furniture	relief_feature	waterbody
citymodel	room	waterboundary_surface
cityobject		

“3D City Database” (3/5)



Complicated direct data usage:

- 66 tables for 1 scenario (“citydb” schema) → Possibly multiple scenarios in the same database
- **Attributes are split over multiple tables**
- Complex geometry table structure
- Features do not conform to the “Simple Feature Model” (SFM)



Resulting in:

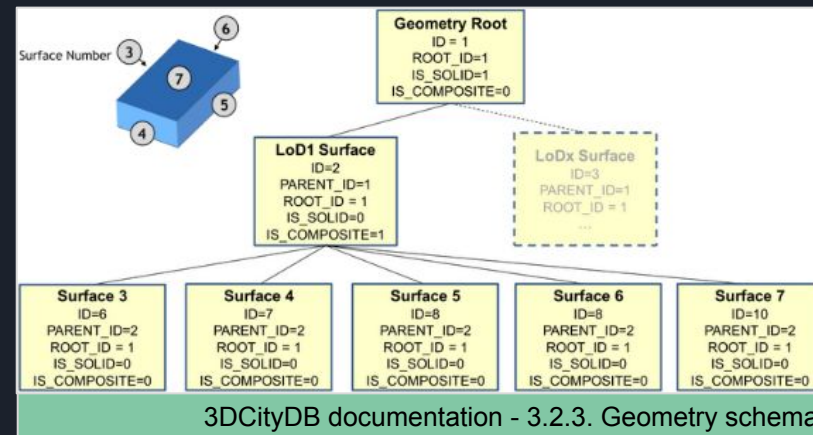
- Requiring complex queries,
- Overall difficult interaction with the database

“3D City Database” (4/5)



Complicated direct data usage:

- 66 tables for 1 scenario (“citydb” schema) → Possibly multiple scenarios in the same database
- Attributes are split over multiple tables
- **Complex geometry table structure**
- Features do not conform to the “Simple Feature Model” (SFM)



Resulting in:

- Requiring complex queries,
- Overall difficult interaction with the database

“3D City Database” (5/5)



Complicated direct data usage:

- 66 tables for 1 scenario (“citydb” schema) → Possibly multiple scenarios in the same database
- Attributes are split over multiple tables
- Complex geometry table structure
- **Features do not conform to the “Simple Feature Model” (SFM)**

Resulting in:

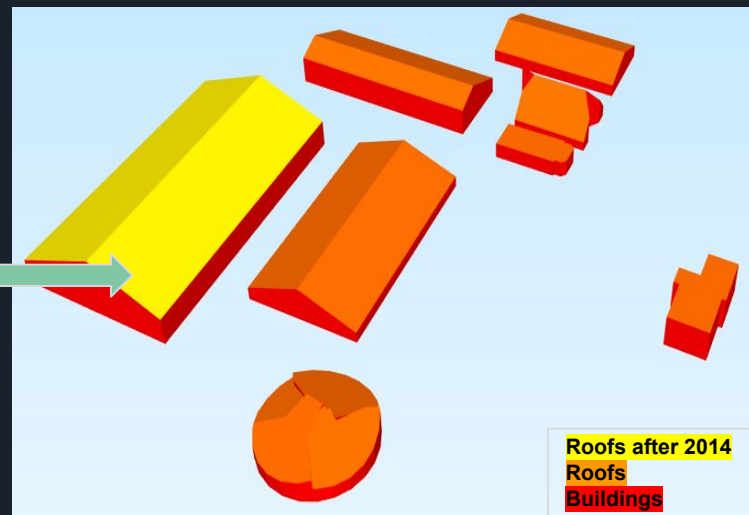
- Requiring complex queries,
- Overall difficult interaction with the database

#	Field name	Field type
1	id	Feature attribute
2	gmlid	Feature attribute
3	Feature attribute
15	class	Feature attribute
16	class_codespace	Feature attribute
17	Feature attribute
33	geom	Feature geometry

SFM layer example

Example query of building roofs constructed after the year 2014

```
1 SELECT
2   ts.id AS roof_id,
3   co_ts.gmlid AS roof_gmlid,
4   b.id AS building_id,
5   co.gmlid AS building_gmlid,
6   b.year_of_construction,
7   ST_Collect(sg.geometry) AS roof_geom
8 FROM
9   citydb.thematic_surface AS ts
10  INNER JOIN citydb.cityobject AS co_ts
11    ON (co_ts.id = ts.id)
12  INNER JOIN citydb.surface_geometry AS sg
13    ON (ts.lod2_multi_surface_id = sg.root_id)
14  INNER JOIN citydb.building AS b
15    ON (b.id = ts.building_id)
16  INNER JOIN citydb.cityobject AS co
17    ON (co.id = b.id)
18 WHERE
19   ts.objectclass_id = 33 AND -- roofsurfaces
20   b.objectclass_id = 26 AND -- buildings
21   b.year_of_construction >= '2015-01-01'::date
22 GROUP BY
23   ts.id,
24   co_ts.gmlid,
25   b.id,
26   co.gmlid,
27   b.year_of_construction
28 ORDER BY
29   b.id,
30   ts.id;
```



Objective - Research questions

How to simplify user interaction with the 3DCityDB in terms of:

- ❖ Visualization in 2D/3D of multi-LoD geometries?
- ❖ Accessing and editing attributes (if user is allowed)?

In terms of implementation:

- ❖ How to create a bi-directional interface between QGIS GUI (front-end) and the 3DCityDB (back-end)?
- ❖ Can we “restructure” data in 3DCityDB in order to conform to the SFM?

Related work (1/3)

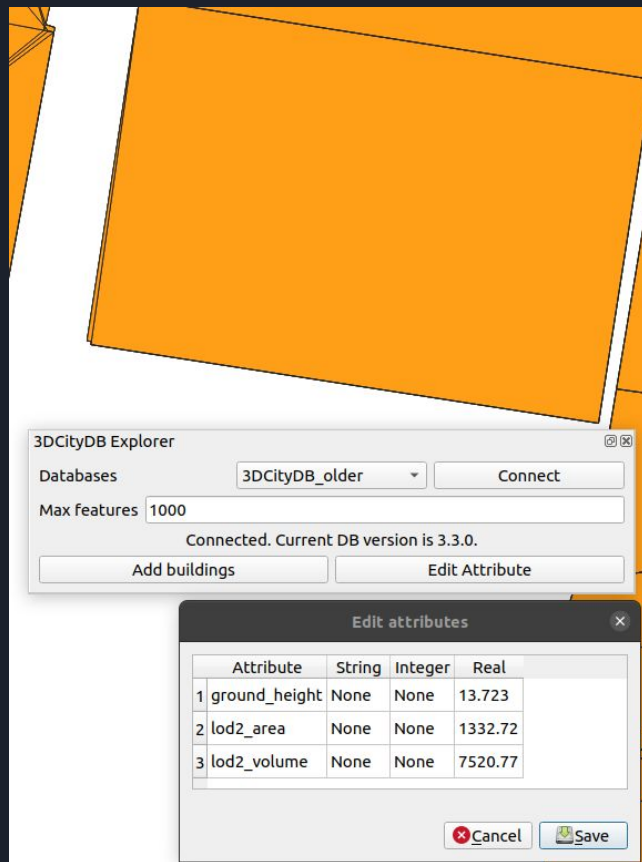
QGIS plugin: “3DCityDB Explorer” (08-03-2021)

Functionalities:

- Load data from database
- Update attributes
- Handles data size

Limitations:

- Loads ONLY the “Building” class of LOD2
- Updates ONLY the “genericAttribute” class
- Works only for the default “citydb” scenario
- Does not account for multiple users/privileges.



Related work (2/3)

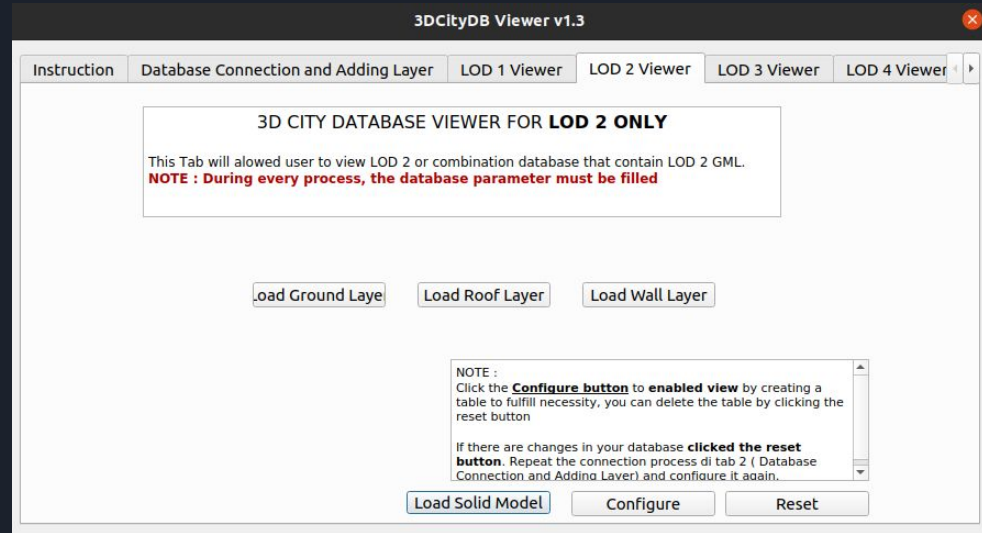
QGIS plugin: “3DCityDB Viewer” (16-06-2021)

Functionalities:

- Load data from database

Limitations:

- Loads ONLY the “Building” module (of all representations)
- ONLY the geometries of “Building” module classes are loaded.
- Works only for the default “citydb” scenario.
- Does not account for multiple users/privileges.
- Does not handle data size
- Confusing GUI (IMO)



Related work (3/3)

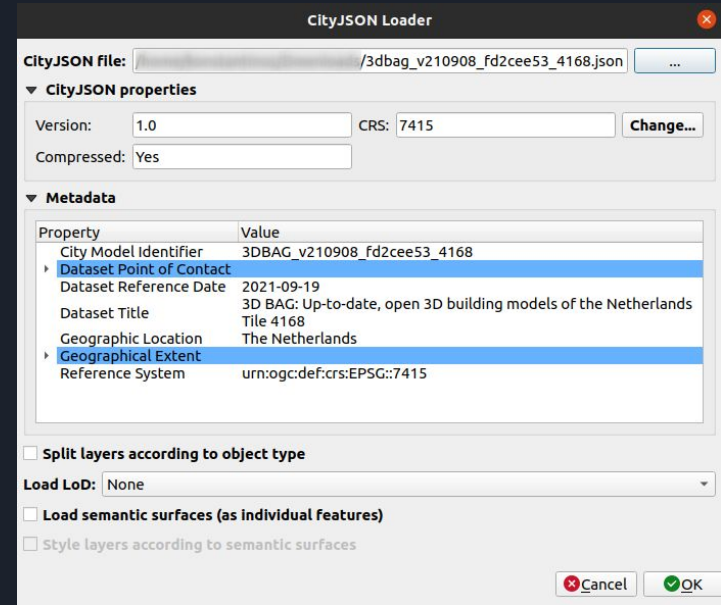
QGIS plugin: “CityJSON Loader” (23-03-2021)

Functionalities:

- Load only CityJSON files
- Allow for multiple layer representations
- Allow use of standard color schema

Limitations:

- It is not possible to store updates back at the original file.
- Does not handle data size



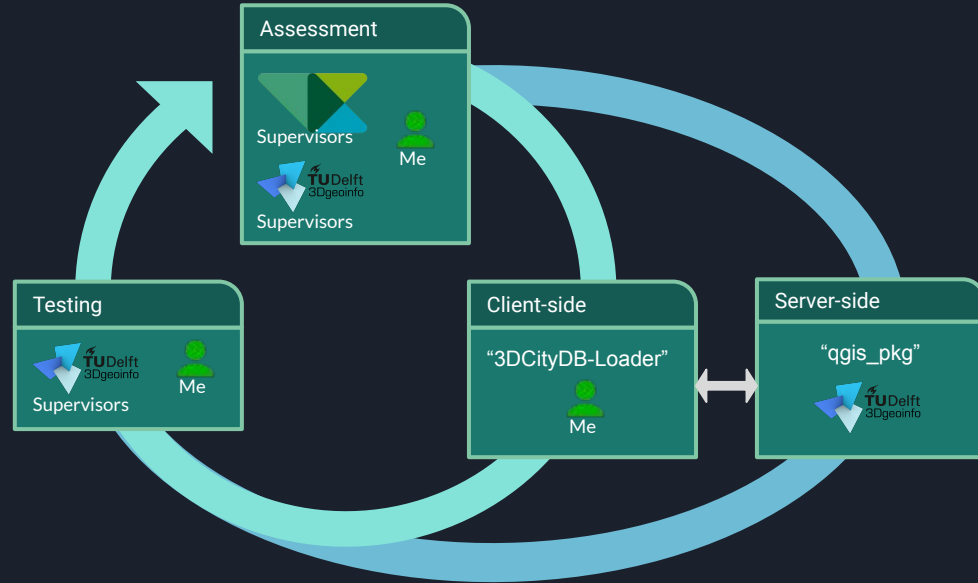
Methodology (1/2): Requirements

Explore/discover requirements:

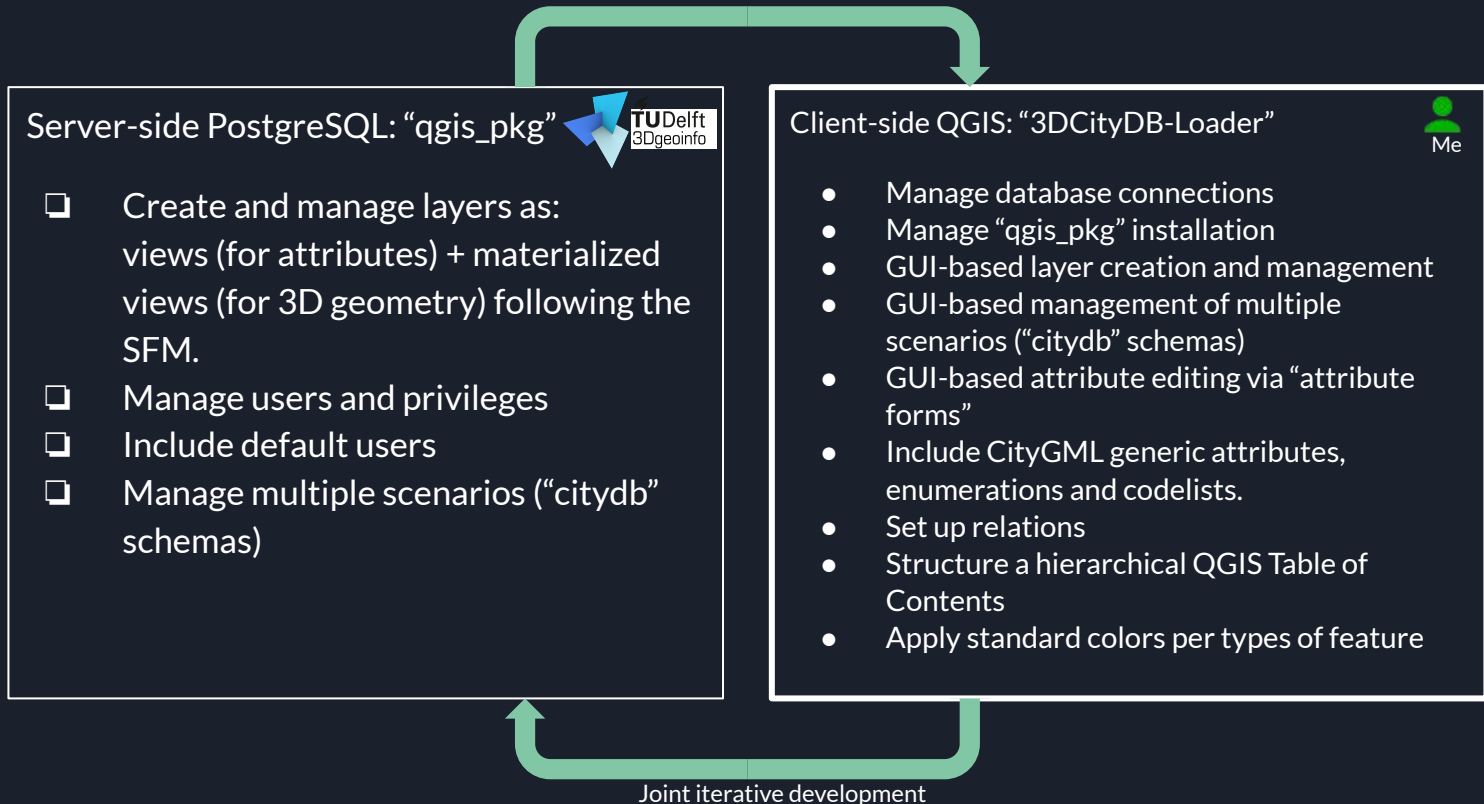
- Meetings with “TU Delft” and “VirtualCitySystems” supervisors
- Iterative process (4 iterations)

Primary requirements:

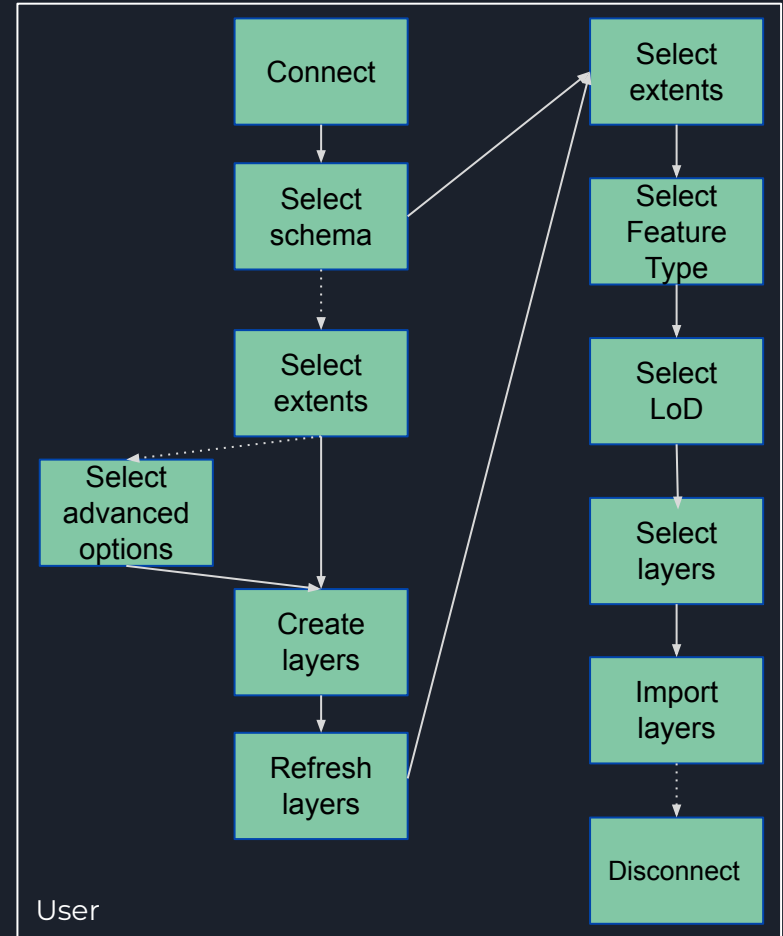
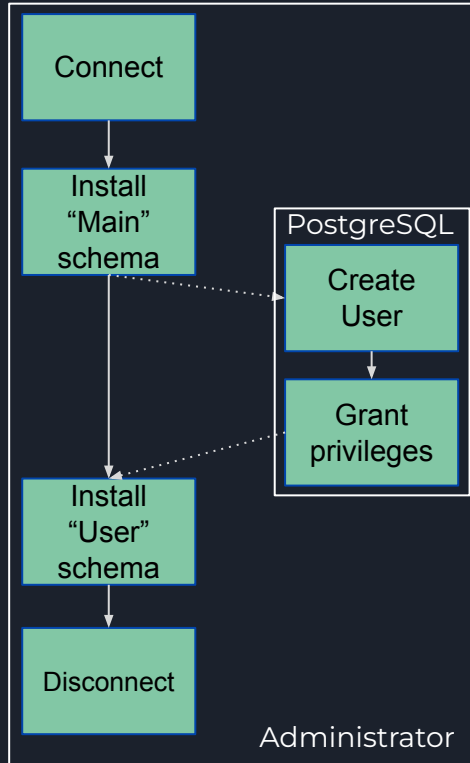
- “Data layers” able to interact with 3DCityDB data
- Multi-user with different privileges (Read-only, Read-write)
- Allow multiple scenarios (“citydb” schema).
- Edit ONLY attributes, no geometries
- Deal with multi-LoD, multi-geometry representations
- Operate from a GUI (QGIS)



Methodology (2/2): Design decisions



“3DCityDB-Loader” Workflow



"3DCityDB-Loader" GUI

3DCityDB-Loader (Administration)

Database Administration
User Connection
Layers
About

Connection

Select an existing connection:

or

Make new connection:

Connect to plugin_dev

Main Installation

User Installation

Select User:

Connection Status

Connected to Database: ✔ plugin_dev

PostgreSQL installation: ✔ 12.10 (Ubuntu 12.10-1.pgdg20.04+1+b1)

3DCityDB installation: ✔ 4.1.0

Main installation: ✔ qgis_pkg is already installed! (v.0.6.1)

User installation: ✔ qgis_user_ro is already installed!

Close current connection

Administrator

3DCityDB-Loader

User Connection
Layers
About

Connection

Select an existing connection:

or

Make new connection:

Database

Connect:

Select citydb schema:

Basemap (OSM)

▼ Extent (current: map view)


North

West East

South

Calculate from Layer:

Set to den_haag schema



► Advanced options

Connection Status

Connected to Database: ✔ plugin_dev

PostgreSQL installation: ✔ 12.11 (Ubuntu 12.11-1.pgdg20.04+1)

3DCityDB installation: ✔ 4.1.0

Main installation: ✔ qgis_pkg is already installed! (v.0.7.0)

User installation: ✔ qgis_postgres is already installed!

Schema support: ✔ Layers already exist for den_haag!

Layer refresh state: ✔ Last refresh: 2022-06-14 18:12:41.486000+02:00

Close current connection

3DCityDB-Loader

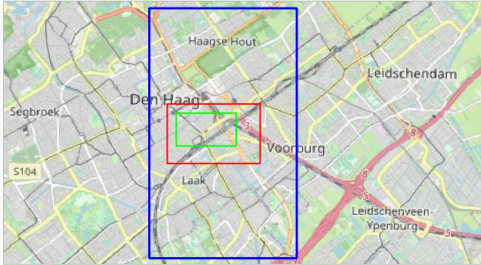
User Connection
Layers
About

Database: plugin_dev

Current user: postgres

Current citydb schema: den_haag

Basemap (OSM)



▼ (current: map view)

North

West East

South

Calculate from Layer:

Set to layers extents schema

Layer selection

Feature type:

Level of Detail:

Available layers

Import selected layers

User

“3DCityDB-Loader” in practice (1/7)

Database administration example

Schemas (10)	
> citydb	“citydb” schemas (scenarios)
> citydb_dutch	
> citydb_pkg	Aux. schemas
> qgis_pkg	“Main” schema
> qgis_postgres	
> qgis_user_ro	“User” schemas
> qgis_user_rw	

client	Custom users
employee	
postgres	Default users
qgis_user_ro	
qgis_user_rw	

“3DCityDB-Loader” in practice (2/7)

Use example as database regular user

Select citydb schema:


Basemap (OSM)

▼ Extent (current: map view)

North

West East

South



KIT_house
citydb_dutch
den_haag
rail

- Blue square: scenario extents (“citydb” schema in database - e.g. citydb_dutch)
- Red square: User-defined layer extents (layers in database)

"3DCityDB-Loader" in practice (3/7)


Use example as database regular user

Select citydb schema:

Basemap (OSM)

▼ Extent (current: map view)

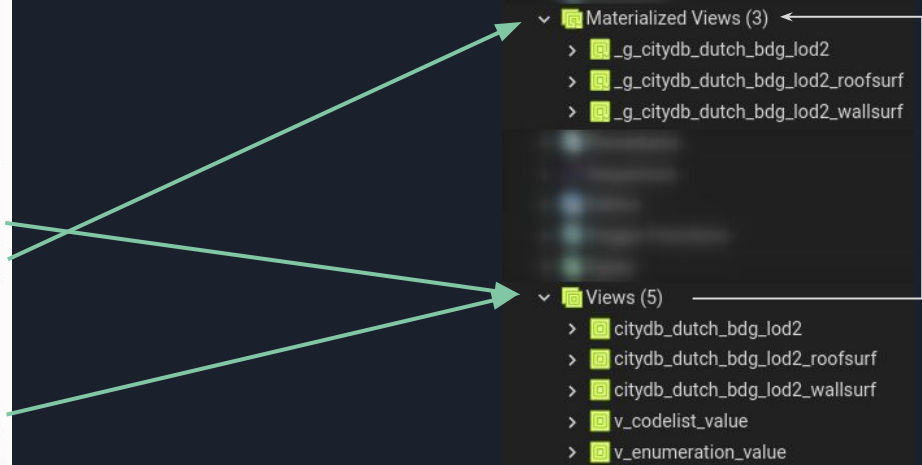
North
 West East
 South



► Advanced options

▼ qgis_postgres

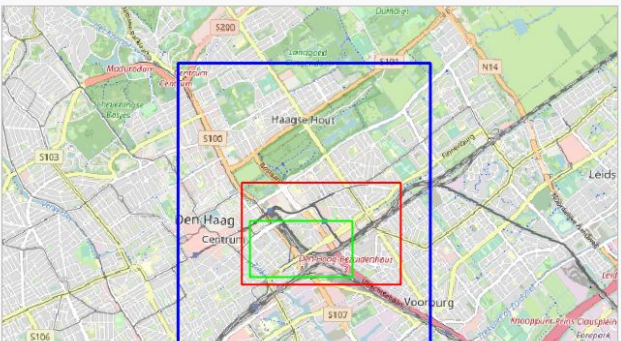
- ▼ Materialized Views (3)
 - > _g_citydb_dutch_bdg_lod2
 - > _g_citydb_dutch_bdg_lod2_roofsurf
 - > _g_citydb_dutch_bdg_lod2_wallsurf
- ▼ Views (5)
 - > citydb_dutch_bdg_lod2
 - > citydb_dutch_bdg_lod2_roofsurf
 - > citydb_dutch_bdg_lod2_wallsurf
 - > v_codelist_value
 - > v_enumeration_value



“3DCityDB-Loader” in practice (4/7)

Use example as database regular user

Basemap (OSM)



▼ (current: map view)

North	<input type="text" value="455161.072546875"/>		
West	<input type="text" value="81859.188575431"/>	East	<input type="text" value="83248.839630792"/>
South	<input type="text" value="454399.500000000"/>		

- Blue square: scenario extents (“citydb” schema in database - e.g. citydb_dutch)
- Red square: layer extents (layers in database)
- Green square: User-defined layer extents (QGIS layers)

“3DCityDB-Loader” in practice (5/7)

Use example as database regular user

Layer selection

Feature type:

Level of Detail:

Available layers

- bdg_door_lod3 (9)
- bdg_lod3 (3)
- bdg_lod3_groundsurf (3)
- bdg_lod3_outerceilingsurf (3)**
- bdg_lod3_outerfloorsurf (1)
- bdg_lod3_roofsurf (8)
- bdg_lod3_wallsurf (25)
- bdg_out_inst_lod3 (56)
- bdg_window_lod3 (34)

- Views (32)
 - > citydb_dutch_bdg_lod2
 - > citydb_dutch_bdg_lod2_roofsurf
 - > citydb_dutch_bdg_lod2_wallsurf
 - > rail_bdg_door_lod3
 - > rail_bdg_lod3
 - > rail_bdg_lod3_groundsurf
 - > rail_bdg_lod3_outerceilingsurf
 - > rail_bdg_lod3_outerfloorsurf
 - > rail_bdg_lod3_roofsurf
 - > rail_bdg_lod3_wallsurf
 - > rail_bdg_out_inst_lod3
 - > rail_bdg_window_lod3
 - > rail_bri_constr_elem_lod3
 - > rail_bri_lod3
 - > rail_bri_out_inst_lod3
 - > rail_city_furn_lod3
 - > rail_gen_cityobj_lod3
 - > rail_railway_lod3
 - > rail_relief_feat_lod3
 - > rail_sol_veg_obj_lod3
 - > rail_tin_relief_lod3
 - > rail_tun_lod3
 - > rail_tun_lod3_closuresurf
 - > rail_tun_lod3_groundsurf
 - > rail_tun_lod3_roofsurf
 - > rail_tun_lod3_wallsurf
 - > rail_tun_out_inst_lod3
 - > rail_waterbody_lod3
 - > rail_waterbody_lod3_watergroundsurf
 - > rail_waterbody_lod3_watersurf
 - > v_codelist_value
 - > v_enumeration_value

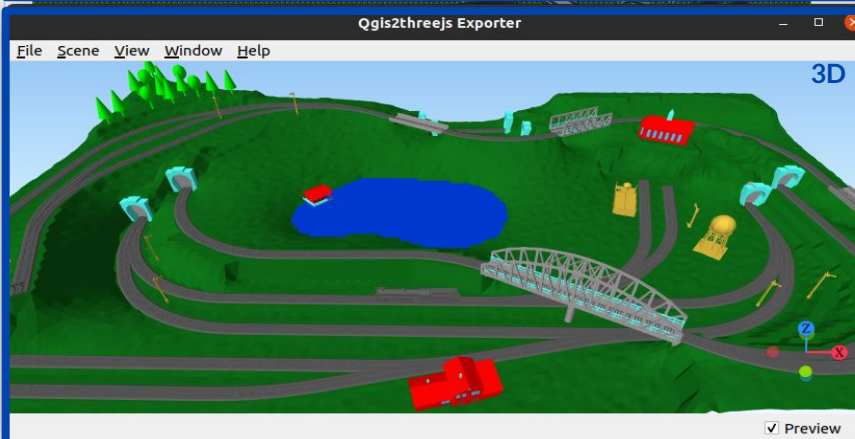
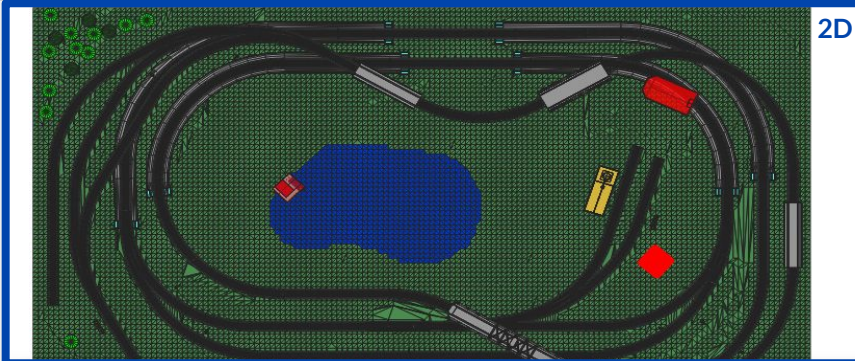
*Untitled Project — QGIS

Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Database Viewer Processing Help

Layers

- plugin_dev
 - postgres@rail
 - FeatureType: Bridge
 - Bridge
 - lod3
 - rail_bri_constr_elem_lod3
 - rail_bri_lod3
 - rail_bri_out_inst_lod3
 - FeatureType: Building
 - Building
 - lod3
 - rail_bdg_door_lod3
 - rail_bdg_lod3
 - rail_bdg_lod3_groundsurf
 - rail_bdg_lod3_outerceilingsurf
 - rail_bdg_lod3_outerfloorsurf
 - rail_bdg_lod3_roofsurf
 - rail_bdg_lod3_wallsurf
 - rail_bdg_out_inst_lod3
 - rail_bdg_window_lod3
 - FeatureType: CityFurniture
 - CityFurniture
 - lod3
 - rail_city_furn_lod3
 - FeatureType: Generics
 - GenericCityObject
 - lod3
 - rail_gen_cityobj_lod3
 - FeatureType: Transportation
 - Railway
 - lod3
 - rail_railway_lod3
 - FeatureType: Tunnel
 - FeatureType: Vegetation
 - FeatureType: WaterBody
 - FeatureType: Relief
 - Generic Attributes
 - Look-up tables
 - 3DCityDB
 - postgres@citydb
 - FeatureType: Building

"3DCityDB-Loader"



Log Messages

Time	Status	Message
2022-06-23T13:13:46	SUCCESS	Create relation: re_rail_waterbody_lod3_watersurf
2022-06-23T13:13:46	SUCCESS	
2022-06-23T13:13:50	SUCCESS	Layer import: rail_sol_veg_obj_lod3
2022-06-23T13:13:50	SUCCESS	Create relation: re_rail_sol_veg_obj_lod3
2022-06-23T13:13:50	SUCCESS	
2022-06-23T13:13:53	SUCCESS	Layer import: rail_gen_cityobj_lod3
2022-06-23T13:13:53	SUCCESS	Create relation: re_rail_gen_cityobj_lod3
2022-06-23T13:13:53	SUCCESS	

Log Panel

Attributes

Feature	Value
▼ rail_city_furn_lod3	
Name	NULL
(Derived)	
(Actions)	
Database ID	227
GML ID	GMLID_1381455_184173_656
GML codespace	NULL
Name	NULL
Name codespace	NULL
Description	NULL
Creation date	17-03-2022 08:05:30
Termination date	NULL
Relative to terrain	NULL
Relative to water	NULL
Last modification	17-03-2022 08:05:30
Updating person	postgres
Reason for update	NULL
Lineage	NULL
Class	NULL
Codespace	NULL
Function	OTHERS
Codespace	NULL
Usage	NULL
Codespace	NULL

Mode: Layer Selection

View: Tree

"3DCityDB-Loader" in practice (7/7)

citydb_bdg_lod2 - Feature Attributes

Main Info Database Info Relation to surface Generic Attributes

Database ID 24918 ✓

GML ID NL.IMBAG.Pand.1742100000008466

GML codespace NULL

Name Building 480-835 ✕

Name codespace NULL

Description NULL

Class Function Usage

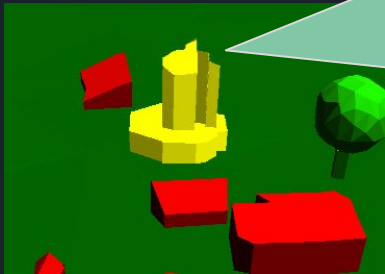
Class Non-residential (multi function) ✕

Codespace NULL

Feature-specific attributes

Year of construction	1889 ✕	Year of demolition	NULL
Storeys above ground	1 ✕	Storeys below ground	-5 ✕
Height	45 ✓	UoM	m ✓
Storey height above ground	3 ✕	UoM	NULL ✕
Storey height below ground	3 ✓	UoM	m ✓
Roof type	slanted ✕	Codespace	NULL

✕ Cancel ✓ OK



Conclusions (1/5)

How to simplify user interaction with the 3DCityDB in terms of:

- ❖ Visualization in 2D/3D of multi-LoD geometries. ✓
- ❖ How to access attributes and edit them (if user is allowed)? ✓

In terms of implementation:

- ❖ How to create a bi-directional interface between QGIS GUI (front-end) and the 3DCityDB (back-end)? ✓
- ❖ Can we “restructure” data in 3DCityDB in order to conform to the SFM? ✓

Conclusions (2/5)

How to simplify user interaction with the 3DCityDB in terms of:

- ❖ Visualization in 2D/3D of multi-LoD geometries. ✓
- ❖ How to access attributes and edit them (if user is allowed)? ✓



Conclusions (3/5)

How to simplify user interaction with the 3DCityDB in terms of:

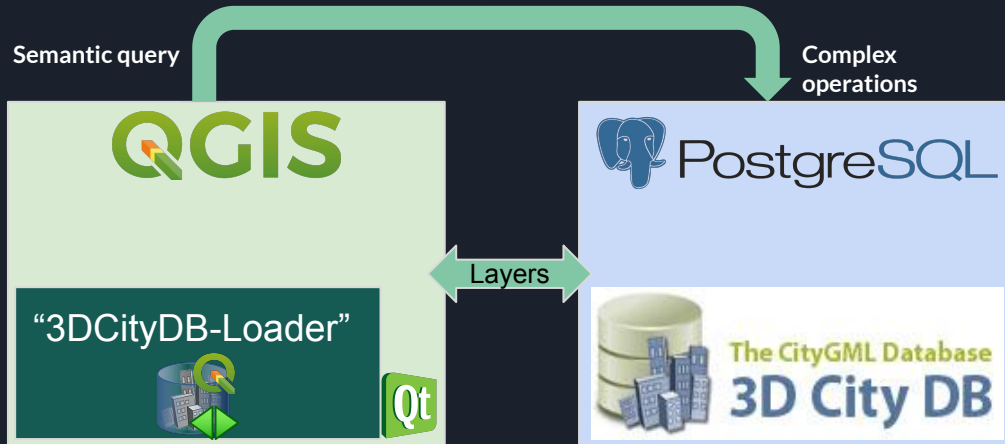
- ❖ Visualization in 2D/3D of multi-LoD geometries. ✓
- ❖ How to access attributes and edit them (if user is allowed)? ✓

The screenshot displays the 3DCityDB web interface for editing a feature. The window title is "KIT_house_bdg_lod0_footprint — Features Total: 1, Filtered: 1, Selected: 0". The interface is divided into several sections:

- Main Info:** Contains fields for Database ID (1), GML ID (UUID_d281adfc-4901-0f52-540b-4cc1a9325f82), GML codespace (NULL), Name (AC14-FZK-Haus), and Name codespace (NULL).
- Class:** Shows Class (1000) and Codespace (http://www.sig3d.org/codelists/citygml/2.0/building/2.0/_AbstractBuilding_class.xml).
- Feature-specific attributes:** A table of attributes for editing, each with a value, a delete icon, and a checkmark indicating editability.

Feature-specific attributes	
Year of construction	2020
Year of demolition	NULL
Storeys above ground	2
Storeys below ground	0
Height	6.52
UoM	m
Storey height above ground	NULL
UoM	NULL
Storey height below ground	NULL
UoM	NULL
Roof type	1030
Codespace	nl/2.0/building/2.0/_AbstractBuilding_roofType.xml

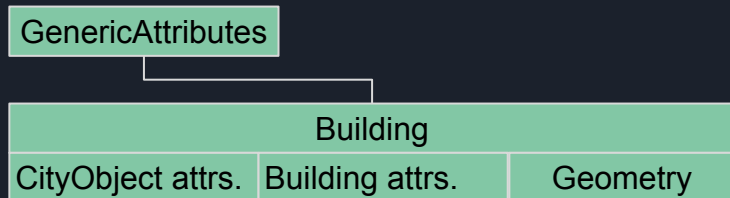
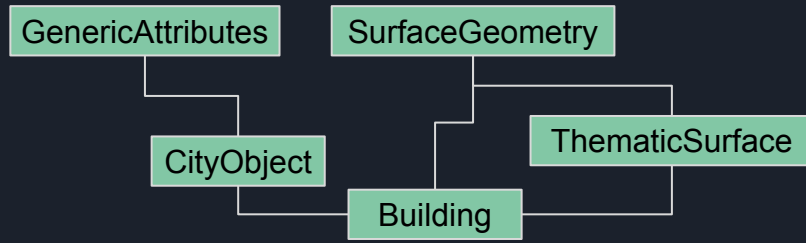
Conclusions (4/5)



In terms of implementation:

- ❖ How to create a bi-directional interface between QGIS GUI (front-end) and the 3DCityDB (back-end)? ✓
- ❖ Can we “restructure” data in 3DCityDB in order to conform to the SFM? ✓

Conclusions (5/5)



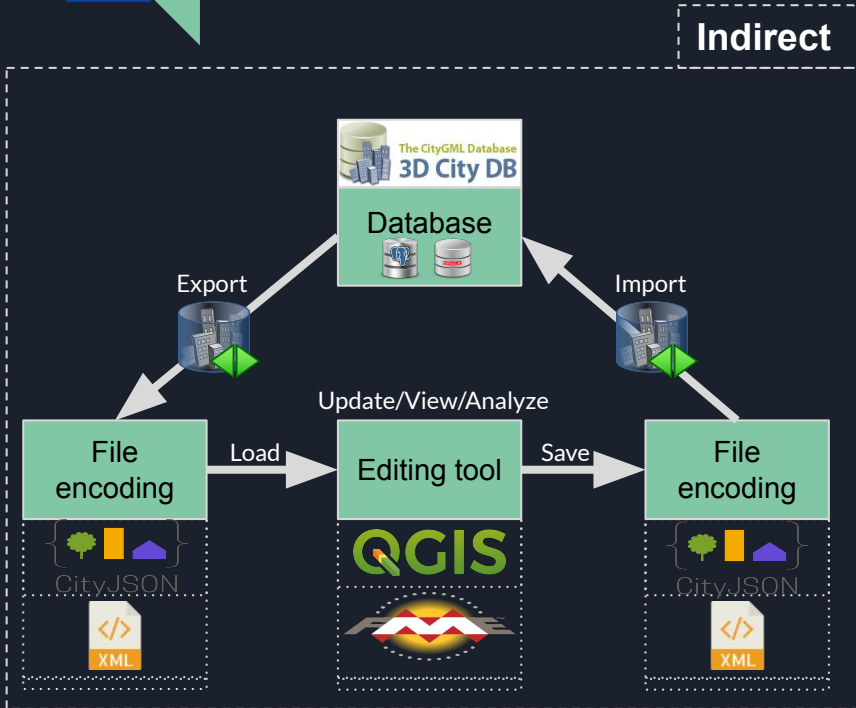
In terms of implementation:

- ❖ How to create a bi-directional interface between QGIS GUI (front-end) and the 3DCityDB (back-end)? ✓
- ❖ Can we “restructure” data in the 3DCityDB in order to conform to the SFM? ✓

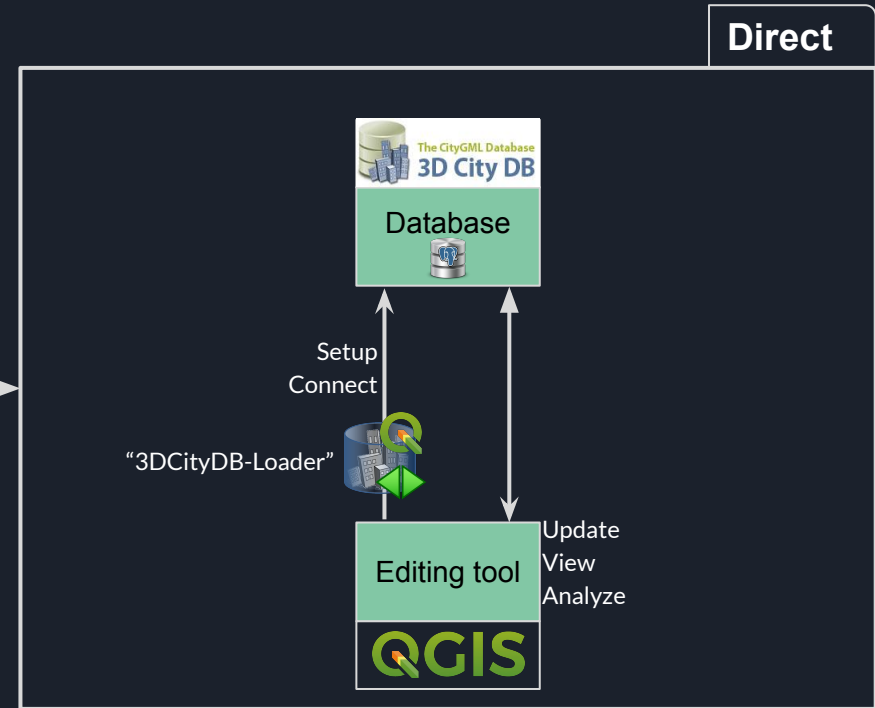
#	Field name	Field type
1	id	Feature attribute
2	gmlid	Feature attribute
3	Feature attribute
15	class	Feature attribute
16	class_codespace	Feature attribute
17	Feature attribute
33	geom	Feature geometry

SFM layer example

Indirect



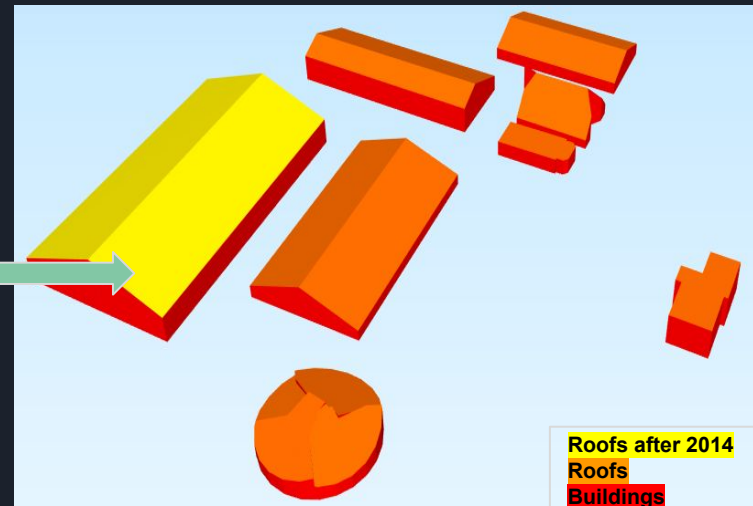
Direct



Example query of building roofs constructed after the year 2014.

```
1 SELECT
2   ts.id AS roof_id,
3   co_ts.gmlid AS roof_gmlid,
4   b.id AS building_id,
5   co.gmlid AS building_gmlid,
6   b.year_of_construction,
7   ST_Collect(sg.geometry) AS roof_geom
8 FROM
9   citydb.thematic_surface AS ts
10  INNER JOIN citydb.cityobject AS co_ts
11    ON (co_ts.id = ts.id)
12  INNER JOIN citydb.surface_geometry AS sg
13    ON (ts.lod2_multi_surface_id = sg.root_id)
14  INNER JOIN citydb.building AS b
15    ON (b.id = ts.building_id)
16  INNER JOIN citydb.cityobject AS co
17    ON (co.id = b.id)
18 WHERE
19   ts.objectclass_id = 33 AND -- roofsurfaces
20   b.objectclass_id = 26 AND -- buildings
21   b.year_of_construction >= '2015-01-01'::date
22 GROUP BY
23   ts.id,
24   co_ts.gmlid,
25   b.id,
26   co.gmlid,
27   b.year_of_construction
28 ORDER BY
29   b.id,
30   ts.id;
```

Using vanilla 3DCityDB

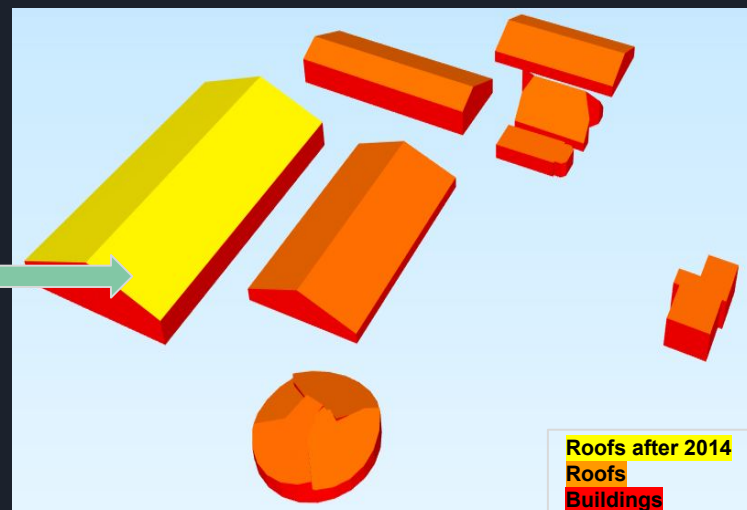


*this queries all of the roofs stored in the database scenario. The image shows only a small sub-area

Example query of building roofs constructed after the year 2014.

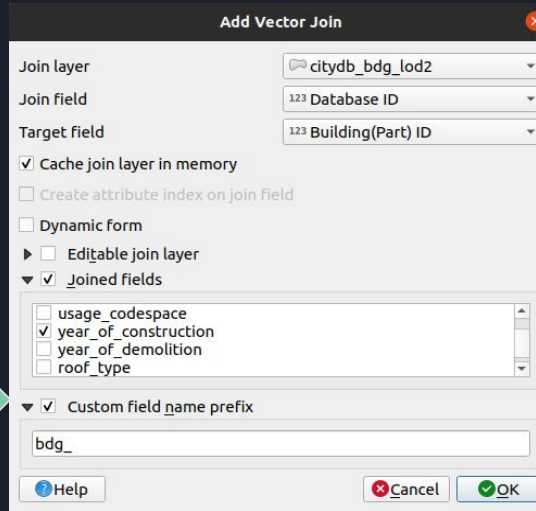
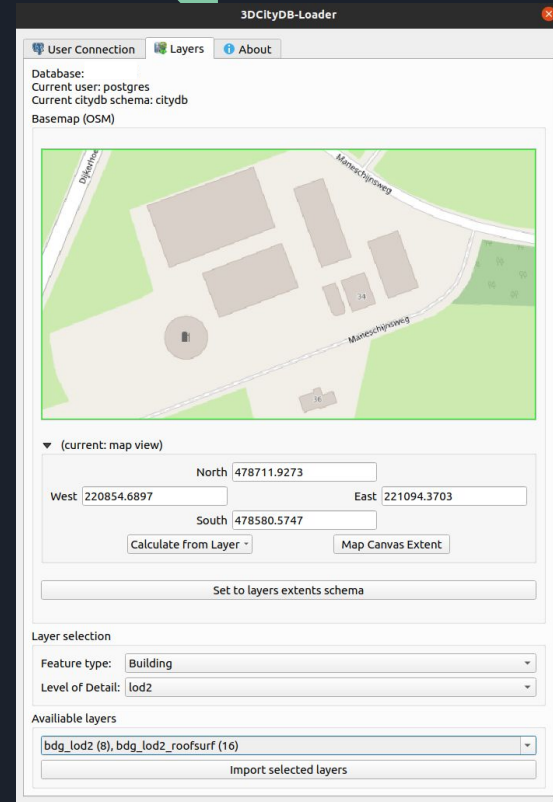
```
1 SELECT
2   rs.id AS roof_id,
3   rs.gmlid AS roof_gmlid,
4   rs.building_id AS bdg_id,
5   b.gmlid AS bdg_gmlid,
6   b.year_of_construction,
7   rs.geom AS roof_geom
8 FROM
9   qgis_user_ro.citydb_bdg_lod2_roofsurf AS rs
10  INNER JOIN qgis_user_ro.citydb_bdg_lod2 AS b
11    ON b.id = rs.building_id
12 WHERE
13   b.year_of_construction >= '2015-01-01'::date
14 ORDER BY
15   b.id,
16   rs.id;
```

Using server-side of "3DCityDB-Loader"

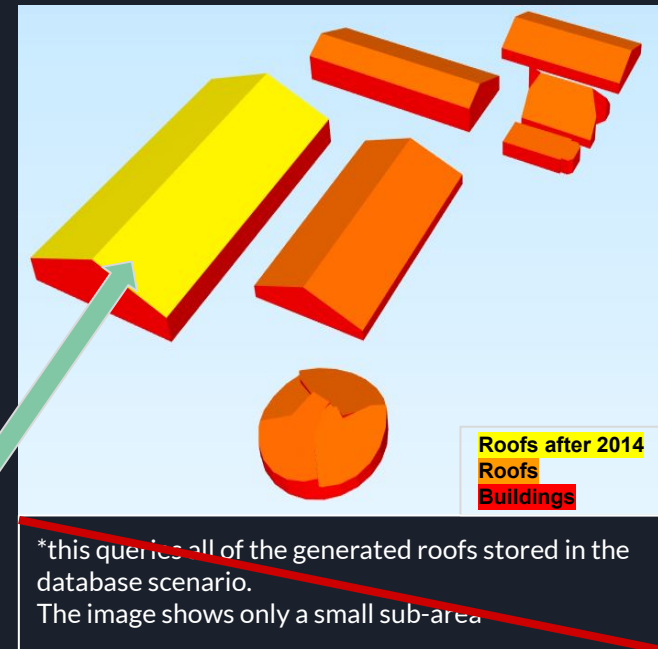


*this queries all of the generated roofs stored in the database scenario.
The image shows only a small sub-area

Example query of building roofs constructed after the year 2014.



`"bdg_year_of_construction" >= make_date(2015,1,1)`

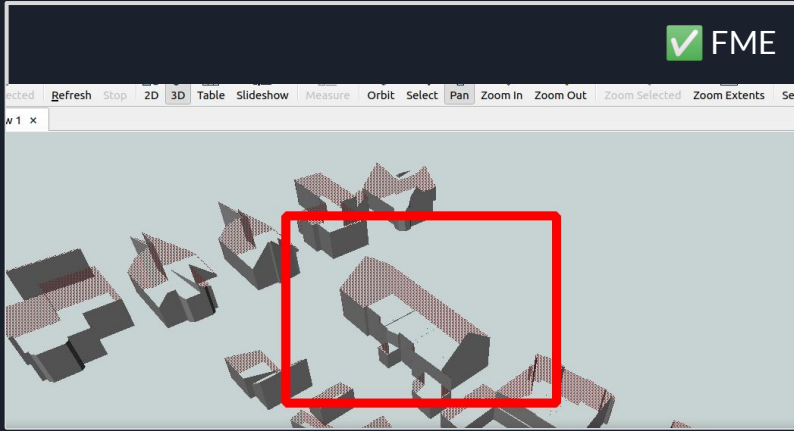




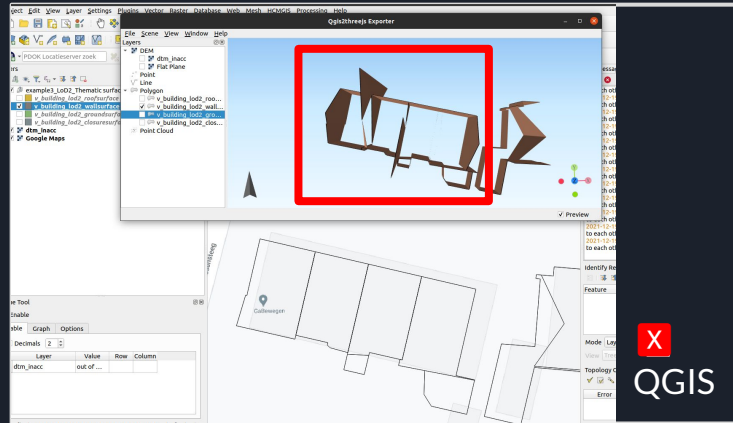
Limitations

- Not developed to insert or update geometries
- Custom users (along with privileges) currently can only be set outside of the GUI workflow
- No maximum size limit for layers (large datasets)
- Base-map tool to select extents by drawing a square on map is unavailable
- Potential issues with 3D visualization due (probably) to precision loss from projected coordinates

3D Visualization issue



✓
Google earth
(as KML)





Outlook

- ❑ Further testing with more and heterogeneous data-sets
- ❑ Add management of the CityGML “Appearances” class
- ❑ Collect more feedback from selected early adopters
- ❑ Explore and add new functionalities
(tracked in the project’s GitHub repository)

Feedback from test users

Some proposals for improvement #37

Open msanchezaparclo opened this issue 12 days ago · 0 comments



msanchezaparclo commented 12 days ago



As a beginner on 3DCityDB - Loader, i suggest some improvements that would be interesting:

- At least in me case, i have to activate the tool manually in the toolbar. Its would be interesting if this step were automatic.
- Have a different icon for the administration than for the normal user to be able to identify them without any problems.
- If the user decide to desinstall the plugins, give the possibility to delete all the connections created.
- In the *administration window*:

1. A botton to delete or edit connections already created.
2. When the user creates a new connection, in the window "Establish new postgresQL connection", an
3. Once the values have been introduced in the different line edit, it would be nice that when cor

I hope it will help you

22 Open ✓ 14 Closed

Some proposals for improvement

#37 opened 2 days ago by msanchezaparclo

Add option to Delete and Edit existing connection enhancement

#36 opened 2 days ago by Konstantinos-Pantellos

QGIS 3.22: it doesnt work

#35 opened 9 days ago by msanchezaparclo

Add check for QGIS version before loading enhancement

#34 opened 10 days ago by gioagu

QGIS 3.20: it works

#33 opened 11 days ago by gioagu

QGIS 3.24: it works

#32 opened 11 days ago by gioagu

QGIS 3.16: Cannot load main puglin window (for user) bug

#31 opened 11 days ago by gioagu

QGIS 3.18: Cannot import selected layers bug question

#30 opened 12 days ago by yaozhilang



Thank you for your attention!