

SYSTEMS ARCHITECTURE FOR MANAGEMENT OF BIM, 3D GIS AND SENSORS DATA

M. Aleksandrov^{1,*}, A. Diakité¹, J. Yan¹, W. Li¹, S. Zlatanova¹

¹ UNSW Built Environment, Red Centre Building, Kensington NSW 2052, Sydney, Australia
(mitko.aleksandrov, a.diakite, jinjin.yan, wei.li, s.zlatanova)@unsw.edu.au

Commission IV, WG IV/10

KEY WORDS: BIM, GIS, UML, Database, Visualisation

ABSTRACT:

This paper presents a system architecture for structuring and manipulation of Building Information Models (BIM), three-dimensional (3D) geospatial information, point clouds and time series data obtained from sensors. The system consists of four layers including data pre-processing, data structuring and storage, system interface and front-end data manipulation. To enable the integration of different data, a unified UML model is developed. The paper explains all steps of 3D reconstruction, BIM georeferencing, storage of spatial data and visualisation. Special attention is given to the integration of sensors data. The data model and the system architecture are tested for a university campus. The results demonstrate an approach for BIM-GIS-Sensor integration as part of Precinct Information Modelling (PIM). The system architecture allows for a flexible structuring and manipulation of different spatial data towards managing various 3D spatial and non-spatial data.

1. INTRODUCTION

Recent advances in technology allow for a quick collection of 3D data and reconstruction of 3D realistic models (Sequeira et al., 1999, Suveg, Vosselman, 2004). These come in addition to detailed 3D BIM models, which are becoming increasingly available. This results in a strong demand for the integration of these data with the existing GIS solutions (Amirebrahimi et al., 2016, Isikdag, Zlatanova, 2009, El-Mekawy, Östman, 2010, El Meouche et al., 2013, Ohori et al., 2017). Such integration can enable the connection between a geometrically and semantically rich model like BIM and georeferenced spatial information which model successfully outdoor information. There have been several approaches tackling this problem and dealing with some of the main components (Irizarry et al., 2013, Kang, Hong, 2015). However, data and models are scattered in different file formats and maintained by different departments and institutions, which complicates the update and analysis of data.

Apart from GIS and BIM, various other geospatial data are also available and could be interconnected in some way. A typical example are point clouds (Dore, Murphy, 2012) and sensors real-time information (Park et al., 2016) which could be also integrated with 3D models. This can be extremely beneficial for precinct information modelling - PIM (Trubka et al., 2016) combining and analysing all possible geospatial information in one place.

In this paper, we propose a system architecture for structuring and manipulation of BIM and 3D GIS data as part of PIM, as well as the integration of sensor information with these data. The paper is organised as follows: Section 2 explains briefly the system architecture. In Section 3, the unified UML data model is showed supporting integration of different data. The

case study is presented in Section 4. Section 5 highlights the conclusions and future work suggestions, where potential aspects that needs improvement are identified.

2. SYSTEM ARCHITECTURE

Spatial data can come from different sources, as 2D or 3D, and in different file formats such as Open Street Map, IFC, CityGML, Shape, DXF. It can have different types of geometry (e.g., point, line, polygon, solid, point cloud), and attributes. In order to put in place a system that manages geospatial data in a consistent and coherent way, an appropriate data model and a system architecture are needed. This paper elaborates on a system architecture considering four major layers namely data preprocessing, data storage and structuring, interface and front-end.

The first layer covers the preprocessing steps which are necessary to bring the input data in a structured way. The most common steps are related to geometric cleaning and editing, creating 3D representations, georeferencing, and attributes extraction.

The second layer relates to structuring of data and database organisation. This layer could be considered as the most important one, bringing together different spatial data in one organised data structure. Therefore, we propose a unified UML data model establishing all necessary connections between various data sources.

The third layer represents an interface that establishes a connection between the database and a front-end application. This component enables querying, processing and update of structured data.

The fourth and last component of the system architecture explains the front-end layer. Within this layer data visualisation and manipulation could be performed.

*Corresponding author

3. DATA MODEL

In this section, we describe the 3D unified model at precinct scale, which encapsulates classes from both the CityGML and IFC standards. PIM has been firstly introduced in the study for the Australia CRC for Low Carbon Living (Newton et al., 2013) and we continue to use this appellation in our work.

To build the PIM, all classes and related concepts are based on the CityGML and IFC. The principle used is: overlapping concepts were merged, new objects are created in some cases to ensure capturing of both indoor and outdoor objects, and spatial relationships between the objects are redefined. Unified Modeling Language (UML) is used for representing objects and relationships between them.

Due to most of outdoor objects being similar to the CityGML thematic classes, we have adopted them in the PIM. The following thematic models are presented in an integrated model: Building, Sensor, Vegetation, CityFurniture, LandUse, Transportation, and Terrain.

In order to enhance the readability of the UML diagrams, classes are depicted in different colours considering different standards and data sources. Classes in green colour are adopted from IFC4 (Liebich, 2013) and their class names are preceded by the prefix *Ifc*, classes coloured in yellow belong to the CityGML thematic classes, while in purple objects which represent sensors and statistic datasets are showed. Figure 1 highlights the conceptual design of PIM where different colours are used to represent different elements which are briefly described below. The base class of all above thematic classes

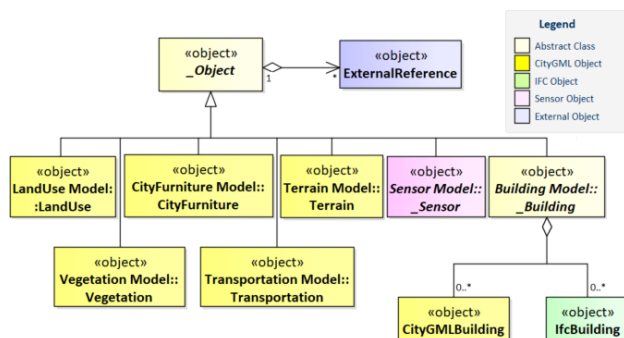


Figure 1. UML diagram for conceptual design of PIM.

in our PIM is the abstract class *_Object*, which is an abstract class and provides generic attributes (e.g., ID, creation and termination date for the management of histories of objects) of all kinds of objects in PIM. Class *ExternalReference* in Figure 1 represents a link to any possible external information, through which, rich data (such as energy) can be imported into our model to strengthen understanding of resource management. Subclasses of *_Object* comprise several different thematic fields corresponding to practical built environment. In terms of abstract *_Building*, it covers both CityGMLBuilding and IfcBuilding model (The detail of these two models can be found in our supplementary material.). On the other hand, *_Sensor* as an abstract subclass of *_Object* may have linking attributes referenced to building or room objects.

4. CASE STUDY

As a case study area, the UNSW campus is selected. Among all available data, we have concentrated on four types: point

clouds, GIS (e.g., 2D or 3D spatial data), BIM and sensors information. The UNSW campus is located on a hill, which requires correct representation of all objects with respect to the terrain. Therefore, the height values of all objects are preprocessed to fit the terrain surface. The total number of buildings is 96 as 6 of them have a BIM model. LOD1 is created for all buildings, where these simplified buildings representations are suitable for establishing the link to some of the sensor data. The sensor data (Figure 2) come from Greensense system¹ (i.e., energy consumption per building) and MyAir system² (i.e., air quality data) are used. Point clouds of the UNSW campus are obtained from ELVIS³ online system. Archibus⁴ data, storing metadata of rooms, are used as an external reference to collect some additional attribute to rooms of buildings.

The identified system architecture for this project is presented in Figure 2.

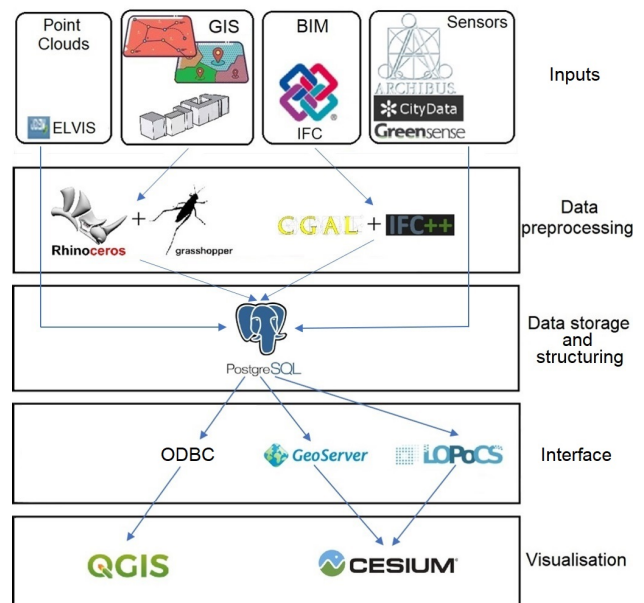


Figure 2. System architecture for this project.

As mentioned before, data preprocessing is the first step to bring different data together. Since most of the identified GIS data is mainly in 2D we reconstruct 3D objects from it. Rhinoceros⁵ (with Grasshopper) is used to perform the entire reconstruction. We import BIM (obtained as IFC files) models using the IFC++⁶ library and process the geometric, topological and semantic information using CGAL library (cgal Project, 2016).

PostgreSQL is selected to store and structure the data (Postgresql, 2019). Structuring of identified spatial and non-spatial data is a key component of the system architecture. Considering the unified UML model connections between the identified data are established.

¹<https://www.estate.unsw.edu.au/campus-developments/greensense-live-energy-project> (retrieved on 27/05/2019)

²<https://citydata.be.unsw.edu.au/layers/geonode/%3Amyair> (retrieved on 27/05/2019)

³<http://elevation.fsdf.org.au/> (retrieved on 27/05/2019)

⁴<https://archibus.unsw.edu.au/> (retrieved on 27/05/2019)

⁵<https://www.rhino3d.com/> (retrieved on 27/05/2019)

⁶<http://ifcquery.com/> (retrieved on 27/05/2019)

The fourth layer of the system architecture involves defining the interface. Thus, Geoserver is used for manipulation of 2D data and simple 3D models on the web. For visualisation of more complex 3D models, py3dtiles library is used (Oslandia, 2019b) as part of LOPoCS server (Oslandia, 2019a). In this way, all previously structured data can be obtained from the database and forwarded to the web-visualisation layer.

In terms of desktop visualisation, QGIS is used (QGIS Development Team and others, 2016), while for web visualisation Cesium platform is selected (Cesium, 2019a). Thus, users can visualise BIM models and spatial information via different platforms.

4.1 Data preprocessing

Data preprocessing is a key step to bring the identified 2D/3D geospatial data into a data structure as implemented in the database.

As mentioned above, we need to identify the third-dimension for 2D data. As a main source for this process, a terrain generated from point clouds is used. Thus, data representing BIM models, 2D floorplanes, roads, green areas and trees are draped to the terrain. For visualisation purposes, 3D reconstruction of buildings is also performed.

The following section presents the 3D reconstruction of buildings (LOD1 according to CityGML), streets, green areas, trees, and their integration with a Digital Terrain Model (DTM). Also, the process of BIM and GIS integration is presented discussing the steps of how to store IFC models into a database and address the georeferencing problem.

4.1.1 GIS Object pre-processing

The automatic reconstruction of 3D building models was investigated by many researchers in the last two decades (Gruen et al., 1995). The main concept involves capturing building roofs and footprints at the required detail and accuracy to generate a 3D representation (Haala, Kada, 2010). At the same time, 3D urban models should not only contain 3D buildings, but also 3D terrain, especially for hilly places, since buildings and other objects are presented on it (Li et al., 2004). In this work, buildings, roads/streets, green areas and terrain are considered for the reconstruction, and trees are integrated with the terrain surface. It should be noted that all data mentioned below is geo-referenced and transformed (aligned) in the same coordinate system.

Terrain reconstruction

The terrain is reconstructed into two steps: 1) initial terrain is built based on point clouds and 2) terrain is re-calculated considering of all objects. To avoid topological issues (e.g., 3D objects partially floating over or sinking into the terrain) the terrain needs to be rebuilt by using the terrain intersection curve (TIC) (Gröger et al., 2012) as constraints. Constrained Delaney Triangulation (CDT) is used for this purpose, which is built by using the vertices of TIC as Points and its edges as Breaklines. This type of terrain is relative simple (contains the min number of triangles) and is more appropriate for large precinct areas.

3D buildings reconstruction

For the purpose of this project we concentrated on reconstruction of LOD1 buildings. We use 2D footprints of

buildings and point clouds as inputs to obtain the needed models. Point clouds are utilised to (i) compute heights of buildings and (ii) generate the initial terrain. The process of 3D building reconstruction and integration with terrain consists of four steps as follows:

- Extracting and generating the initial terrain from point clouds;
- Projecting building footprints to the terrain;
- Setting height for side surfaces representing walls;
- Constructing roof and floor based on the extruded and projected footprints, respectively;
- Merging roof, walls, and floor as enclosed volumes to represent 3D buildings.

Surface objects and terrain integration

In this project, roads and green areas are two types of surface objects. The original data is 2D footprints (polygons). We reconstruct them into 3D surfaces (rather than 2D polygons) by draping their footprints on the terrain. Using this approach, we ensure that these object follow the terrain and therefore do not partially float over or sink into the terrain. The reconstructed surface objects also follow the LOD1 concept of CityGML.

Trees and terrain integration

Trees in this work are represented as discrete points (i.e., one tree is one point). The initial points of trees are distributed on the same XY plane. Therefore, we project points modelling trees to the terrain.

4.1.2 BIM pre-processing

The integration of BIM and GIS environment is known to be a challenging process (Liu et al., 2017). Here we want to address the geo-referencing issue which allows to geometrically align BIM and GIS models in a same environment. Usually BIM models are coming from the Computer Aided Design (CAD) world and focus on the entities that they represent. Thus, when it comes to coordinate systems, their geometry is described in relation to the local coordinate system of the software used to generate them. The general method to solve this problem is to manually select identified corresponding points between a geo-referenced support and the one not geo-referenced, and the adapted affine transformation is sought. Because it is not always possible to find corresponding points easily, we adopt an automatic approach that allows to project BIM geometries onto the Coordinate Reference System (CRS) of a map.

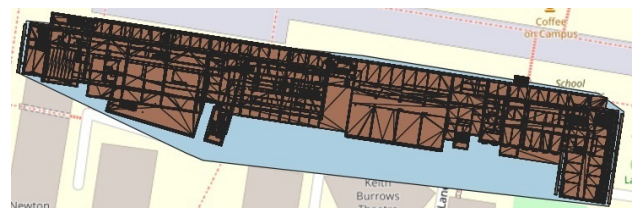


Figure 3. Geo-referenced IFC model of the Red Centre. BIM model on top of the blue convex hull of the OSM feature used in the process.

The method involves finding corresponding points between a geo-referenced footprint and a BIM model of the same building. After that, the affine transformation is applied identifying the translation and rotation for the BIM model. Since BIM is an aggregation of several 3D objects, we use 2D convex hull of the projected IFC geometries. Similarly, we compute the convex hulls of the available 2D features coming from Open Street Maps (OSM). Since the transformation that we are seeking is affine, we convert the coordinates of the map features from degrees to meters. In order to find the affine transformation, we combine two approaches: the approximative rigid matching of (Goodrich et al., 1999) and the closed-form absolute orientation method of (Horn, 1987). Figure 3 shows a geo-referenced BIM model after applying identified affine transformation.

It is worth to mention that the precision of our automatic geo-referencing method depends on the 2D map feature used. When the latter fits well to the projected IFC geometries, centimetre accuracy can be reached for the position matching. In the model in Fig. 3, the maximum error between the matched shapes is of 1.2m. This is because the OSM polygons do not fully correspond to the real projection of the building. Thus, for a better precision, it is recommended to rely on more accurate map features.

Thanks to this process, the retrieved geometric information from the BIM models are put in their correct geographical context and stored along with their semantic information in the database. This enables querying the necessary information from the GIS interface.

4.2 Data structuring and storage

Based on the conceptual design for PIM in Section 3, we develop our own PIM for the UNSW campus modelling all objects we have.

In Figure 4, the subclasses of *_CampusObject* comprise different thematic classes, in the following covered by separate thematic data models: building model (named *_Building*), digital terrain model (named *Terrain*), road model (named *Road*), tree model (named *Tree*), lawn model (named *Lawn*),

and sensor model (named *_Sensor*). Further, sensor model contains three different data sources including *Energy*, *Air Quality* and *Room Property* class shown in the UML diagram as subclasses of *_Sensor*. Among them, *Energy* is a class which stores consumptions of gas and electricity of each building in UNSW campus, while air quality data of each room coming from MyAir system. In addition, the UNSW ARCHIBUS database includes attributes for floors and rooms with the intention to identify individual spaces, which plays an important role in the connection with the recorded statistical data corresponding to each room in the building.

Among all thematic classes, the building class is the special one consisting of GIS, BIM and sensor data in our PIM. To build a unified building data model, all classes with their concepts are collected from both *CityGMLBuilding* and *IFCBuilding*. We list all the attributes of each class and the relationships established by corresponding attributes. For instance, *CO2* (carbon dioxide) emission of some rooms is presented through mapping between “*ifc_name*” in *IfcBuilding* and “*room*” in *Air Quality*. However, matching semantics between ontologies originally designed independently of each other is to expect.

Employing Spatial Relational Database Management System (Spatial-RDBMS) is the state-of-the-art solution to store and manage complex 3D city/urban model. Spatial-RDBMS which the open source software PostgreSQL supports with the help of PostGIS extension (Postgis, 2019) has extensive capabilities in handling 3D spatial data and managing all required geometry types and provide means for proper spatial indexing as well as for geometric and topological analyses. For example, volumes from the IFC models or 3D reconstructed models can be represented as POLYHEDRALSURFACE objects. Each data type contains a spatial reference identifier (SRID) to describe the coordinate system as well. Using POLYHEDRALSURFACE over other possibilities (e.g. MULTIPOLYGON Z) allows the use of more PostGIS geometric functions with the SFCGAL extension (Oslandia, 2019c).

Figure 5 illustrates the information stored in the database of an IFC model. Along with the geometry, it includes the

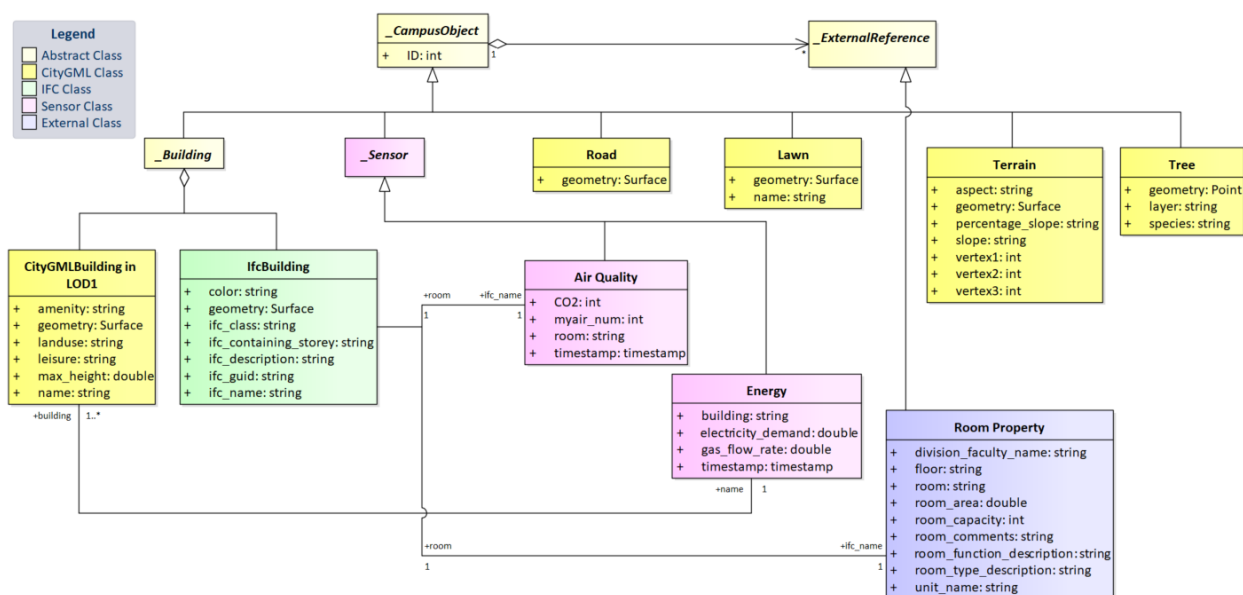


Figure 4. UML diagram of our PIM.

unique IDs (corresponding to the IfcGuid attribute for the BIMs), the semantic (class names for IFC, e.g., IfcSpace, IfcWall, etc.), the name which is optional (e.g., component name and reference from the manufacturer, room number, etc.), the description (optional as well), the storey level, and finally colour information for each component coming from the BIM model. Regarding trees, geometry data type POINT is

	ifc_class	ifc_guid	ifc_name	ifc_description	ifc_containing_storey	color	geom
1	IfcSpace	1tkTUen...	1-2	Room	FIRST FLOOR	rgb(130, 130, 130)	010F0...
2	IfcSpace	1tkTUen...	1-3	Room	FIRST FLOOR	rgb(130, 130, 130)	010F0...
3	IfcSpace	1tkTUen...	1-16	Room	FIRST FLOOR	rgb(130, 130, 130)	010F0...
4	IfcSpace	1tkTUen...	1-17	Room	FIRST FLOOR	rgb(130, 130, 130)	010F0...
5	IfcSpace	1tkTUen...	1-18	Room	FIRST FLOOR	rgb(130, 130, 130)	010F0...

Figure 5. Example of a table created in PostgreSQL to store a IFC model.

selected storing all three coordinates. The triangulated mesh representing the terrain uses MULTIPOLYGON Z geometry data type.

Following is a data interaction example via QGIS which shows the coherent semantical-geometrical modelling. In this example, we try to search all buildings which are within 100 meters from the Red Centre building. The SQL query and visualisation result can be seen in Figure 6.

```
SELECT B.building_name,
ST_3DDistance(B.geom, R.geom) AS distance,
ST_CollectionExtract(B.geom,3) AS geometry
FROM building B, building R
WHERE R.building_name='Red Center - H13'
AND ST_3DDWithin(B.geom, R.geom, 100)
AND B.building_name!=R.building_name
AND B.building_name IS NOT NULL;
```

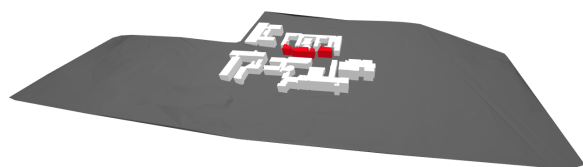


Figure 6. Visualisation result of above query via QGIS. Object painted in red belong to the 'Red Center' building. Query results are in white.

4.3 Interfaces and front-ends

Interfaces enable connection between a database and a web or desktop application. Geoserver is the most famous sever for establishing a web-server connection. It enables fast creation of Web Feature Services (WFS), which store and structure spatial and non-spatial information within it. However, visualisation of complex 3D data such as BIM models becomes too slow. The problem is related to the way of handling 3D spatial data, which is the same as for 2D data, i.e., neglecting the complexity in z-direction. Therefore, we have used 3D Tiles as suggested by Cesium platform. The main idea behind it is to create chunks of data (tiles), which can be loaded separately in a web application. Geoserver uses geojson files which requires parsing of every character, while 3D Tiles are files, which are space-efficient and quicker to read. 3D Tiles exist in two formats. The first can store multipolygons or solids (i.e., b3dm

files), while the second one can save only point cloud data (i.e., pnts files). Currently, LOPoCS server enables distribution of pnts files to a web application. For b3dm files, the server does not provide support yet. As a result, b3dm files representing 3D geospatial data are just exported using the py3dtiles library.

BIM models, 3D extruded buildings, terrain and roads are exported from the Spatial RDBMS and organised as b3dm files. Geoserver is used for data which are represented as points in the database like trees as well as for sensor data.

4.4 Visualisation

Data visualisation is an important component of the system architecture, providing an option to users to interact with the data. In this regard, Cesium and QGIS are selected for web and desktop visualisation, respectively.

Cesium is currently the only option for visualisation of geospatial 3D content on the web. It is recommended platform for visualisation of 3D GIS and BIM (Chen et al., 2018, Cesium, 2019b). Cesium is currently the only free 3D globe environment to visualise 3D objects. It also provides an option to select a terrain (Cesium World Terrain), which provides to some extent a realistic representation of the ground in 3D. However, this option is not used due to discrepancies with the terrain generated from point clouds. Cesium also provides an option to style the geospatial data for different time periods creating animation-like visualisation. In our case, we use this option for energy consumption visualisation of several buildings, and for air quality presentation within some rooms. An example of such time-lapse animation is presented in Video 1⁷. In the left-top corner, a panel with different datasets is positioned where users can select different datasets to present them on the globe. Feature selection presenting its attributes is enabled as well.

The option for desktop visualisation is QGIS. It has been predominantly used for analysis, manipulation and visualisation of 2D content. However, several options exist to visualise 3D content, one of them is made possible with the version 3. To illustrate the advantage of the unified model and the management of integrated DBMS, we have executed, in the following text, several queries within QGIS.

4.4.1 Visualisation cases

As previously mentioned, Cesium provides a terrain to users which is not correct for several meters and this difference is not consistent for one area. Therefore, the terrain built from point clouds is used and all other geospatial content is aligned with it. In this way, the absolute position of all 3D objects is presented in space.

A few available BIM models of the UNSW campus are brought into the integrated BIM-GIS system. Some of the identified models are more complete having more features to load up, while others have only building exteriors. For example, the size of the BIM before the import into the database for the Round House building is 227 MB, while the BIM for Dalton building is only 2 MB. BIM models' features are predominantly represented with fewer colours and no texture (Figure 7). In QGIS, the BIM models are visualised with some basic styling (Figure 8).

⁷available at <https://vimeo.com/336699901> (retrieved on 27/05/2019)

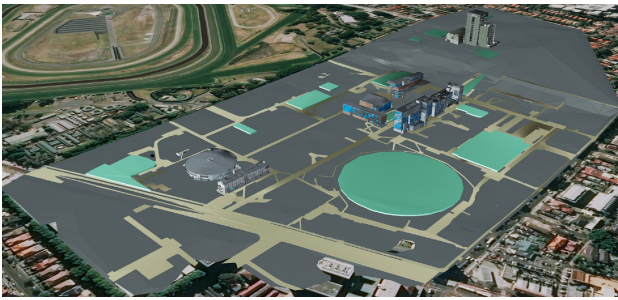


Figure 7. Visualisation of several BIM models, roads and green areas in Cesium.

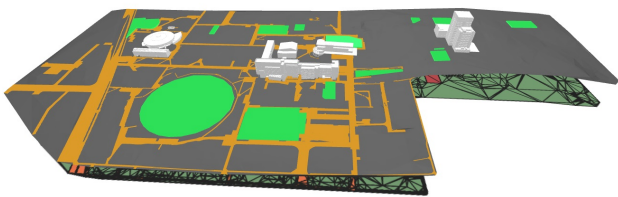


Figure 8. Visualisation of several BIM models, roads and green areas in QGIS.

LOD1 buildings are used explicitly for the time-dynamic styling in Cesium, showing different colours based on a specific measurement. For instance, showing energy consumption at some interval. Content representing green areas and pedestrian roads are slightly extended in the third dimension for visualisation purposes (Figure 9 and 10).



Figure 9. CityGML - LOD1 for buildings, pedestrian roads, green areas and trees in Cesium.



Figure 10. CityGML - LOD1 for buildings, pedestrian roads, green areas and trees in QGIS.

The dataset representing trees is visualised as 3D points in Cesium and QGIS. In terms of Cesium, a request is sent to Geoserver creating a WFS out of the data, which content is automatically visualised on the Cesium globe using geojson file

format.

The point clouds are streamed using LOPoCS server directly to Cesium. The data is a combination of LiDAR data and high-resolution images downloaded from Google Earth creating coloured point clouds (Figure 11). They have been imported in the Spatial-RDBMS as well, but they have been maintained as separate record, not related to the 3D objects. To create the 3D Tiles 2D subdivision is performed by the server. However, this approach is not particularly suitable for points which are distributed in Z direction.



Figure 11. Visualisation of coloured point cloud (approximately 1 million points).

4.4.2 Time-related data visualisation and animation

There are two data types showing changes through time for a specific parameter. Thus, energy consumption is shown for four buildings considering a period of three months (i.e., October – December 2018) downloaded from the Greensense system (Figure 12). Based on this, the timeline in Cesium is set up to show this period, in which case users could drag the time cursor left or right to a specific period. The time is speeded up 900 times compared to real one. The main reason behind that is the timestamp of the collected data of 15 minutes (i.e., 900 seconds). The identified range is between 0 (dark blue colour - no consumption) and 450 (red colour - high consumption), and it is used to style the extruded buildings.

Regarding the air quality data, which are obtained from the MyAir system, the timestamp is the same. The data present CO_2 coming from 54 sensors placed in the same number of rooms in the Red Centre (West Wing) building (Figure 13). Data are only available for December 2018. In this case, spaces representing rooms are styled based on the identified minimum (375) and maximum (1450) value from the data as well as the current measurement.

5. CONCLUSION AND FUTURE WORK

In this paper, we present a system architecture for structuring, analysis and visualisation of BIM, 3D GIS, point clouds and sensor data. The system consists of four layers, where several software packages, in-house software tools and libraries are used to preprocess the data, Postgres database and its extension PostGIS are used to store the data, Geoserver and LOPoCS server are utilised to provide web services, Cesium platform is used to visualise the content on the web and QGIS is used to

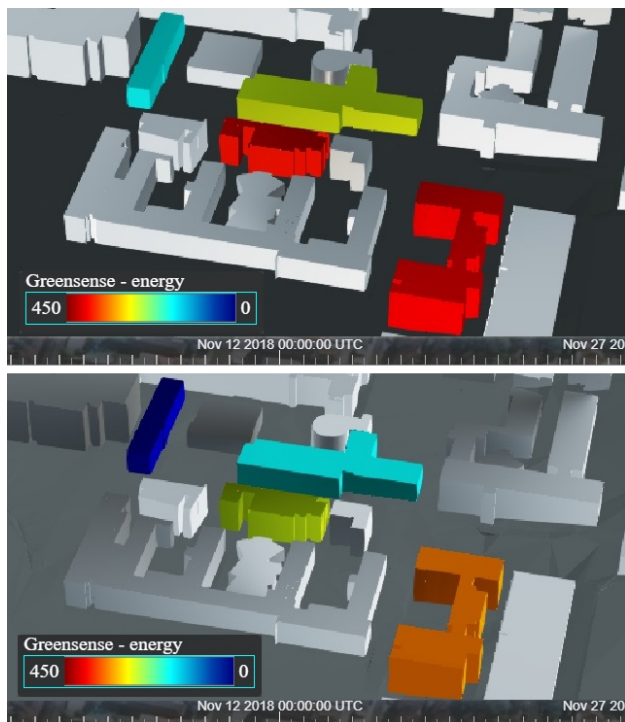


Figure 12. Energy consumption for 4 buildings at two different time slots.

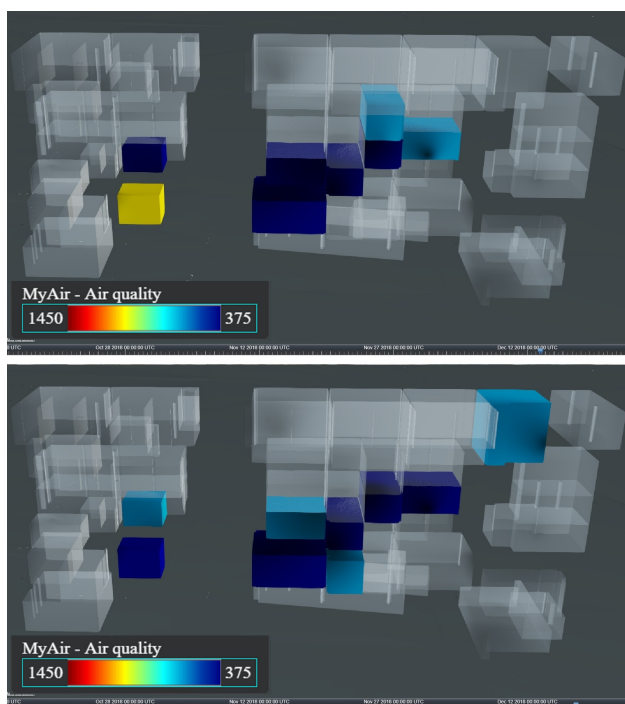


Figure 13. Air quality at two different time slots for several rooms represented as spaces and different colours. For instance, yellow colour means higher concentration, while dark blue shows lower concentration of CO_2 .

query, edit and visualise data on desktop. Overall, the approach enables successfully BIM-GIS-Sensor integration.

Our experiments clearly revealed that the complexity of TIC and terrain are mutually dependent. That is, a detailed terrain

can lead to more complex TIC and the complex TIC can bring more details to the re-built terrain. Specifically, the TIC is computed based on the initial terrain, while the CDT for the new terrain is taking the vertices of TIC as Points and its edges as Breaklines constrains. At this stage, we have considered only LOD1 objects and their terrain intersection curves. Further research is needed to investigate the link between terrain and LOD2 and LOD3 objects.

Organising BIM models in Spatial RDBMS is not only possible but also beneficial for the integration with any other information such as sensor data. Storing BIM models in a database gives the opportunity to query a subset of data to view/analyse, which is a prerequisite for its flexibility and efficiency. Our approach also ensures automatic georeferencing prior the import in the database. Future work will concentrate on developing a data structure, which can represent the hierarchy of BIM models considering the properties of all construction components, and allow to establish more references between the BIM models and the outdoor objects such as terrain, roads and trees as well as to enable more elaborated analysis.

We expect that such system architecture will have advantages in facilitating storage, manipulation and exchange of data. However the integrated model is on a very preliminary stage. New applications would pose challenges to the object type and their relationships and attributes. It is inefficient to keep modifying classes. Modularity and extensibility should be leading in the further work on the conceptual model. As the amount of BIM models increases, we need to strengthen the scalability of that data model as well.

In this research we have used freeware servers, but they still require further developments to serve an integrated BIM-GIS-Sensor model properly. The capabilities of LOPoCS server are currently insufficient and should be extended to support real-time distribution and manipulation of b3dm tiles. At the same time, the server should be able to provide support for Insert, Update and Delete of information coming from the client side. As mentioned before, the subdivision of space in Z direction to create 3D Tiles can be beneficial for BIM models and point clouds.

Spatial-RDMS as Postgres and its extensions provide a rich set of data types, indexing approaches and geometric functions and operations. However the front-end such as Cesium make a little use of this functionality. Cesium should be able to provide a direct connect to the database and benefit from the data structure.

ACKNOWLEDGEMENTS

This research (No.RP2011u1) is funded by the CRC for Low Carbon Living Ltd supported by the Cooperative Research Centres program, an Australian Government initiative. The publication acknowledges the contribution of all authors, researchers, participants and others as appropriate.

REFERENCES

Amirebrahimi, Sam, Rajabifard, Abbas, Mendis, Priyan, Ngo, Tuan, 2016. A BIM-GIS integration method in support of the assessment and 3D visualisation of flood damage to a building. *Journal of spatial science*, 61, 317–350.

- Cesium, 2019a. Cesium: an open-source JavaScript library for world-class 3D globes and maps. <https://cesiumjs.org/> (retrieved on 27/05/2019).
- Cesium, 2019b. Cesium v1.35. <https://cesium.com/blog/2017/07/05/cesium-version-1.35-released/> (retrieved on 27/05/2019).
- cgal Project, 2016. Cgal user and reference manual.
- Chen, Yiqun, Shooraj, Erfan, Rajabifard, Abbas, Sabri, Soheil, 2018. From IFC to 3D Tiles: An Integrated Open-Source Solution for Visualising BIMs on Cesium. *ISPRS International Journal of Geo-Information*, 7, 393.
- Dore, Conor, Murphy, Maurice, 2012. Integration of historic building information modeling (hbim) and 3d gis for recording and managing cultural heritage sites. *2012 18th International Conference on Virtual Systems and Multimedia*, IEEE, 369–376.
- El-Mekawy, Mohamed, Östman, Anders, 2010. Semantic mapping: an ontology engineering method for integrating building models in ifc and citygml. *3rd ISDE DIGITAL EARTH SUMMIT, 12-14 June*.
- El Meouche, Rani, Rezoug, M, Hijazi, Ihab, 2013. Integrating and managing BIM in GIS, software review. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2, W2.
- Goodrich, Michael T, Mitchell, Joseph SB, Orletsky, Mark W, 1999. Approximate geometric pattern matching under rigid motions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, 371–379.
- Gröger, Gerhard, Kolbe, Thomas H, Nagel, Claus, Häfele, Karl-Heinz, 2012. OGC city geography markup language (CityGML) encoding standard.
- Gruen, Armin, Baltsavias, Emmanuel, Henricsson, Olof, 1995. *Automatic extraction of man-made objects from aerial and space images (II)*. Springer Science & Business Media.
- Haala, Norbert, Kada, Martin, 2010. An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65, 570–580.
- Horn, Berthold KP, 1987. Closed-form solution of absolute orientation using unit quaternions. *Josa a*, 4, 629–642.
- Irizarry, Javier, Karan, Ebrahim P, Jalaei, Farzad, 2013. Integrating BIM and GIS to improve the visual monitoring of construction supply chain management. *Automation in construction*, 31, 241–254.
- Isikdag, Umit, Zlatanova, Sisi, 2009. Towards defining a framework for automatic generation of buildings in citygml using building information models. *3D Geo-Information Sciences*, Springer, 79–96.
- Kang, Tae Wook, Hong, Chang Hee, 2015. A study on software architecture for effective BIM/GIS-based facility management data integration. *Automation in Construction*, 54, 25–38.
- Li, Zhilin, Zhu, Christopher, Gold, Chris, 2004. *Digital terrain modeling: principles and methodology*. CRC press.
- Liebich, Thomas, 2013. IFC4—The new buildingSMART standard. *BuildingSMART Inter*.
- Liu, Xin, Wang, Xiangyu, Wright, Graeme, Cheng, Jack, Li, Xiao, Liu, Rui, 2017. A state-of-the-art review on the integration of Building Information Modeling (BIM) and Geographic Information System (GIS). *ISPRS International Journal of Geo-Information*, 6, 53.
- Newton, Peter, Marchant, David, Mitchell, John, Plume, Jim, Seo, Seongwon, Roggema, Rob, 2013. Performance assessment of urban precinct design. *Low Carbon Living CRC, Sydney, NSW*.
- Ohori, Ken Arroyo, Biljecki, Filip, Diakit , Abdoulaye, Krijnen, Thomas, Ledoux, Hugo, Stoter, Jantien, 2017. Towards an integration of gis and bim data: what are the geometric and topological issues? *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, Proceedings of ISPRS 12th GeoInfo Conference, Melbourne, Australia*, 26–27.
- Oslandia, 2019a. LOPoCS (Light OpenSource PointCloud Server). <https://github.com/0slandia/lopocs> (retrieved on 27/05/2019).
- Oslandia, 2019b. py3dtiles: a Python tool and library for manipulating 3D Tiles. <https://github.com/0slandia/py3dtiles> (retrieved on 27/05/2019).
- Oslandia, 2019c. SFCGAL: a C++ wrapper library around CGAL for 3D operations. <https://github.com/0slandia/SFCGAL> (retrieved on 27/05/2019).
- Park, JeeWoong, Kim, Kyungki, Cho, Yong K, 2016. Framework of automated construction-safety monitoring using cloud-enabled BIM and BLE mobile tracking sensors. *Journal of Construction Engineering and Management*, 143, 05016019.
- Postgis, 2019. PostGIS: spatial database extension to PostgreSQL. <https://www.postgis.net/> (retrieved on 27/05/2019).
- Postgresql, 2019. PostgreSQL: the world’s most advanced open source relational database. <https://www.postgresql.org/> (retrieved on 27/05/2019).
- QGIS Development Team and others, 2016. QGIS geographic information system. *Open Source Geospatial Foundation Project*.
- Sequeira, Vitor, Ng, Kia, Wolfart, Erik, Gonçalves, Joao GM, Hogg, David, 1999. Automated reconstruction of 3D models from real environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54, 1–22.
- Suveg, Ildiko, Vosselman, George, 2004. Reconstruction of 3D building models from aerial images and maps. *ISPRS Journal of Photogrammetry and remote sensing*, 58, 202–224.
- Trubka, Roman, Glackin, Stephen, Lade, Oliver, Pettit, Chris, 2016. A web-based 3D visualisation and assessment system for urban precinct scenario modelling. *ISPRS Journal of Photogrammetry and Remote Sensing*, 117, 175–186.