

3D CITY MODEL AS A FIRST STEP TOWARDS DIGITAL TWIN OF SOFIA CITY

Hristiyan Dimitrov¹, *Dessislava Petrova-Antonova²

¹Technical University of Munich, 21 Arcisstraße, 80333 München, Germany

²Sofia University “St. Kl. Ohridski”, FMI, GATE Institute, 125 Tsarigradsko shose Blvd., 1113 Sofia, Bulgaria
hristiyan.dimitrov97@gmail.com, d.petrova@fmi.uni-sofia.bg

KEY WORDS: 3D City Model, CityGML 2.0, Buildings, Relief, Digital Twin, Spatial Data Integration.

ABSTRACT:

Semantic 3D city models are increasingly applied for a wide range of analysis and simulations of large urban areas. Such models are used as a foundation for development of city digital twins, representing with high accuracy the landscapes and urban areas as well as dynamic of the city in terms of processes and events. In this context, this paper presents a 3D city model, which is a starting point for development of digital twin of Sofia city. The 3D model is compliant with CityGML 2.0 in LOD1, supporting integration of the buildings and terrain and enriching the buildings' attributes with address information. District Lozenets of Sofia city is chosen as a pilot area for modelling. An approach for 3D transformation of proprietary geospatial data into CityGML schemas is presented. The integration of the buildings and terrain is an essential part of it, since the buildings often partially float over or sink into the terrain. A web application for user interaction with the 3D city model is developed. Its main features include silhouetting a single building, showing relevant overlay content, displaying shadows and styling of buildings depending on their attributes.

1. INTRODUCTION

Over the last decades, a lot of research efforts are dedicated to development of semantic 3D city models, covering variety city dimensions based on standards such as CityGML. Such models go beyond the simple 3D visualisation by providing a solid foundation for a wide range of urban analysis and simulations. With the rapid adoption of the Digital Twin's concept, 3D city models are used for variety applications such as urban planning, disaster management, solar potential analysis, air pollution simulation, etc. (Biljecki et al., 2015) and (Julin et al., 2018).

Currently, more than a thousand city models exist worldwide (Morton et al., 2012). The majority of them are implemented using either the CityGML or IFC standard (Zlatanova et al., 2020) and (Arroyo Ohori et al., 2018). A number of techniques can be applied for their development such as aerial photographs and laser scanning (Blaschke, 2010) and (Tomljenovic et al., 2015), extrusion from 2D footprints (Arroyo Ohori et al., 2015), airborne point clouds (Shahzad and Zhu, 2015) and (Wang et al., 2020), architectural drawings and plans (Yin et al., 2009) and (Lewis and Séquin, 1998), procedural modelling (Smelik et al., 2014) and (Besuievsky and Patow, 2014), and open street map (Goetz, 2013) and (Over et al., 2010). The 3D models, based on CityGML, represent the cityscape with respect to geometry, topology, semantics and appearance of common urban objects. They can be modelled in five different Levels of Detail (LOD) with respect to their geometry and spatial accuracy (Gröger et al., 2009). As a widely adopted standard, CityGML enables reusability and interoperability of 3D models over different applications. Till now, it is mostly used for modelling of the buildings due to their dominant role within the urban environment and lack of data for other thematic objects such as road infrastructure, underground networks and utilities, water bodies, etc. (Beil and Kolbe, 2017). CityGML influenced the INSPIRE Directive of the European Commission (EC), which aims at the creation of a European Union spatial data infrastructure providing public sector data in an interoperable way (European commission, 2007).

This paper presents the preliminary results from creating a CityGML 2.0 compliant 3D model of the city of Sofia. It explains the 3D transformation of proprietary geospatial data into CityGML schema. A 3D model of terrain and buildings at district scale is created, covering the territory of district Lozenets of Sofia city. A crucial issue is the integration of buildings and the terrain. Problems arise due to float of the buildings over or sink into the terrain. Thus, an interpolation of building's footprints is performed. Additionally, the buildings' features are enriched with address information. A web application for user interaction with the 3D model is developed, including silhouetting a building on mouse hover and mouse click and showing overlay content, displaying shadows and styling of the buildings.

The 3D city model and its visual representation are the first step towards implementation of City Digital Twin flagship project of Big Data for Smart Society (GATE) Institute. The project aims to create a city digital twin platform for planning, design, exploration, experimentation and optimization of urban processes and services. It involves close collaboration between academia, Municipality of Sofia and IT companies.

The rest of the paper is organised as follows. Section 2 presents the technological framework used for initial implementation of the city digital twin platform. Section 3 points out data used and describes the study area. Section 4 deals with data transformation and generation of CityGML 2.0 compliant 3D model of district Lozenets, covering terrain and buildings. Section 5 presents the obtained results. Section 6 concludes the paper and gives directions for future work.

2. BACKGROUND

City Digital Twin is an interdisciplinary flagship project of GATE Institute in Future Cities application domain that aims at developing a city digital twin platform for designing, testing, applying and servicing the entire lifecycle of the urban environment. The core of the platform is a semantically enriched

* Corresponding author

3D model of the city. Simulation, analytical and visualization tools will be developed on top of it enabling the basic idea of the digital twin – “design, test and build first digitally”. The layered

technological framework used for initial implementation of the platform is shown in Figure 1.

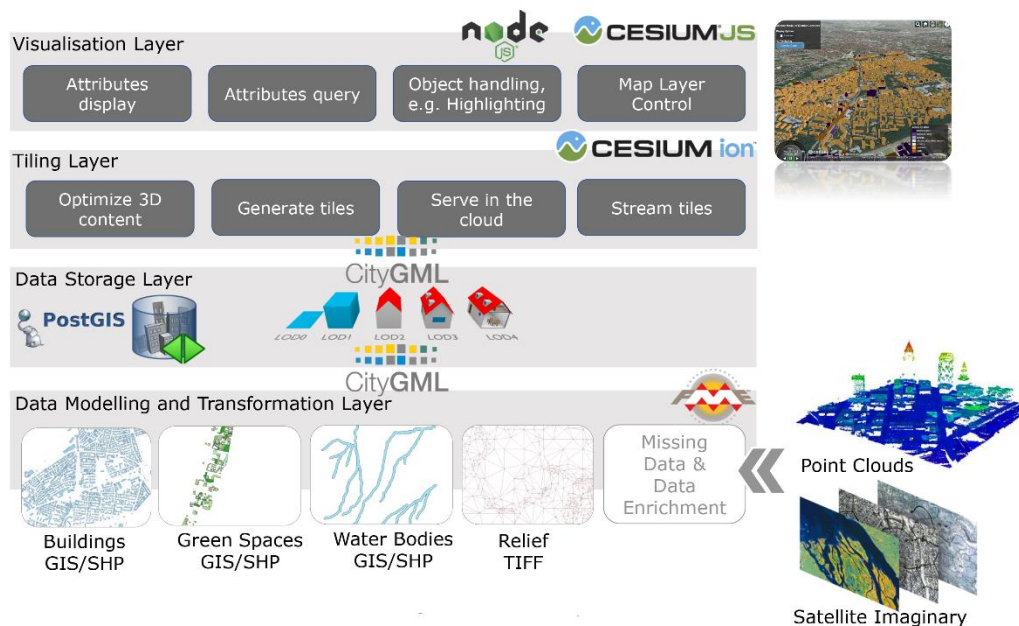


Figure 1. Layered technological framework.

The development of the platform has started with implementation of CityGML compliant 3D model. CityGML standard is chosen since it is based on GML, and thus can be used with variety GML-compatible web services for data access, processing, and cataloguing, such as Web Feature Services, Web Processing Services, and Catalogue Services. Along with the 3D geometry, 3D topology, semantics, and visual appearance, CityGML supports explicit relationships and component hierarchies between objects. That is why a lot of applications, including urban planning, environment analysis, 3D cadastres and complex simulations, could be implemented based on it (Gröger et al., 2009).

The initial development of the 3D model is based on cadastral data, which is appropriate for modelling of buildings, green spaces, relief, road network and other CityGML objects in LOD1. Feature Manipulation Engine (FME) plays the most integral part in data transformation, because of its transformation methods and ability to read, convert and write various data types. Satellite imagery and point cloud data will be further used for semantic enrichment of the 3D model as well as for urban analysis, such as cadaster validation and urban change detection. The 3D model is stored in a 3D City Database (3DCityDB) (Yao et al., 2018), which is implemented on PostgreSQL/PostGIS database.

Cesium ion serves the 3D model in the cloud, allowing to be optimized, tiled and streamed to any device. CesiumJS is used for implementation of a web application for visualisation due to support of rich functionality such as attributes display and query, object handling, such as highlighting, map layer control, etc. The web application is hosted on a local web server, which is set up with Node.js – an asynchronous event-driven JavaScript runtime (Chhetri, 2016).

In order to show the potential of the city digital twin platform, two use cases of the 3D model are defined and are in process of implementation. The first one is related to urban planning. The

main idea behind it is to develop an integrated tool for *parametric urban design*, which is based on preliminary defined neighbourhood indicators related to population, green areas, transport connectivity, etc., construction rules and constraints. A second use case is defined for *analysis and simulation of air quality*, focusing on pollution dispersion depending on the wind direction and velocity as well as the geometry (shape and height) of the buildings.

3. STUDY AREA AND DATA SOURCES

This section describes the study area and data sources used for its 3D modelling based on CityGML 2.0. Since semantics on constituting geometries is not required for the implementation of the currently defined use cases of the city digital twin platform, the 3D model is developed in LOD1. LOD1 provides a volumetric model with an optimal ratio between the cost and possible uses.

3.1 Study Area

Sofia is the capital and the largest city of Bulgaria, with a population of nearly 1.27 million. It consists of 24 districts. Although some preliminary work was carried out using city-wide datasets, the district of Sofia, named Lozenets, was chosen as a case study for the generation of the 3D city model.

Given its heterogeneity in shape, structure and characteristics, the district of Lozenets was deemed adequate to represent a good case study. It is located on a hill, south of the old town of Sofia and extends to the northern foothills of Vitosha Mountain. It covers an area of 9.24 km². Two small rivers flow through the territory and almost 30% of this area is covered by forests. That is why Lozenets is called the greenest area of Sofia city.

A significant part of the territory of the district is occupied by low-rise residential buildings among trees and shrubs (neighbourhood Lozenets). At the same time, there are regions with intensive construction, where problems with accessibility and availability of the road infrastructure, public spaces and public services arise (e.g., neighbourhood Krastova vada).

3.2 Data Sources

The data sources were provided for research purposes by Sofiaplan, which is a municipal enterprise responsible for the spatial and strategic planning of Sofia Municipality. Table 1 provides description of the data sources.

Type of object	File type	Geometry/ Data type	Num. objects	Description
Budlings	.SHP	Polygon	6566	Building footprints, number of floors above the ground, function, etc.
Addresses	.SHP	Point	4371	Addresses points and locality
Terrain	.TIFF	Float32	1	Digital Terrain Model
Surface	.TIFF	Byte	1	Digital surface model

Table 1. Data sources.

The buildings' data is stored in PostGIS database and exported in .shp format. In addition, a Digital Surface Model (DSM) and Digital Elevation Model (DEM) were provided in .tiff format. Fig. 2 shows the visualization of source data in QGIS. The relief is presented with grey colour, while the buildings' footprints are coloured in blue. The coordinate reference system of source data is BGS2005 / CCS2005 (Bulgaria Geodetic System 2005, EPSG: 7801), which is the one generally used by the city of Sofia.



Figure 2. Visualisation of source data (DTM and buildings).

The buildings have 11 attributes, such as cadastre region, function, floor count above the ground, apartments count,

footprint area, etc. The addresses are described with 12 attributes, including district, neighbourhood, street name and number, postal code, etc.

4. 3D MODELLING

This section explains the modelling of relief and building with a special focus on their intersection and integration of address information. The main steps of the modelling process are shown in Figure 3. The blue boxes show the input files, while the green ones – the files with the intermediate transformations. The generated CityGML objects are presented with yellow boxes.

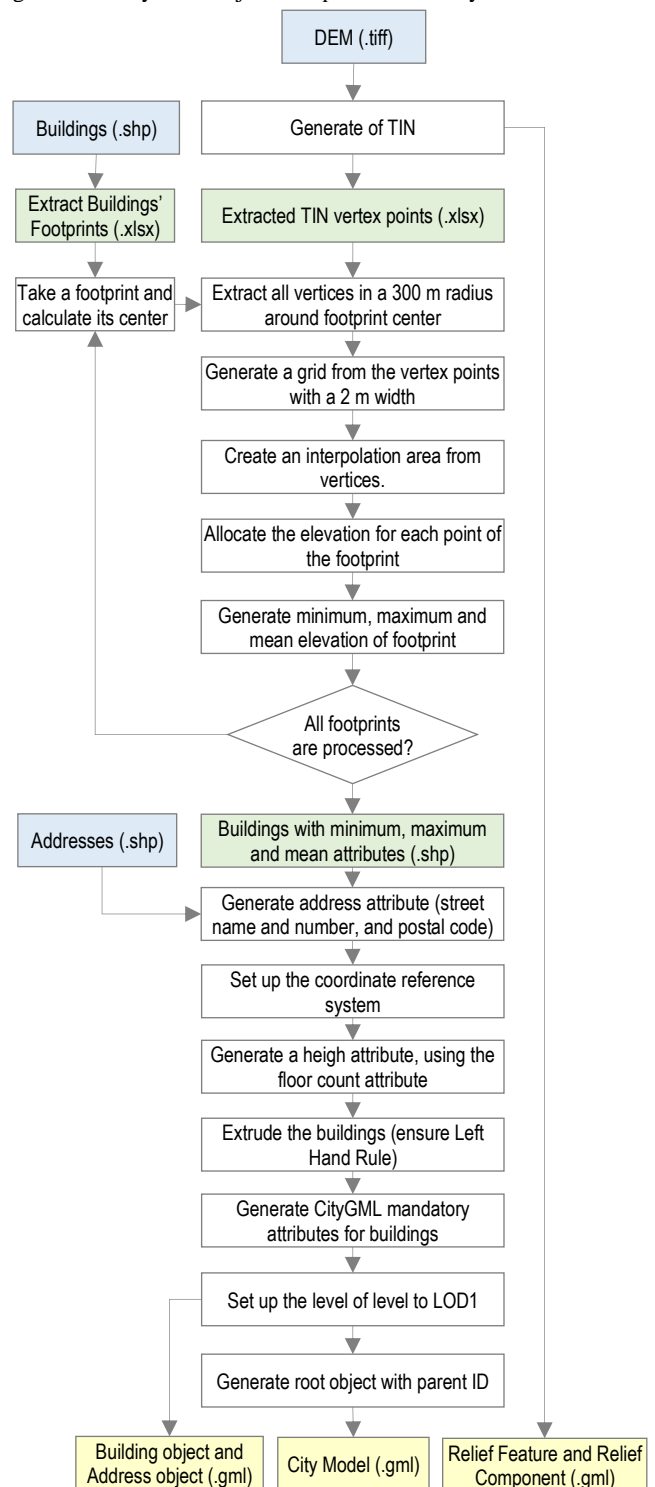


Figure 3. Modelling process.

4.1 Relief Construction

The development of the 3D model starts with the terrain modelling, since it is necessary for correctly extrusion of the buildings' footprints into the third dimension and to reduce their degree of freedom regarding translation (note, that the buildings do not contain height information).

From a digital perspective, the ground surface is commonly modelled by means of DTM (Brandli, 1996), such as TIN that is an alternative to the dense grid DEM to present terrain surface (Lee, 1991). The surface is represented as non-overlapping contiguous triangular facets that are with irregular sizes and shapes, describing elevation changes of terrain. There are other alternative names such as digital height model (DHM), digital ground model (DGM) and digital terrain elevation model (DTEM) (Yan et al., 2019).

Initially, a FME workspace for the relief is created. It uses the georeferenced DEM as input and creates a CityGML 2.0 compliant model, describing the relief of the study area. A Triangulated Irregular Network (TIN) is generated using the "TINGenerator" transformer. The network is divided into two smaller workflows. The main one is responsible for generating the relief itself, while the secondary one extracts all the vertex points from the TIN and writes them in an .xlsx file. The latter file is used for interpolating the buildings footprints. Coming back to the main workflow, a few things are necessary to generate a CityGML compliant model. First, a root object describing the city model itself is created. Its identifier (ID) attribute serves as a parent ID for every other object in the model. The relief is presented with two objects, namely *Relief Feature* and the *Relief Component*. The *Relief Feature* object represents an entire object from the Relief Module of CityGML 2.0 standard, which consists of several parts – the *Relief Components* objects. This is useful also for situations where there are holes in the data or when the components possess different resolutions. The FME workbench can be seen in Appendix (Workbench 1).

4.2 Interpolation of Buildings' Footprints

A small FME workbench is developed to extract the geometry of the buildings' footprints and export it in .xlsx format. This is done using the "Geometry Extractor" transformer. The obtained polygons of the buildings' footprints are needed to interpolate their elevation.

The buildings' polygons and vertex points of the relief available as 3D points (the buildings' footprints have a Z-coordinate equal to 0) are used for interpolation in MATLAB. The process is divided into three steps implemented by corresponding code scripts as follows:

- reading data and putting it into matrices;
- interpolation;
- writing data back so that it can be imported into FME workbench.

Two matrices are generated, representing the vertex points and the polygons, respectively. Every row in a matrix contains one column for the ID and three more columns for coordinates, whereas every coordinate from a polygon is written on one row. Thus, the resulting matrix with vertices has a size of [4477x4], or 4477 points. The polygons are represented through a matrix with size[6566x892], which corresponds to 6566 buildings with up to 297 points in one polygon.

After the data has been prepared, the interpolation can be conducted. During the interpolation the following steps are performed for each polygon:

- A polygon is taken from the matrix and its centre is calculated.
- All vertices in a 300 m radius for that centre are extracted. This ensures that vertex points will be available for every side of the polygon, which is important for the interpolation. In this way, the interpolation is performed locally and generating a TIN representing the entire area for every polygon is avoided.
- A grid is generated from the vertex points with a specified width. In this case it is 2 m. When the grid is smaller, the calculation time increases without significantly increasing accuracy. A reason for that is the vector density of the area.
- The vertices are used to create the interpolation area through the "griddata" function of MATLAB, whereas the method of interpolation is the natural neighbour method.
- The polygon is divided into its points and for each point the elevation is allocated separately. This is done by comparing the horizontal coordinates of the points with the corresponding coordinates in the grid. The information about the Z-coordinate on the grid is then transferred to the points.

In cases of polygons lying outside of the grid, no interpolation is conducted. Instead, the elevation of the nearest vertex is taken.

In addition to the height allocation, three new attributes representing the minimum, maximum and mean elevation of every building footprint, are generated. This information is later used to adjust the buildings' footprints, since they are no longer flat. The situation is illustrated in Figure 4.

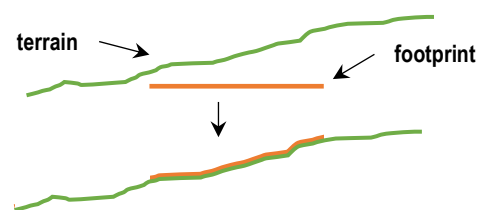


Figure 4. Interpolated building's footprint.

The last script creates an .xlsx file, which contains five columns – polygon id, geometry in gml format, and three columns for the new elevation attributes.

A FME workbench is created for replacing the old geometry of the buildings with the new one and adding the additional attributes. Input datasets are the original buildings in .shp format and the output from MATLAB. Both datasets are initially merged. The geometry is replaced with the "Geometry Replacer" transformer, where the encoding is set to gml. Lastly, a "Writer" is used to write a new .shp file, identical to the old one but with Z-coordinates corresponding to the real elevation.

4.3 Buildings' Addresses

A FME workflow is created to assign an additional information about the addresses of the buildings to their polygons. Since significant number of the buildings are garages or similar constructions, a lot of polygons do not receive additional address attributes. The addresses in .shp format are provided as input to

the workbench together with the .shp file of the buildings. The geometrical representation of the addresses is point. For this reason, it is appropriate to allocate a specific address to a building when its point geometry lies inside of the polygon of the corresponding building. Figure 5 shows a snapshot of data in FME.



Figure 5. Address point and buildings' footprints.

The “Neighbour Finder” transformer is used to connect both .shp datasets. The constraints on the transformer are set in a way that every polygon can have one address, which point must be inside of the polygon. While merging the features of buildings and addresses, the geometries of the buildings are kept, and the address information is added as additional attributes. As a result, around a half of the polygons are enriched with addresses. The process is repeated, since there are address point geometries falling outside of the buildings' polygons. In the second implementation, a tolerance of 5 m is specified. Thus, addresses' points, which are very near to the polygons are connected. Analogous to the previous FME workbench, the output dataset is also a .shp file, which is identical to the input one, but enriched with address information.

4.4 Buildings Construction

The last FME workbench deals with the transformation of the buildings as well as the addresses into CityGML 2.0. It takes as input the recently generated .shp file of the buildings and an .shp file containing information about one of the highest buildings in the study area (hotel). The second file is needed, since the original input .shp file contains a base footprint of the hotel, which actually consists of two parts with a significant difference in the corresponding heights (see Figure 6). That is why, additional .shp file is used to obtain the footprint of the higher part. A transformation is performed to produce more realistic view of the hotel's construction.

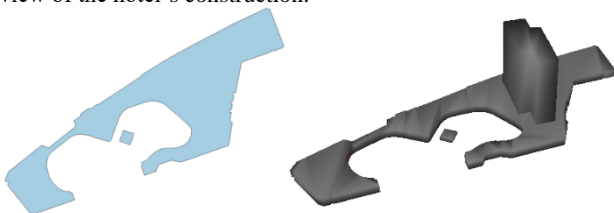


Figure 6. Footprint of the hotel (left) and 3D view after transformation (right).

Since information about the heights of the buildings is missing, they are calculated approximately to the number of floors. In this case, each floor represents a 3m change height. In order to extrude the polygons of the buildings into the third dimension the “Area Builder” transformer is first used. It builds areas, which serve as basis for the volumetric entities.

A minimum Z-value is set for all buildings, which represents their lowest points. The buildings are extruded with the

“Extruder” transformer using a value, which corresponds to the difference between the maximum and minimum elevation of the footprint in addition to the calculated height based on the number of floors (see Figure 7). The attributes `max_z` and `min_z` represent the highest and lowest points of the footprint's elevation, respectively.

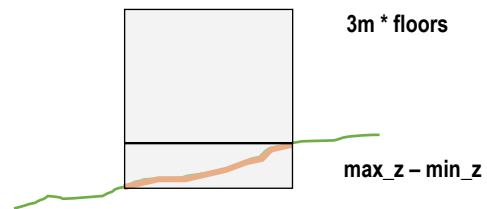


Figure 7. Example of an extruded building.

The difference between the values of `max_z` and `min_z` is added to the resulting height of the building due to the following reasons:

- Achieving flat roofs after the interpolation of the footprints, which are no longer flat. Note, that the transformer extends every point by an equal amount.
- If the maximum height is taken as a basis, there are buildings, which lose the intersection with the terrain.
- If the minimum height is taken as a basis, there are buildings with one floor that partly disappear into the terrain.

Thus, this difference between the values of `max_z` and `min_z` serves as a kind of a building foundation. Once the volumetric objects are generated, a series of attributes are mapped to the buildings. Those attributes are generally necessary for objects of the Building Module in CityGML 2.0 and describe the classes and functions of the buildings as well as their corresponding descriptions. A mapping table between the buildings' functions, defined in the input dataset and class and function attributes, defined in the CityGML 2.0 standard is created. The table is loaded into an “Attribute Value Mapper” transformer.

The unique IDs and the correct geometry are essential for the creation of a CityGML 2.0 compliant model. The geometry is set to a LoD1 solid, since this level corresponds to a rectangular cuboid with no additional features such as a roof or a facade. At this point, the objects in the workflow are ready for the creation of the CityGML file. First, a root object representing the city model is created. Next, the workflow is split into two parts – one for the buildings themselves and another for their addresses. This is necessary because the addresses are not implemented as buildings' attributes. They have their own schema – extensible Address Language (xAL) (OASIS, 2002). A Writer for the buildings is created, where every building has a parent ID attribute containing the root ID. The geometry of the objects is removed for the second scenario. Additionally, a template for the xAL schema is added. Furthermore, the ID attribute is changed to parent ID, so that the address belongs to the building. The address itself does not require identification. A separate Writer is created for the addresses.

5. VALIDATION AND RESULTS

The final result, including both buildings and relief models is presented in Figure 8.

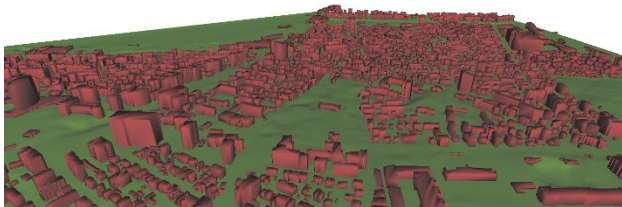


Figure 8. 3D buildings and relief model of District Lozenets.

The quality of semantic 3D models is essential, especially when they will be used for analysis and simulations. The presented 3D models of the buildings and terrain are validated using the 3DCityDB Importer/Exporter tool. The tool checks whether the models are actually compliant with CityGML 2.0 XML schemas and can be further imported into 3DCityDB. The CityGML XML schemas are integrated in the Importer/Exporter and cannot be changed by the user. External XML schemas are only considered if unknown XML content is available (Yao et al., 2018). Both Relief and Buildings models are successfully validated. The covered CityGML classes are shown in Figure 9 and Figure 10 respectively.

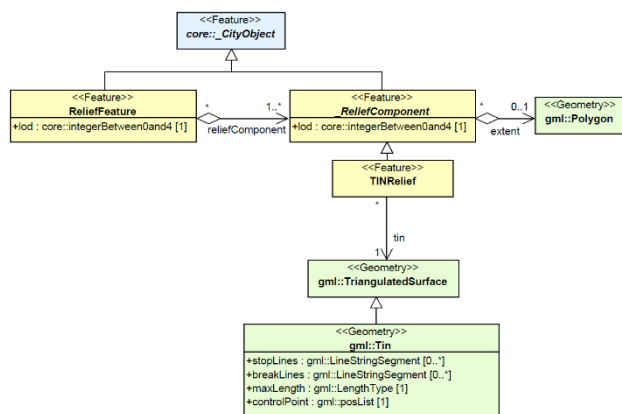


Figure 9. UML diagram of the Relief model.

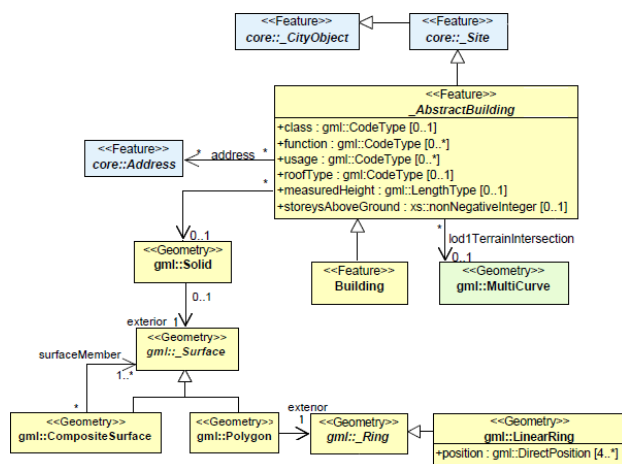


Figure 10. UML diagram of the Buildings model.

The web application developed for visualizing the CityGML building model of district Lozenets is shown in Figure 11. The 3D content is accessed through its asset identifier (AssetId), provided by Cesium ion.

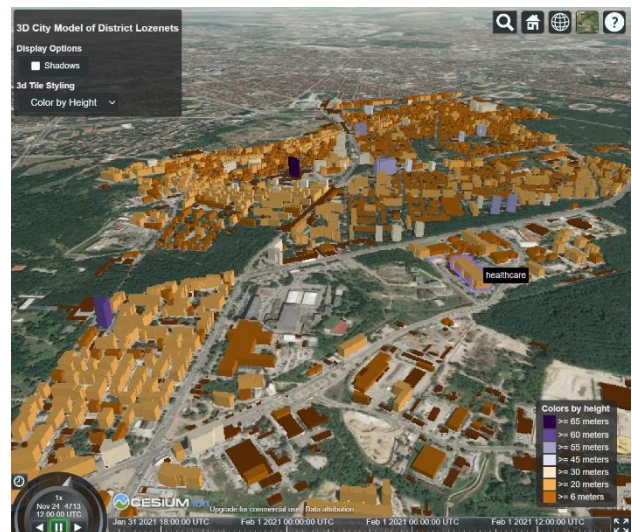


Figure 11. Visualisation of the 3D buildings model.

The following functionality is implemented for user interaction:

- Silhouette a building on mouseover and show its class as overlay content;
- Silhouette a building on selection and show its class, function, floor count and height in an information box;
- Show shadows depending on the current time;
- Show buildings in different colours depending on their height, class and latitude;
- Show buildings in transparent style;
- Show buildings with height over 50 meters.

6. CONCLUSIONS AND FUTURE WORK

The rapid adoption of the Digital Twin concept in city domain to support planning, analytics and community engagement, led to the need for purpose-built 3D models, which successfully integrate 3D objects with the terrain. In this paper, a CityGML 2.0 compliant 3D city model is developed, covering terrain and buildings objects. The intersection of the buildings with the terrain (footprint and DTM) is modelled as address information to the buildings is added. A limitation of the approach is that the intersection of the buildings with the terrain is achieved through interpolation of their footprints and the TIC is not actually presented in the 3D model. The 3D model of the buildings is hosted on Cesium ion platform and visualized through a simple web application, developed with CesiumJS. The main contribution of the paper is that a complete solution for building 3D city model is proposed, starting from raw data, implementing all data transformations needed to obtain LOD1 model and finally visualising the model in web application allowing user interaction. The developed FME workbenches are reusable, meaning that they could be used for generation of 3D models using different input data for other area of interest. They can be easily extended to enable generation of higher LODs.

In future work, further development of 3D city model is considered in order to include additional types of objects such as green spaces, roads, pedestrian network, street lights, etc. A TIC computation is considered, properly integrating 3D objects and terrain. A challenge would be the modelling of the underground infrastructure such as electricity and gas pipe networks.

ACKNOWLEDGEMENTS

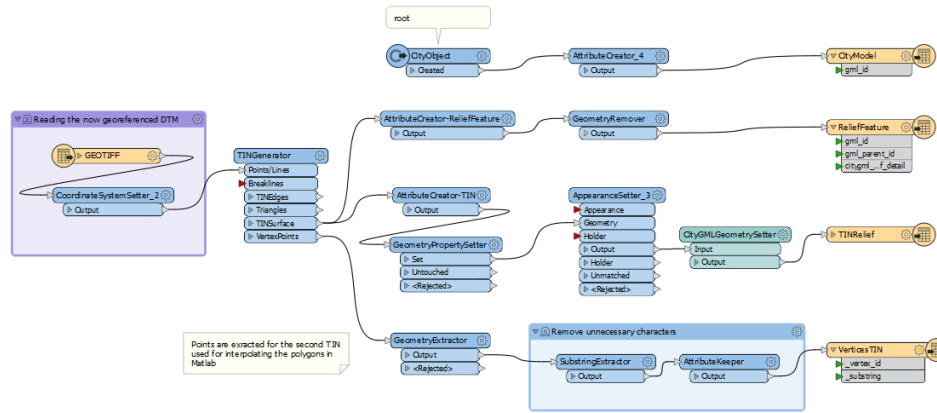
This research work has been supported by GATE project, funded by the Horizon 2020 WIDESPREAD-2018-2020 TEAMING Phase 2 programme under grant agreement no. 857155, by Operational Programme Science and Education for Smart Growth under Grant Agreement No. BG05M2OP001-1.003-0002-C01 and by the Bulgarian National Science fund under project Big4Smart agreement no. DN12/9.

REFERENCES

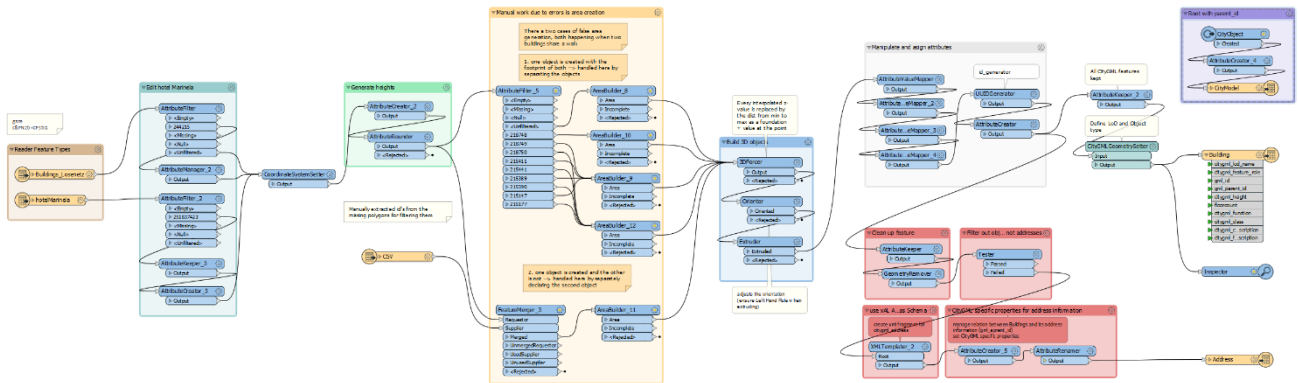
- Arroyo Ohori K., Biljecki F., Kumar K., Ledoux H., Stoter J.: 2018: Modeling Cities and Landscapes in 3D with CityGML. In: Borrmann A., König M., Koch C., Beetz J. (eds) *Building Information Modeling*. Springer, Cham. doi.org/10.1007/978-3-319-92862-3_11
- Arroyo Ohori K., Ledoux H., Stoter J., 2015: A dimension-independent extrusion algorithm using generalised maps. *Int. J. Geogr. Inf. Sci.*, 29(7), 1166–1186. doi.org/10.1080/13658816.2015.1010535.
- Beil C., Kolbe T. H., 2017: Citygml and the Streets of New York - a Proposal for Detailed Street Space Modelling. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W5, 9-16. doi.org/10.5194/isprs-annals-IV-4-W5-9-2017.
- Besuiuevsky, G., Patow, G., 2014: Recent advances on LoD for procedural urban models. *Proceedings of the 2014 Workshop on Processing Large Geospatial Data*, Cardiff, UK.
- Biljecki F., Stoter J., Ledoux H., Zlatanova S., Çöltekin A., 2015: Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information*, 4(4):2842-2889. doi.org/10.3390/ijgi4042842.
- Blaschke T., 2010: Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing* 65 (1), 2-16. doi.org/10.1016/j.isprsjprs.2009.06.004.
- Brandli, M., 1996: Hierarchical Models for the Definition and Extraction of Terrain. *Geographic objects with indeterminate boundaries*, 2, 257.
- Chhetri, N., 2016: A Comparative Analysis of Node.js (Server-Side JavaScript). *Culminating Projects in Computer Science and Information Technology*. 5. core.ac.uk/download/pdf/232792216.pdf
- European Commission: 2007: INSPIRE Directive. inspire.ec.europa.eu/about-inspire/563 (29 January 2021).
- Götz, M., 2013: Towards generating highly detailed 3D CityGML models from OpenStreetMap. *International Journal of Geographical Information Science* 27 (5), 845-865. doi.org/10.1080/13658816.2012.721552.
- Gröger, G., Kolbe, T.H., Nagel, C., & Häfele, K., 2012: OGC City Geography Markup Language (CityGML) Encoding Standard.
- Julin A., Jaalama K., Virtanen J-P., Pouke M., Ylipulli J., Vaaja M., Hyypä J., Hyypä H., 2018: Characterizing 3D City Modeling Projects: Towards a Harmonized Interoperable System. *ISPRS International Journal of Geo-Information*. 2018; 7(2):55. doi.org/10.3390/ijgi7020055.
- Lee, J.: 1991: Comparison of existing methods for building triangular irregular network, models of terrain from grid digital elevation models. *International Journal of Geographical Information System* 5, 267-285.
- Lewis R., Séquin C., 1998: Generation of 3D building models from 2D architectural plans. *Computer-Aided Design*, 30 (10), 765-779. doi.org/10.1016/S0010-4485(98)00031-1.
- Li W., Zlatanova S., Diakite A.A., Aleksandrov M., Yan J., 2020: Towards Integrating Heterogeneous Data: A Spatial DBMS Solution from a CRC-LCL Project in Australia. *ISPRS International Journal of Geo-Information* 9(2):63. https://doi.org/10.3390/ijgi9020063.
- Morton, P.J., Horne, M., Dalton, R., & Thompson, E., 2012: Virtual City Models: avoidance of obsolescence. *Proceedings of the 30th eCAADe Conference in Digital Physicality*, Prague.
- OASIS, eXtensible Address Language, docs.oasis-open.org/election/external/xAL.xsd (29 January 2021)
- Over, M., Schilling, A., Neubauer, S., Zipf, A., 2010: Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany. *Computer, Environment and Urban Systems* 34(6), 496-507. doi.org/10.1016/j.compenvurbsys.2010.05.001
- Shahzad, M., & Zhu, X., 2015: Robust Reconstruction of Building Facades for Large Areas Using Spaceborne TomoSAR Point Clouds. *IEEE Transactions on Geoscience and Remote Sensing* 53, 752-769. doi.org/10.1109/TGRS.2014.2327391.
- Smelik, R., Tutenel, T., Bidarra, R., & Benes, B., 2014: A Survey on Procedural Modelling for Virtual Worlds. *Computer Graphics Forum*, 33(6), doi.org/10.1111/cgf.12276.
- Tomljenovic I., Höfle B., Tiede D., Blaschke T., 2015: Building Extraction from Airborne Laser Scanning Data: An Analysis of the State of the Art. *Remote Sensing* 7(4):3826-3862. doi.org/10.3390/rs70403826
- Wang, Y., Cheng, Y., Zlatanova, S., & Palazzo, E., 2020: Identification of physical and visual enclosure of landscape space units with the help of point clouds. *Spatial Cognition & Computation* 20(3), 257-279. doi.org/10.1080/13875868.2020.1767625.
- Yao, Z., Nagel, C., Kunde, F. et al., 2018: 3DCityDB - a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. Open geospatial data, software and standards, 3(5). https://doi.org/10.1186/s40965-018-0046-7
- Yan, J., Zlatanova, S., Aleksandrov, M., Diakite, A. A., Pettit, C., 2019: Integration of 3D Objects and Terrain for 3D Modelling Supporting the Digital Twin. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* IV-4/W8, 147-154. doi.org/10.5194/isprs-annals-IV-4-W8-147-2019.
- Yin, X., Wonka, P., & Razdan, A., 2009. Generating 3D building models from architectural drawings: a survey. *IEEE computer graphics and applications* 29(1), 20–30. doi.org/10.1109/mcg.2009.9

APPENDIX

The appendix shows the FME workbenches developed for modelling the terrain and buildings.



Workbench 1. FME workbench of terrain modelling.



Workbench 2. FME workbench of buildings' modelling.