

A GEO-DATABASE SOLUTION FOR THE MANAGEMENT AND ANALYSIS OF BUILDING MODEL WITH MULTI-SOURCE DATA FUSION

W. Li^{1*}, S. Zlatanova¹, J. J. Yan¹, A. Diakite¹, M. Aleksandrov¹

¹ Faculty of the Built Environment, The University of New South Wales, NSW 2052, Australia

{wei.li, s.zlatanova, jinjin.yan, a.diakite, mitko.aleksandrov }@unsw.edu.au

Commissio n IV, ICWG IV/III, WG IV/4

KEY WORDS: Database, Built Environment, Data Management, Data Analysis, Data Fusion

ABSTRACT:

Over the last decades, building models have become valuable for a multitude of application scenarios, such as visualisation, simulations and decision support. As the growth of multi-source data consisting of semantic and 2D/3D spatial information, data management becomes feasibility means for facilitating the development and deployment of building model applications. In addition, most studies focus on modelling buildings at geometric level, while semantic analysis can become a promising approach to get a better understanding of built environment. How to utilize multi-source data in a joint manner to further express the building model, therefore, is an emerging challenge. In this paper, we develop a semantic 3D building model based on complex multi-source data. Then, we tackle data management and analysis problems in a geo-database solution for our unified building model. Performance studies on the University of New South Wales (UNSW) campus demonstrate the efficiency of our solution.

1. INTRODUCTION

Due to the strong expressive power of visual building model, many real-world regions model 3D geometry and graphical characteristics, as well as spatial and semantic interrelationships as a semantic 3D building model. Since a broad spectrum of heterogeneous data come from different formats, a unified building model that ensuring consistent and interoperable data structuring is becoming a promising but challenging work. With the proliferation of 3D building model applications, such as navigation, disaster management, energy assessment, significant research efforts have been devoted towards efficiently and effectively managing and analyzing 3D building models that carry rich semantic information. Among them, 3D geo-database is a prominent field to support data fusion, semantic explorations and spatial analysis in order to get better understanding of built environment.

Geo-databases play a fundamental role as data integration, management and analysis tools for (large)geo-referenced 2D and 3D datasets in multiple application scenarios (Zlatanova, 2006). In the meanwhile, 3D city/urban modelling work under different standards (e.g., CityGML and IFC) can be supported by 3D geo-databases to efficiently store and retrieve 3D models. The state-of-the-art geo-database systems extend a traditional relational database management system (RDBMS) by incorporating spatial data types, spatial index and spatial operators in its data model and query language. Furthermore, new concepts developed in the fields of geo-databases can be applied to both 2D and 3D geo-related applications.

The continuous development and maintenance of assets, infrastructure, facilities and logistics at city buildings requires management of a broad spectrum of heterogeneous information. Estate management (EM) department and stakeholders such as companies, councils, institutions, researchers and residents are constantly involved in the use

and exchange of critical information in buildings. Much of this information concerns infrastructural or sensor components that are physically distributed in different departments, which requires to unitary use of data. For instance, the power sector can only realize the statistical results of the power consumption of buildings, however, there is no way for them to make decisions or optimize energy supply through connecting building structure information mastered by EM department.

Majority of current 3D building models are designed by making use of either GIS or BIM data. Some research towards interoperating CityGML and IFC building models have been developed to transform information towards the generation of knowledge and intelligence in recent years (Deng et al., 2016, de Laat, Van Berlo, 2011, El-Mekawy et al., 2011). However, as BIM and GIS data are created, managed and visualized in different ways in terms of coordinate systems, scope of interest, and data structures, data incompatibility is becoming a significant issue. Besides, the sharp and recent increase in the availability of data captured by different data sources combined with their considerably heterogeneous nature opens up the possibility of utilizing multimodal datasets in a joint manner to further improve the performance of the processing approaches with respect to the application at hand. Multi-source data fusion, as a general and popular multi-discipline approach, therefore, has received enormous attention (Ghamisi et al., 2018).

This paper presents a geo-database solution for the management and analysis of building models based on multi-source data fusion. Firstly, we create a unified building model in a joint manner to enrich semantic expression of building object. Then, we conduct case studies on buildings in UNSW campus. The empirical studies confirm that our unified building model improves understanding of built environment.

Contributions. Our primary contributions are as follows:

1. A novel perspective on building models is proposed

*Corresponding author

considering multi-source data fusion for the sake of improving the ability of expression of same built environment from multiple perspectives.

2. A high-level conceptual design and relational database design have been developed for building models in a joint manner.

Organization. The rest of this paper is structured as follows: Section 2 gives a brief retrospect to the data fusion and geo-database research. In addition, the essential aspects and approaches for geo-data modelling using geo-database are discussed in order to provide the foundation for designing a unified compliant relational database schema for a building. Section 3 proposes our building model fusing different data (GIS, BIM, LiDAR, statistics and sensor) with details about conceptual design and relational database design. Section 4 shows our case study on UNSW campus to demonstrate the efficiency of our model. The last section draws the conclusions about the presented work and outlines the relevant aspects of our future research and developments tasks.

2. BACKGROUND AND RELATED WORK

2.1 Smart Built Environment and Data Fusion Opportunities

With the emergence of the concept of “Smart Built Environment”, which refers to a built environment that has been embedded with smart objects, such as sensors and actuators, with computing and communication capabilities, making the environment sufficiently “smart” to interact intelligently with and support their human users in their day-to-day activities (Nakashima et al., 2009), researchers have explored how to design building models hosting multiple information throughout its life cycle and what is the benefit from harnessing BIM or GIS information and capabilities (see (Zhang et al., 2015)).

Currently, an enormous amount of data is produced in a quick span of time in the built environment. How to make these multiple sources data precise and highly accurate is an open problem which needs to be considered since the quality of these information plays an important role in visualization, planning and decision-making. Data fusion (Bleiholder, Naumann, 2009, Dong et al., 2009), which is regarded as a part of data integration, is an effective process of integration of multiple data representing the same real-world object into a consistent, accurate, and useful representation. Figure 1 presents the paradigm of general data fusion. For example, there are several construction datasets for general building domain generated by different data providers. Data fusion aims to merge these datasets into a database with a consistent data schema, through a process of ingestion, duplicate removal and integration. The records (from different datasets) describing the same object, e.g., a commercial building, are generated in the same domain.

2.2 Geo-data Modelling in Geo-databases

Geo-database management system (geo-DBMS) is usually considered to be a traditional commercial Database Management System (DBMS) with extension which incorporates spatial data types and geometry, supports for spatial indexes, and provides functions and operators for analysis and processing of spatial objects. Concepts developed

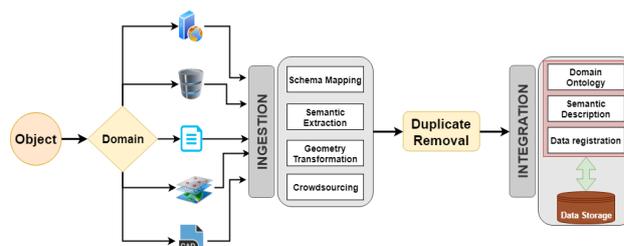


Figure 1. Paradigm of the data fusion

in the fields of geo-databases and spatial databases may be applied to both 2D and 3D geo-applications (Breunig, Zlatanova, 2011).

Applications relying on geo-models have distinct needs: some applications may require models only for visualisation, while others may require models for analysis and statistics. Defining implementations for geo-models that provide efficient storage and retrieval of the models in geo-databases received enormous attention from researchers. The heterogeneity of current geo-spatial geometric and topological data models shows the importance of integration for geo-databases. Figure 2 presents an integrated geo-database. For example, there are multiple data sources with different data formats, such as reports, images, maps, shapefiles. After data pre-processing, GIS, BIM data with spatial data type or geometry are stored in geo-database.

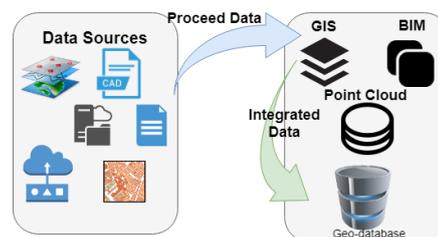


Figure 2. An integrated geo-database

2.3 CityGML and IFC Building Model

City Geography Markup Language (CityGML (Gröger et al., 2012)) is an open standard issued by the Open Geospatial Consortium (OGC) for the representation and exchange of urban information in the GIS domain. Also, it is one of the most prominent semantic 3D modelling formats. The buildingSmart Industrial Foundation Class (IFC), as the most comprehensive and popular exchange format for BIM within the industry, is designed to present the building context (ISO et al., 2013). It is an established generic information exchange standard for BIM. For buildings, CityGML model often focuses on the geographical information and shape of buildings and building components from a geographical perspective. In contrast, IFC model often focuses on the detailed building components, such as architecture and construction perspective (Cheng et al., 2015). With the recent demand for merging outdoor and indoor applications for different purposes, attempts have been made to design methods and tools to integrate building models within a geospatial context (El-Mekawy et al., 2011, El-Mekawy et al., 2012). Overview of integration research have been discussed by several authors (Zhang et al., 2009, Fosu et al., 2015, Ma, Ren, 2017, Liu et al., 2017, Zhu et al., 2018).

3. A NOVEL BUILDING MODEL WITH DATA FUSION

The following sub-sections explain the process of data modelling. Important design decisions are pointed out. The two main steps are marked as conceptual design and relational database design in Figure 3.

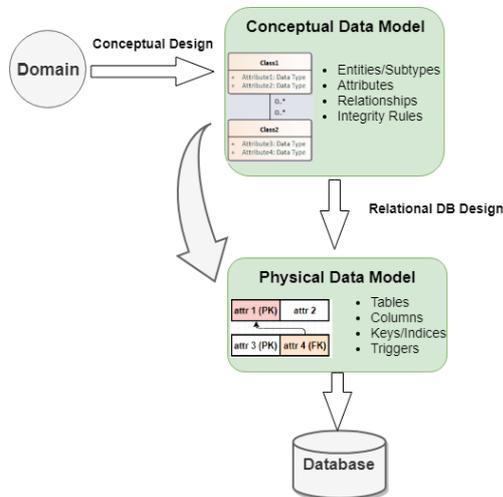


Figure 3. The process of data modeling

1. An Unified Building Model (Section 3.1)
 In order to achieve a more compact database schema and improve query performance, the CityGML and IFC model are combined into a simple and unified model at some critical points.
2. Derivation of The Relational Database Schema (Section 3.2)
 The unified object-oriented data model has been mapped to relational tables. The number of tables was optimized in order to minimize the number of joins for typical queries.

3.1 An Unified Building Model

The design and management of our unified building model generally focus on two aspects: architecture and sensing. The architecture refers to that provided by our model to support the lifecycle of a construction project, including plan, design, construction, operation, and dismantling, also provide analysis and visualization of location-based services. Thus, we adopt IFC model which includes building structures and appearances, as well as CityGML model which is used for spatial analysis based on the functional and physical relationship of outdoor environment. On the other hand, sensing is considered as one essential component of our building model. Home-based sensor-driven services can come into being and stakeholders can make use of outcome of sensor-driven services to trigger more servicesenhance or improve the current design to use resources more wisely. Our novel unified building model hosts the collaborative architectural information and provides the semantic knowledge of the building. With the emergence of smart built environment tendency, our model should further developed to be capable of seamlessly integrating smart objects in building design and feed objects with relevant building-related information.

In this section our slightly unified building model with respect to CityGML and IFC building models is described at the conceptual level using UML class diagrams. This diagram forms the basis for the implementation-dependent realization of the model with a relational database system. In addition, UML diagrams may also form the basis for other implementations e.g., for the definition of an exchange format based on XML or GML. In order to enhance the readability of the UML diagrams, classes are depicted in different colors if they belong to different standards or data sources. Classes in green colour are adopted from IFC4 (Liebich, 2013) and their class names are preceded by the prefix *Ifc*, classes colored in yellow belong to the CityGML thematic classes and classes in purple represent objects, which are mainly from sensors or statistic datasets. Figure 4 shows the conceptual design of our unified building model where different colors are used to represent different elements which are briefly described below.

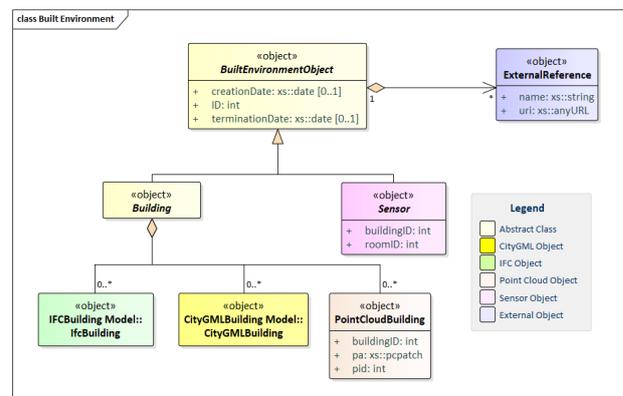


Figure 4. Conceptual design of unified building model

The base class within our unified building model is the abstract class *BuiltEnvironmentObject*, which provides a ID, creation and termination date for the management of histories of objects, as well as external references *ExternalReference* to corresponding objects in external systems. Through *ExternalReference*, rich data (such as energy, statistic data) can be imported into our model which strengthen understanding of building’s resource management to achieve a balance between resource efficiency and construction design. The subclasses of *BuiltEnvironmentObject* comprise two different thematic fields corresponding to aforementioned two aspects – architecture and sensing. In terms of abstract class *Building*, it covers both *IfcBuilding Model* and *CityGMLBuilding Model* (The detail of these two models can be found in our supplementary material.), also *PointCloud* object which carries *buildingID* attributes allowing a reference with respect to the building objects. On the other hand, *Sensor* as an abstract subclass of *BuiltEnvironmentObject* may have attributes *buildingID* and *roomID* referenced to building or room code lists.

3.2 Derivation of The Relational Database Schema

As aforementioned in Section 2, geo-DBMS have been developed in response to new requirements for geo-information applications in recent years. Among them, employing geo-extended relational database management systems (geo-RDBMS) to store and manage complex building models will bring lots of benefits. On one hand, geo-RDBMS support all kind of spatial data type, spatial access method and spatial query language, as well as provide means for high-efficient

spatial indexing structure and for geometric and topological analyses. On the other hand, geo-DBMS play an important role in bridging the geometric modelling of man-made and natural geo-objects. Therefore, it is useful to provide geometric primitives such as points, line segments, triangles, and tetrahedrons for both manmade and natural objects to construct more complex objects consisting of surfaces and solids (Breunig, Zlatanova, 2011). Therefore, geo-RDBMS such as the commercial software ORACLE Spatial/Locator and the Open Source software PostgreSQL with PostGIS extension play a major role for 2D/3D geo-scientific models (Agoub et al., 2016) due to their extensive capabilities in handling 3D spatial data.

The conceptual solution for handling object-oriented data models like CityGML and IFC in geo-RDBMS can be abstracted to solving the problem of mapping the object-oriented data model onto a relational data model. Following mapping rules are adopted in our relational database design procedure:

- **A class shall be mapped into one single table.** The mapped table shall have at least one primary key column to store the object identifier which may be named as “ID” and must be unique within the table. Additional columns can also be added to the mapped table for storing the spatial and non-spatial attribute values of the respective class objects.
- **A foreign key constraint needs to be added in case of 1:1 or 1:N relationship.** For each binary 1:1 or 1:N relationship type, we choose one of relations and include as a foreign key in the chosen relation the primary key of another relation. It is better to identify the relation S that represents the participating at the N-side of the relationship type.
- **An associative table in case of M:N relationship shall be utilized to link the tables mapped from the associated classes.** For each binary N:M relationship type, create a new relation to represent this relationship. Include as foreign key attributes in the chosen relation the primary keys of the relations that represent the participating relations; their combination will form the primary key of the chosen one.
- **A foreign key constraint or an associative table needs to be set for inheritance relationship.** The inheritance relationship between two classes can either be implemented using a foreign key constraint to link the subclass and superclass tables by joining their primary keys or mapped to a table that represents the two inherited classes at the same time.

However, although these mapping rules allow to map building model onto a relation database model, they may easily lead to a large number of database tables due to plenty of building objects (e.g., door, windows, slab, plate, wall and so on) in CityGML and IFC building models. In the meanwhile, different objects may have completely different attribute list and vast attributes may have numbers of null values. Furthermore, this may result in lots of join relations when queries are requested. An analysis of the existing relational database systems indicated that a more compact database schema is much more efficient for querying and processing of large and complex-structured data to facilitate

good performance when interacting with the database in a real-time application (Stadler et al., 2009). To reach this purpose, our database schema shall result from a careful manual process by identifying and simplifying the complex classes and data type and mapping them onto fewer tables with respect to the database interoperability. Concerning this requirement, the types of the attributes are customized to corresponding database (PostgreSQL) data types (see Table 1).

Table 1. Data type mapping

UML	PostgreSQL/PostGIS
String, anyURI	VARCHAR/TEXT
Int	INT/NUMERIC
Double	FLOAT/REAL/NUMERIC
Boolean	BOOLEAN
Date	DATE/TIMESTAMP
Patch	PATCH
GML Geometry	GEOMETRY

Further, we propose a set of fine-grained mapping rules:

- **Mapping classes in inheritance relationship or same hierarchy level into one table.** We assume that, in most cases, subclasses may have or set same attribute list due to data missing or multiple unique attributes make no contribution to special applications. With this consideration, some classes belonging to an inheritance hierarchy can be mapped into one single table, which results in the retrieval of data in all subclasses just needs to perform queries on one table in order to avoid multiple table joins for speeding up the overall performance. This way, the single table allows for rapid retrieving a list of different objects through a query on the category attribute which distinguishes instances objects stored in the table from different types. For detail, we can add an additional column named “OBJECTCLASS_ID” or “OBJECTCLASS_NAME” which can store a numeric value or string value in each row for representing the respective class type.
- **Mapping aggregations and compositions into one table.** Due to our building is object-oriented, aggregation and composition relations of classes can be properly modelled by using a foreign key for joining each classes with its parent class. For special case that recursion appears in aggregation or composition relationships, a single table for mapping of all the involved classes along with their inheritance relationship can be added in database. For detail, we can add an additional column “PARENT_ID” as foreign key which is used for representing the composition relationship.

4. CASE STUDY

As a case study area, the UNSW campus is selected. Among all available data, we have concentrated on three types: GIS (e.g., 2D or 3D spatial data), BIM and sensors information.

The unified building model is stored in an object-oriented database – PostgreSQL/PostGIS in our project, which can be remotely accessed via host address, port, and password. Data in our database can also be accessed via web requests, which allows external application developers to create either web-based or standalone applications that interact with relevant

data entities in our model for particular purposes. Stakeholders are allowed to query and update their objectives against the existing precinct entities in our model.

Further, QGIS software can be used for visualising entities. It can open a model across internet connection to the database server, edit that model and save the amended model either to a model file on the local system or merge it back into the source model on the server.

4.1 Conceptual Design

According to the high-level conceptual design for unified building model in Section 3.1, we develop our specific 3D building model for UNSW campus based on the elements we have. In Figure 5, the subclasses of *_CampusBuildingObject* comprise the different thematic classes related with built environment in UNSW campus, in the following covered by separate data models: building model (named *_Building*), sensor model (named *_Sensor*) and external model (named *_ExternalReference*). In detail, external model receives information from UNSW ARCHIBUS Facilities Management System¹, which includes current attributes for floors and rooms with the intention identify individual working spaces/desks, which plays as statistic data corresponding to each room in the building. Sensor model contains two different data sources which donated by classes *Energy*, *Air Quality* shown in UML diagram as subclass of *_Sensor*. Among them, *Energy* is a class which contains consumption of gas and electricity of each building in UNSW campus, while air quality data of each room coming from MyAir system². The interval is 15 seconds for both sensor equipments.

The building model is the complex one consisting of both GIS and BIM data. To build a unified building model, all classes with their concepts are collected from both CityGMLBuilding and IFCBuilding model. Here, we have simplified two models based on the data we have and listed attributes of each class and the relationships are established by corresponding attributes (such as the analysis of relationship between geometry and sensor information in rooms can be done through mapping between “*ifc_name*” in *IfcBuilding* and “*room*” in *Air Quality*).

4.2 Relational Database Design

Employing geo-RDBMS is the state-of-the-art solution to store and manage complex 3D building model, such as the open source software PostgreSQL with PostGIS³ extension has extensive capabilities in handling 3D spatial data and supports all required geometry types and provide means for proper spatial indexing as well as for geometric and topological analyse. For example, volumes from the IFC models or the reconstructed from the 2D data can be represented as a POLYHEDRALSURFACE object. Each data type contains a spatial reference identifier (SRID) to the describe the coordinate system as well. Using POLYHEDRALSURFACE over other possibilities (e.g. MULTIPOLYGON Z) allows to use more PostGIS geometric functions with the SFCGAL extension⁴.

Figure 6 illustrates the information stored in the database. Along with the geometry, it includes the 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 IFC model. From Figure 6, it is not hard to see that new proposed mapping rules are applied in our case study during relational database design. Due to lack of attribute list of each Ifc building element object (e.g., IFCBEAM, IFCPLATE, IFCROOF, etc.) in our IFC model, we map all these Ifc classes into one table in which different objects can be distinguished by attribute IFC.CLASS. That makes our tables higher cohesive and avoid creating columns for some missing attributes.

Furthermore, GIS data describing a complete building often uses multiple geometries each depicts a component of a building. Therefore, we create table BUILDING to store building ID and NAME, and BUILDINGPART to store semantic information of a building, respectively. A foreign key column BUILDING_ID is used for representing the composition relationship as proposed mapping rule in Section 3.2. In the same way, we create a separate table named ROOM which holds the ID and NAME of the room object and a foreign key BUILDING_ID to refer to the building it belongs to. In that case, the sensor information (table MYAIR and ARCHIBUS) regarding to room can straightforward connect to ROOM by a foreign key reference. Figure 7 illustrates the logical data model for our relational database design.

4.3 Experimental Results

Eval-I: Analysis of distance relationship among buildings.

In this section, we analyze some positional relations among buildings in UNSW campus based on pure GIS data. To achieve this query, it is essential to know the following relevant functions in PostGIS:

- *float ST_3DMaxDistance(geometry g1, geometry g2);*
- *boolean ST_3DDFullyWithin(geometry g1, geometry g2, double precision distance);*

First, we are willing to search all buildings who are fully within 100 meters from “Red Center” building, the following query is conducted:

```
SELECT *
FROM building
WHERE building_name IN (
SELECT DISTINCT B.building_name
FROM building B, building R
WHERE R.building_name='Red Center - H13'
AND ST_3DMaxDistance(B.geom, R.geom) <= 100
AND B.building_name!=R.building_name
AND B.building_name IS NOT NULL
);
```

For simple comparison without specific distance value, *ST_3DDFullyWithin()* is more efficient due to fuzzy calculation for distance. As one of conditions in WHERE clause, *ST_3DDFullyWithin()* returns true if the 3D geometries of one building are fully within the 100 meters of “Red Center” building, so for those return buildings may not totally fall into the range, only some parts of polygons are returned. To show the complete building, we add a layer of query based on

¹<https://archibus.unsw.edu.au/>

²<https://citydata.be.unsw.edu.au/layers/geonode%3Amyair>

³<https://postgis.net/>

⁴<http://www.sfcgal.org/>

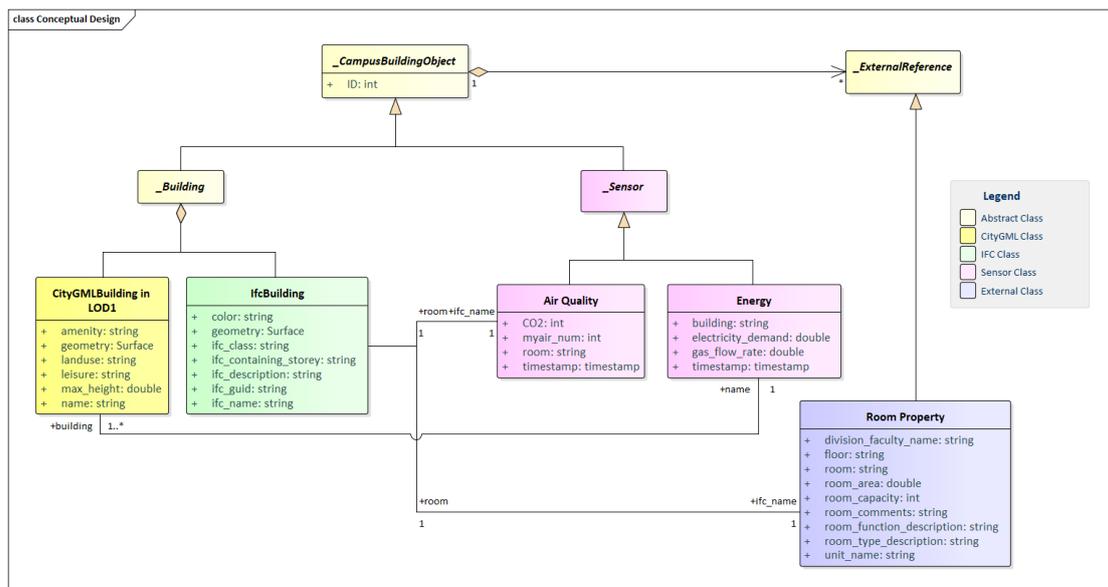


Figure 5. UML diagram for our conceptual design.

id	building_name	ifc_class	ifc_guid	ifc_name	ifc_description	ifc_containing_story	color	geom
1	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...
2	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...
3	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...
4	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...
5	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...
6	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...
7	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...
8	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...
9	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...
10	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...
11	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...
12	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...
13	Bloothouse -05	RBuildingElementProxy	2b4VE9t898mfr...	FIRE_STAIR_CONC...	Character varying (20)	GROUND FLOOR	rgb(153, 153, 153)	010F000A0A6E1...

Figure 6. Example of a table created in PostgreSQL to store a IFC building model.

building name returned by the sub-query. Our query result is visualized in Figure 8, where the caption of Figure 8 gives detail of visualization.

Eval-II: Analysis of internal objects in building. In this experiment, we analyze some positional relations among rooms in UNSW campus based on pure BIM data, in which we are going to search all rooms in “Red Center” building whose height is larger than the roof “Science Theatre”, the following query is conducted:

```
SELECT tmp1.room, tmp1.geom
FROM
(SELECT ifc_name AS room, ST_ZMin(geom) AS z, geom
FROM bim
WHERE building_name='Red Center - H13'
AND ifc_class='IfcSpace'
AND ifc_name NOT LIKE 'Lift%') AS tmp1,
(SELECT ST_ZMax(geom) AS roof
FROM bim
WHERE building_name='Science Theatre - F13'
AND ifc_class='IfcRoof'
ORDER BY roof DESC LIMIT 1) AS tmp2
WHERE tmp1.z > tmp2.roof;
```

To achieve this query, we first should acquire the roof height of “Science Theatre” by function *ST_ZMax* which returns maximum z value of a geometry. In BIM data table, attribute value “IfcSpace” represents room space and we pick the

minimum z value as height of one room. Our query result is visualized in Figure 9, where the caption of Figure 9 gives detail of visualization.

Eval-III: Analysis of semantic relationship between building and sensor data. In this part, we combine sensor data with BIM and GIS data to analyze energy consumption of each building or *CO₂* level of each room in UNSW campus. First, we are willing to calculate weekly electricity consumption statistics for the building which is nearest from “Science Theatre” (only consider the last month in 2018), the following query is conducted:

```
SELECT timestamp::DATE-(EXTRACT(dow FROM
timestamp::TIMESTAMP)||'day')::interval AS week,
SUM(electricity_demand) As elec
FROM greensense
WHERE building_name=(
SELECT B.building_name
FROM building B, building R
WHERE R.building_name='Science Theatre - F13'
AND B.building_name!=R.building_name
AND B.building_name IS NOT NULL
ORDER BY ST_3DMaxDistance(B.geom, R.geom) LIMIT 1
)
AND timestamp>='2018-12-01'
AND timestamp<'2019-01-01'
GROUP BY week
ORDER BY week;
```

To achieve this query, we first should acquire the name of the building closest to the “Science Theatre” by distance function. And then, divide electricity demand by week. Our query result is demonstrated in Table 2. Generally, consumption is much higher at the beginning of the month than at the end of the month due to the beginning is the final review while the end is Christmas holiday.

We also evaluate the performance of linking sensor data to BIM information. Supposing that, we care about *CO₂* level in “Red Center” building and are interested in the association between

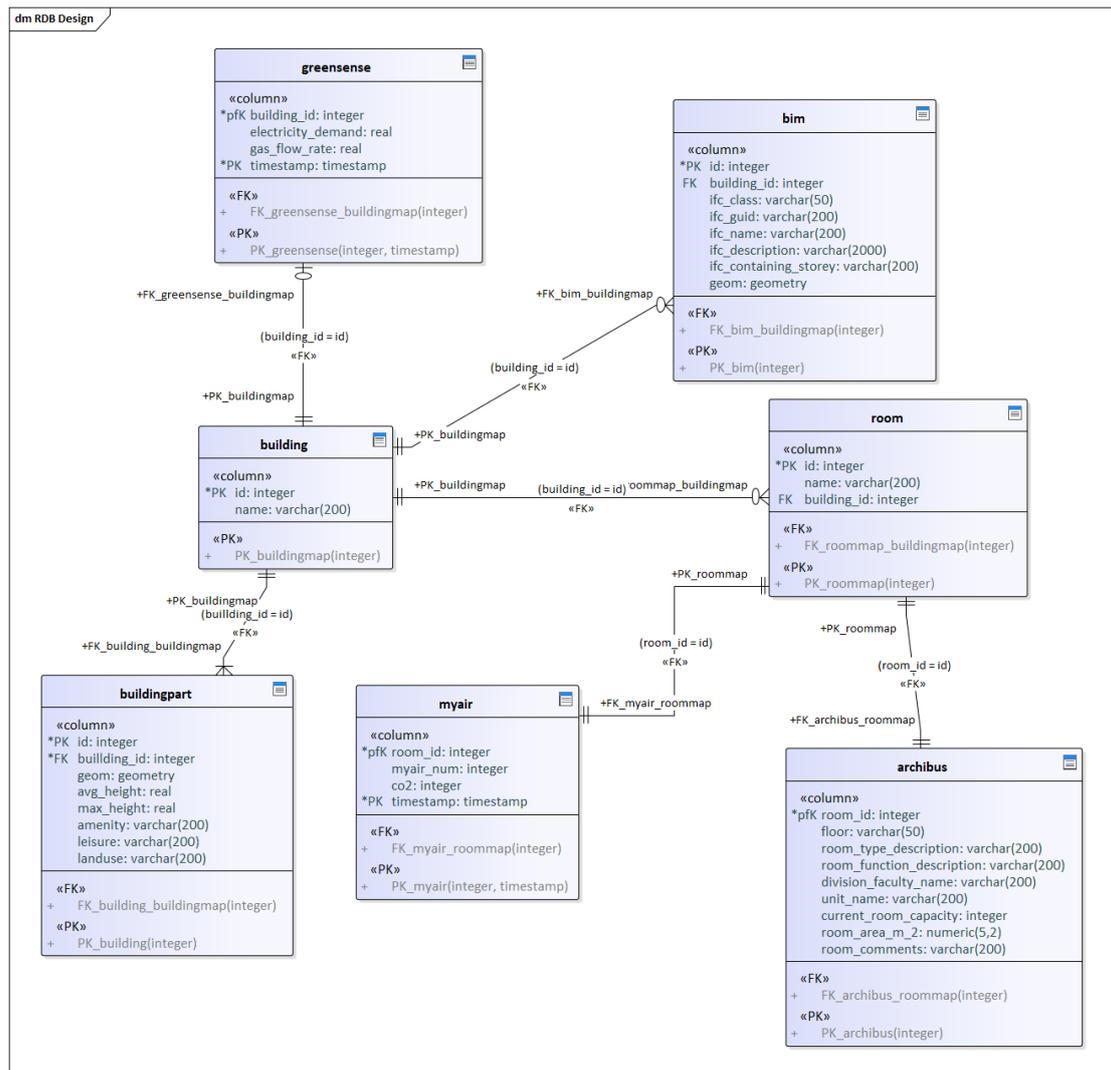


Figure 7. Logical data model for our relational database design.

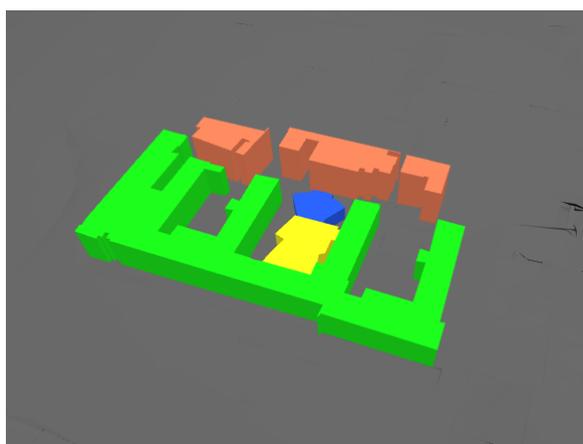


Figure 8. Visualisation of above query via QGIS. Object painted in red belong to the “Red Center” building (not in query result). Query results include “Keith Burrows Theatre” in blue, “Physics Theatre” in yellow and “Old Main” building in green.

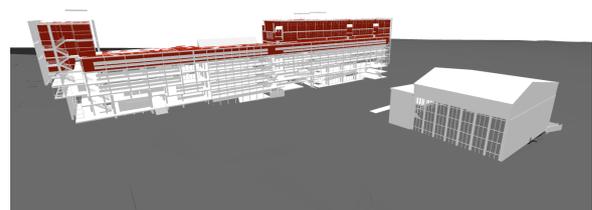


Figure 9. Visualisation of above query via QGIS. Query results include all rooms in storey 4 and 5 painted in red.

the volume/area of a teaching room and CO_2 level. To achieve this goal, it is essential to know the following relevant functions in PostGIS:

- *float ST_Volume(geometry geom1);*
- *geometry ST_MakeSolid(geometry geom1);*

The following query with respect to the top-3 rooms in “Red Center” building of CO_2 level per unit volume is conducted:

```
SELECT room_name AS room, co2, volume,
```

Table 2. Weekly electricity demand in December

Week(start week on Sunday)	Electricity(kW)
2018-11-25 00:00:00	2167
2018-12-02 00:00:00	23810
2018-12-09 00:00:00	24023
2018-12-16 00:00:00	18995
2018-12-23 00:00:00	12310
2018-12-30 00:00:00	4007

```
(co2::float / volume::float)::numeric AS fraction
FROM (
SELECT ifc_name,
ST_Volume(ST_MakeSolid(geom)) AS volume
FROM bim
WHERE building_name='Red Center - H13'
AND ifc_class='IfcSpace'
) AS tmp1,
(
SELECT room_name, round(avg(co2),2) AS co2 FROM (
SELECT to_char(timestamp, 'YYYY-MM-DD') AS d,
SUM(co2) AS co2, room_name
FROM myair
WHERE timestamp BETWEEN '2018-12-01'
AND '2019-01-01'
AND building_name='Red Center - H13'
AND co2 IS NOT NULL
GROUP BY room_name, d ORDER BY room_name
) AS tmp
WHERE room_name IN (
SELECT room_name FROM archibus
WHERE room_function_description LIKE '%TEACHING'
AND current_room_capacity>10)
GROUP BY room_name
) AS tmp2
WHERE tmp1.ifc_name=tmp2.room_name
ORDER BY fraction DESC LIMIT 3;
```

To achieve this query, we first determine which are teaching rooms based on statistic information from ARCHIBUS system. Then, calculate average CO_2 level of candidate rooms only considering the date when CO_2 was recorded in December 2018. At last, combine average CO_2 level with room volume calculated by ST_Volume function to compute the top-3 rooms in “Red Center” building of CO_2 level per unit volume. Our query result is demonstrated in Table 3.

Table 3. Top-3 teaching room in “Red Center” building

Room	CO_2	Volume	Fraction
3035	39952.07	1.79E-08	2.24E+12
2035	39310.47	1.95E-08	2.02E+12
3034	39189.42	1.95E-08	2.01E+12

5. CONCLUSIONS AND DISCUSSION

In this paper, a relational 3D geo-database solution for the management and analysis of unified building model with multi-source data fusion was presented. We proposed a high-level conceptual and physical data model of building domain. To improve the performance of our unified building model, we fused multi-source built environment data to enhance semantic power and developed efficient mapping rules to simplify relational database design. Case study



Figure 10. Visualisation of above query via QGIS. Query results include 3035 painted in red, 2035 in green and 3034 in blue.

shows that our unified building model can achieve better understanding of built environment and it will be a promising approach for future city or urban modelling work. Besides, it is an important contribution to The United Nations 2030 Agenda for Sustainable Development (2030 Agenda), which makes the world on a more sustainable economic, social and environmental path.

There are several possible directions that can be explored for future works. First, how to enrich built environment models for specific purposes or applications is interesting and promising. Second, integrating existing models (e.g., IFC model and CityGML model) is still a challenge work. For instance, two alternatives of implementation for integration can be discussed. In the first alternative the common geometric representations of all objects are identified and stored in separate tables. And thematic semantic tables would be linked to the geometric tables. In the second alternative the thematic semantic tables would integrate the geometries. Design and comparison of different integration implementations will be the next stage of our work.

ACKNOWLEDGEMENTS

Case study is supported by CRC for Low Carbon Living Ltd supported by the Cooperative Research Centres program (No.RP2011u1). The publication acknowledges the contribution of all authors, researchers, participants and others as appropriate.

REFERENCES

Agoub, Amgad, Kunde, Felix, Kada, MARTIN, 2016. Potential of graph databases in representing and enriching standardized Geodata. *Tagungsband der*, 36.

Bleiholder, Jens, Naumann, Felix, 2009. Data fusion. *ACM Computing Surveys (CSUR)*, 41, 1.

Breunig, Martin, Zlatanova, Sisi, 2011. 3D geo-database research: Retrospective and future directions. *Computers & Geosciences*, 37, 791–803.

Cheng, Jack CP, Deng, Yichuan, Anumba, Chimay, 2015. Mapping BIM schema and 3D GIS schema semi-automatically utilizing linguistic and text mining techniques. *Journal of Information Technology in Construction (ITcon)*, 20, 193–212.

de Laat, Ruben, Van Berlo, Leon, 2011. Integration of bim and gis: The development of the citygml geobim extension. *Advances in 3D geo-information sciences*, Springer, 211–225.

- Deng, Yichuan, Cheng, Jack CP, Anumba, Chimay, 2016. Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison. *Automation in Construction*, 67, 1–21.
- Dong, Jiang, Zhuang, Dafang, Huang, Yaohuan, Fu, Jingying, 2009. Advances in multi-sensor data fusion: Algorithms and applications. *Sensors*, 9, 7771–7784.
- El-Mekawy, Mohamed, Östman, Anders, Hijazi, Ihab, 2012. A unified building model for 3D urban GIS. *ISPRS International Journal of Geo-Information*, 1, 120–145.
- El-Mekawy, Mohamed, Östman, Anders, Shahzad, Khurram, 2011. Towards interoperating citygml and ifc building models: a unified model based approach. *Advances in 3D geo-information sciences*, Springer, 73–93.
- Fosu, Richelle, Suprabhas, Kamal, Rathore, Zenith, Cory, Clark, 2015. Integration of building information modeling (bim) and geographic information systems (gis)—a literature review and future needs. *Proceedings of the 32nd CIB W78 Conference, Eindhoven, The Netherlands*, 27–29.
- Ghamisi, Pedram, Rasti, Behnood, Yokoya, Naoto, Wang, Qunming, Hofle, Bernhard, Bruzzone, Lorenzo, Bovolo, Francesca, Chi, Mingmin, Anders, Katharina, Gloaguen, Richard et al., 2018. Multisource and Multitemporal Data Fusion in Remote Sensing. *arXiv preprint arXiv:1812.08287*.
- Gröger, G., Kolbe, T.H., Nagel, Claus, Häfele, K.H., 2012. OGC city geography markup language (CityGML) encoding standard.
- ISO, BS et al., 2013. Industry foundation classes (IFC) for data sharing in the construction and facility management industries.
- Liebich, Thomas, 2013. IFC4The 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.
- Ma, Zhiliang, Ren, Yuan, 2017. Integrated application of BIM and GIS: an overview. *Procedia Engineering*, 196, 1072–1079.
- Nakashima, Hideyuki, Aghajan, Hamid, Augusto, Juan Carlos, 2009. *Handbook of ambient intelligence and smart environments*. Springer Science & Business Media.
- Stadler, Alexandra, Nagel, Claus, König, Gerhard, Kolbe, Thomas H, 2009. Making interoperability persistent: A 3d geo database based on citygml. *3D Geo-information sciences*, Springer, 175–192.
- Zhang, Jianchao, Seet, Boon-Chong, Lie, Tek, 2015. Building information modelling for smart built environments. *Buildings*, 5, 100–115.
- Zhang, X, Arayici, Y, Wu, S, Abbott, C, Aouad, GF et al., 2009. Integrating BIM and GIS for large-scale facilities asset management: a critical review.
- Zhu, Junxiang, Wright, Graeme, Wang, Jun, Wang, Xiangyu, 2018. A critical review of the integration of geographic information system and building information modelling at the data level. *ISPRS International Journal of Geo-Information*, 7, 66.
- Zlatanova, Sisi, 2006. 3d geometries in spatial dbms. *Innovations in 3D geo information systems*, Springer, 1–14.

Revised Oct. 2019