



# A semantic graph database for the interoperability of 3D GIS data

Eva Savina Malinverni<sup>1</sup> · Berardo Naticchia<sup>1</sup> · Jose Luis Lerma Garcia<sup>2</sup> · Alban Gorreja<sup>1</sup> · Joaquin Lopez Uriarte<sup>3</sup> · Francesco Di Stefano<sup>1</sup>

Received: 9 January 2020 / Accepted: 12 August 2020  
© The Author(s) 2020

## Abstract

In the last decades, the use of information management systems in the building data processing led to radical changes to the methods of data production, documentation and archiving. In particular, the possibilities, given by these information systems, to visualize the 3D model and to formulate queries have placed the question of the information sharing in digital format. The integration of information systems represents an efficient solution for defining smart, sustainable and resilient projects, such as conservation and restoration processes, giving the possibilities to combine heterogeneous data. GIS provides a robust data storage system, a definition of topological and semantic relationships and spatial queries. 3D GIS makes possible the creation of three-dimensional model in a geospatial context. To promote the interoperability of GIS data, the present research aims first to analyse methods of conversion in CityGML and IndoorGML model, defining an ontological domain. This has led to the creation of a new enriched model, based on connections among the different elements of the urban model in GIS environment, and to the possibility to formulate queries based on these relations. The second step consists in collecting all data translated into a specific format that fill a graph database in a semantic web environment, while maintaining those relationships. The semantic web technology represents an efficient tool of interoperability that leaves open the possibility to import BIM data in the same graph database and to join both GIS and BIM models. The outcome will offer substantial benefits during the entire project life cycle. This methodology can also be applied to cultural heritage where the information management plays a key role.

**Keywords** 3D GIS · CityGML · Ontology · Graph database · Semantic web · Interoperability

## Introduction

Projects in the architecture, engineering and construction (AEC) sector are becoming increasingly large and complex, generating different types of information. The use of information technologies in the building data processing led to radical changes to the methods of production, management and archiving of data documentation. In particular, 3D model visualization and query

formulation are offered by many information systems at the building and terrain scale for data access (graphics, photos, technical documents, regulations, etc.) linked to the object.

Information systems are based on the digital representation with an informative 3D model linked to a relational database, which is an archive where all data of the object, in different formats, can be collected together under the *entity-relationship* model. Therefore, they represent useful tools for

---

✉ Eva Savina Malinverni  
e.s.malinverni@staff.univpm.it

Berardo Naticchia  
b.naticchia@univpm.it

Jose Luis Lerma Garcia  
jllerma@cgf.upv.es

Alban Gorreja  
a.gorreja@pm.univpm.it

Joaquin Lopez Uriarte  
lopezuriartej@gmail.com

Francesco Di Stefano  
f.distefano@pm.univpm.it

<sup>1</sup> Department of Construction, Civil Engineering and Architecture, Università Politecnica delle Marche, Via Brecce Bianche 12, 60131 Ancona, Italy

<sup>2</sup> Escuela Técnica Superior de Ingeniería Geodésica, Cartográfica y Topográfica, Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Spain

<sup>3</sup> School of Civil Engineering, Universidad de Cantabria, Av. de los Castros, 44, 39005 Santander, Cantabria, Spain

project management. In recent years, the research in AEC sector did not focus on the use of a single information modelling for the data management, but it tried to make an integration of different information systems to create a multi-disciplinary, multi-scale, multi-scope, multi-user approach. In order to achieve a model technically richer and more complete of the object, it is necessary to share data among the information systems used and so, where it needs, to proceed to the data conversion or transformation in a digital exchange formats to promote the concept of interoperability. The development of this more advanced environment, supported by the interconnection, the communication, the data transferring and enrichment among the information systems, requires the creation of a common platform. For example, using the web network, where all data from different sources are collected and connected in a unique graph.

In AEC field of work, among the information systems and technologies, GIS (Geographic Information System) is a useful tool for building and terrain data management. The integration of GIS with data coming from other information modelling environment represents an efficient way for defining a smarter, more sustainable and resilient project. It gives the possibilities to combine heterogeneous data: geometric shapes, quantitative analysis, enrichment of semantic knowledge, application of different technologies and multi-scale management (Ma and Ren 2017; Fosu et al. 2015; Yamamura et al. 2016). This innovative integration between information systems becomes suitable to many building fields of application. Some representative examples are the design and construction stages, the management of the construction sites, the site layout planning and the location of temporary objects (Sebt et al. 2008). GIS integration with other technologies permits to define, follow and control each step of the building or infrastructure project and its effects on the territory or the elements connected to such as installations and services. Moreover, GIS provides to be an additional support system for the historical building or sites management, offering a new way of co-working for the preservation, conservation, monitoring and restoration activities of cultural heritage. GIS data can be linked with high detailed information models, like those created through Building Information Modelling (BIM) software. For this reason, the integration of GIS and Heritage (or Historical) Building Information Modelling (HBIM) (Murphy et al. 2009) can be considered as an adequate solution for managing the information for conservation and restoration projects of existing building at architectural scale (Malinverni et al. 2018; Malinverni et al. 2019; Matrone et al. 2019).

Although GIS was originally used to manage geospatial data in 2D scale, it provides a robust data storage system

(Vacca et al. 2018) based on a sort of hierarchy of classes and subclasses identified by the levels of detail of CityGML schema. The latter permits the definition of topological and semantic relationships between the objects. By the development of 3D GIS it makes possible the creation of three-dimensional geospatial modelling, allowing the data management of specific building, offering a precise visualization of the geographical contextualisation and permitting the formulation of spatial queries (Rinaudo et al. 2007). So, the 3D building model can be raised on a modelled terrain in a relative urban context with its surroundings. While CityGML describes and classifies object in the urban scale, IndoorGML focuses on the connections between the rooms of a building helping to create navigable routes, creating a network. In this research we used IndoorGML to make connection between buildings and outdoor elements such as roads.

After this preliminary overview, it is clear that the attention focuses on the in-depth analysis about the management of GIS data. In particular, this paper illustrates a research topic based on the conversion of 3D GIS data into an exchange format that can be readable and interpretable by other information systems, such as BIM or HBIM. In this research we tried to find out a possible solution to convert 3D GIS data, giving those data a CityGML ontology and, after a process of elaboration, importing the transformed data in a semantic web database based on RDF (Resource Description Framework) graph. The final result of the work is to identify the benefit in using the semantic web platform to collect GIS data, translated in a unique common exchange format, based on RDF connecting triples (Hor et al. 2018). Then, in the same web environment, it is possible to add and gather other converted data coming from other information systems. This semantic interoperability needs to be provided by a domain for the description and the individuation of the objects (Karan & Irizarry 2015). In computer science, ontologies are adopted as domains providing a formal structure for sharing and managing data defining objects (taxonomies) and their relationships (El-Diraby et al. 2005).

Keeping a standardized ontology can play a key role in a project development (such as reconstruction or building restoration projects). It helps the designers to have a quick overview of infrastructure affected by any work decisions. Furthermore, they allow linking information between many models created under different standards (GIS, BIM, HBIM) to make more complete decisions.

This paper is structured as follows: the first part is dedicated to the state-of-the-art of the concept of interoperability between information systems (“*State-of-the-art*”); then after a small bracket about the semantic web, the methodology workflow is described (“*Methodology*”), defining so the standards and the tools, used for 3D GIS data conversion, analysing their characteristics, properties and functions (“*Standardization*”). Hence, the process of translation from CityGML to RDF-graph will be outlined (“*Implementation*”).

of graph database”), and finally, some considerations and results will be discussed (“Conclusions”).

## State-of-the-art

The data integration process, as said before, between information systems represents an innovative approach offering substantial benefits, and this combination has to take into account the strong points from each system (Zhang et al. 2009). Considering for example the most used systems like BIM and GIS, their integration represents an efficient tool for AEC projects. In short, BIM describes geometry, semantic relationships and identifies the building components. GIS provides a well-structured database (level of detail) and a geospatial model with topological and semantic relationships. But there are some dissimilarities between them, such as spatial scale, level of representation of geometric models and structure of database.

By literature, the topic of translation of data between GIS and other information systems, BIM in particular, has been already dealt with, as you can read in some examples below, and it is still an ongoing research, but it led to a specific solution, not simple to understand or to propose again for different case studies. Especially, it is to be highlighted that some software producers started a collaboration trying to get easier and to overcome the complexity of the data conversion and sharing between information systems, so it is already possible to import BIM data into a GIS environment. Although this has shown good performance, it must be said that this data conversion has a limitation because it only proceeds in unidirectional way, from BIM to GIS. This is not the aim of our research work.

Analysing research activities focusing on the topic of integrated information system, the interoperability between BIM and GIS can be presented in different ways: syntactic interoperability and semantic interoperability (Bishr 1998).

The syntactic interoperability refers to use common data format to exchange information between BIM and GIS systems, using a domain of one of the information systems. Examples of syntactic interoperability are systems that combine building data with landscape maps, data formats of BIM object on GIS environment, and solutions to convert BIM data from IFC (Industry Foundation Class) to CityGML (Karimi and Akinci 2009). The approach of IFC to GIS (IFG) project has been developed to provide geographic information between the frameworks of IFC, in order to get a more efficient planning (Kolbe et al. 2005). Other examples of syntactic way are based on the use of CityGML schema. The GeoBIM extension from BIMServer allows conversion from IFC files to CityGML files by defining additional information in CityGML entities (Van Berlo and De Laat 2011). Nagel et al. (2009) proposed a conceptual method of transitioning

from a KML graphics model to BIM through CityGML. The authors developed mapping rules to allow transforming CityGML model to IFC model. Another attempt, tried by Hagedorn et al. (2009), was to create a conceptual dual graph for representing topological relationships among indoor entities of a building, but not the whole one with the geospatial context. All these examples chosen to explain this type of interoperability show a common characteristic: they follow a unidirectional way of translation and do not consider the semantic information mapping in the process. A bidirectional approach is needed (Deng et al. 2016), if we want to get a dual interoperability between BIM and GIS.

The highest level of interoperability is guaranteed by the semantic aspect of data integration. The key point of semantic interoperability is to make sure that features and relations between information management systems are maintained during data conversion (Peachavanish et al. 2006). Objects as entities and their relationships are defined under a domain called ontology that makes possible the representation, the sharing and the management of the knowledge (El-Diraby et al. 2005). The information should be described and classified in a standard way. The semantic web technology represents new efficient online platform to make possible this kind of interoperability. The web ontology language (OWL) expresses the data in terms of classes. A collection of these classes, their attributes and relations can be stored as RDF triples describing each individual object, its properties and features, which can be understood as a graph based on nodes (entities) and vertices (relationships) (Hor et al. 2016). Semantic web technologies have been used by several researchers to facilitate construction project information sharing. Anumba et al. (2008) explored the use of semantic web technologies to meet the challenges of collaborative project information management. Akinci et al. (2008) developed a web-based approach to enable semantic interoperability between CAD and GIS platforms. Beetz (2009) demonstrated the feasibility of semantic web tool to address information exchange and integration problems in AEC interoperability.

Taking this approach into account, it becomes possible to move forward a graph database direction, useful also for a BIM data interaction providing so a complete interoperability in the web environment and so allowing an eventually bidirectional way of data transfer between information systems.

## Semantic web

As we previously discussed the main obstacle of the integration between different information systems is the lack of interoperability across both domains, which could be solved using the semantic web. The semantic web is a set of technologies used for the representation, publication and browsing of structural data on online platforms. It is used in this study to convey meaning, which is interpretable by construction

project stakeholders as well as BIM and GIS applications processing the transferred data (Ebrahim and Irizarry 2015).

The main elements belonging to a semantic web platform (Hor 2015) are as follows:

- Uniform Resource Identifiers (URIs), a string of characters that identifies a particular resource;
- Web Ontology Language (OWL), a type of knowledge representation languages form authoring ontologies for representing the conceptual schema;
- Resource Description Framework (RDF), a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata model for defining the data according to the schema;
- SPARQL, a SQL-type language for carrying out queries in data.

RDF, RDFS (Resource Description Framework Schema) and OWL are languages with clearly defined semantics or mathematical basis for the meaning of each construct. Since concepts in RDFS and OWL ontologies are expressed formally, they can be processed by computer programs.

The ontology is the key element of the semantic web. It classifies objects, data, etc. (entities), with their attributes and the relationships among them inside a domain of knowledge. In other words, it can be defined as a data structure that represents a model of semantic representation of reality. The shared language used to describe the semantics of the data is the uniform way to facilitate the communication among different users to understand each other (Mohammad 2010). Ontologies are used to overcome the barriers to heterogeneous semantic data sharing. They are commonly used for many purposes such as network management, data exchange on the World Wide Web and information retrieval (Hor 2015). There are several examples in literature where ontology is required before the data conversion (El-Diraby et al. 2005; Hor et al. 2016). Also, it remarks the importance to maintain the ontology of a model in order not to lose the meaning of each feature during the data format conversion.

As is described in Kolbe and Nagel (2012), main elements of RDF are triples composed by three elements: subject, predicate and object. They can be represented in a graph (upper of Fig. 1) with three linked nodes. The meaning of predicate

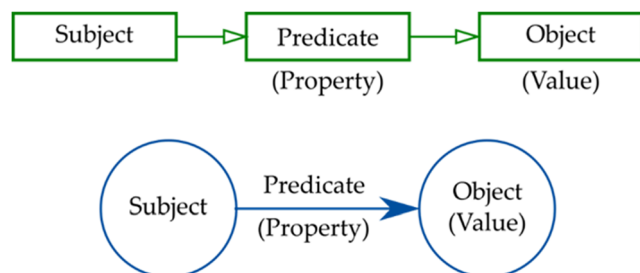


Fig. 1 RDF triples

could be interpreted as property and the meaning of object as value. The final interpretation of the triple will depend on the data stored, e.g., if the predicate is “hasChild”, the meaning of the object could be a value if a boolean data (yes or no) is stored, or another object if all data regarding the child is linked. Other manner to create this RDF graph is using two nodes (subject and object) connected by an edge (predicate) (bottom of Fig. 1). This paper uses the latter structure to maintain CityGML ontology and to keep semantic web techniques.

Semantic interoperability is crucial for its ability to exchange information between existing standards. It is also capable to automatically interpret the exchanged information in meaningful and accurate way. This process allows users to combine 3D models and get information from them. To do so, these models must converge to a common information domain. In this way, the exchanged information using semantic interoperability is unambiguously defined. This means that the sent data are the same as what is understood (Wiki.GIS 2019).

## Methodology

The improvement of GIS model, by creating connections and data conversion towards another domain, such as BIM, has been already discussed. The aim of this paper is to transfer the data from a GIS model to a common web database where all its features remain interconnected. Other models, as BIM one, can be also imported in this web database and linked to the GIS model. The hypothesis is that every model is created using open standardized ontologies and keeping them is the main purpose. The resulting database defines the connections between models in the same information system, where a feature can influence features of other models. Every feature in the model is connected to other thanks to the ontological relationships.

Taking the urban context as example, objects composing the city model can be created and collected in different information systems where, basing on own ontological schema, specific features are managed. A unified model, which gather all the information coming from several information systems, will be built in a semantic web environment where all the figures involved to the project can interact with it. Among the urban services, this approach can be used for analysing the road system, in particular the management of the emergency, security and traffic when an obstruction may compromise the passage. For example, works on a façade building near the adjacent road may imply a traffic diversion. Furthermore, the unified model will be useful to create spatial connections between elements of the urban area. In this work it was considered a simple case, as a demonstration, that recognizes buildings rising in sequence along a street, testing the operation of building-road connection.

In order to test the connection building-road, we have chosen a case of study concerning an urban area of the city of Bologna, characterized by the presence of porticos which need interventions. The methodology is centred on GIS model and its transformation to the common database. The workflow is structured on steps that are outlined as follow. First, it is necessary to prepare the GIS model based on the available data. Shapefiles (.shp) of roads and buildings have been downloaded free from the OpenData, an open source platform provided by the Municipality of Bologna. These objects allowed to realize a 3D GIS model, where first we made the Triangulated Irregular Network (TIN) of the terrain. Then we built the 3D shapes of the buildings and defined the road map, laying all them on the TIN. The second step consists in importing shapefiles and other data into FME (Feature Manipulation Engine) software to create CityGML and IndoorGML models. FME is an ETL (Extract, Transform and Load) tool, which automatically translate the information from one format to another one (FME 2019). Finally, using an algorithm developed in Python, to transform CityGML and IndoorGML models into JSON data, we collected them into a graph database on web environment. We have chosen the solution of a NoSQL database, because this kind of database, thanks to its flexibility, scalability, and high performance, is able to collect a wide variety of heterogeneous data in modern applications. Next paragraphs explain deeply the methodology adopted (Fig. 2).

## Standardization

### 3D GIS

The information, provided by cartographic services, is expressed in bidimensional way where sometimes elevation values are specified. In order to create a 3D GIS model, we need to make modifications in the original 2D data. As the case study represents an urban area, a 3D city model has been created from Digital Terrain Model (DTM) and 2D shapefiles of buildings and roads has been imported. The building shapefiles consist in polygon features including height data, whereas roads are drawn by poliline features. The latter have planar coordinates ( $x, y$ ) but do not have elevation values ( $z$ ). As previously indicated, the aim of this research is to create

spatial connection between GIS object of buildings and roads, through a representation based on nodes. In this case, the connection point of a building with a road is assumed to be the nearest point of the road (road nodes) to centre of gravity of the building (building nodes) calculated within ArcGIS (ArcGIS 2018). The roads are splitted at these connection points obtaining a logical network (bottom left of Fig. 3). Furthermore, for an easy and fast recognition, an univocal identifier (ID code) is assigned to the resulting roads, to the road nodes and to the buildings. All these data, ID codes and number of nodes of the roads, are stored in a specific datatable saved in CSV format. Therefore, to fill the building datatable, a new column will be added to contain the ID code of the corresponding road node (bottom left of Fig. 3). In the same way, the ID codes of the starting and ending nodes for the corresponding segment are selected to the resulting road shapefile. The data preprocessing and the code identification help the following creation of the IndoorGML model, built after defined the CityGML schema. However, buildings and roads are still 2D features. In passing to the 3D level, first, the  $z$  coordinate (altitude) of the base feature needs to be retrieved from the TIN (Triangular Irregular Networks) of the terrain, which has been obtained from contour maps with accuracy of 2 m. Thus, knowing the height of the buildings, they become 3D polygons. In this way, 3D polygons are represented by solids built extruding the baseline polygon to their height (obtainable from the building datatable) and using multipatch features (upper right of Fig. 3). At the same time, lines of the road map follow the three-dimensional orography of the TIN model.

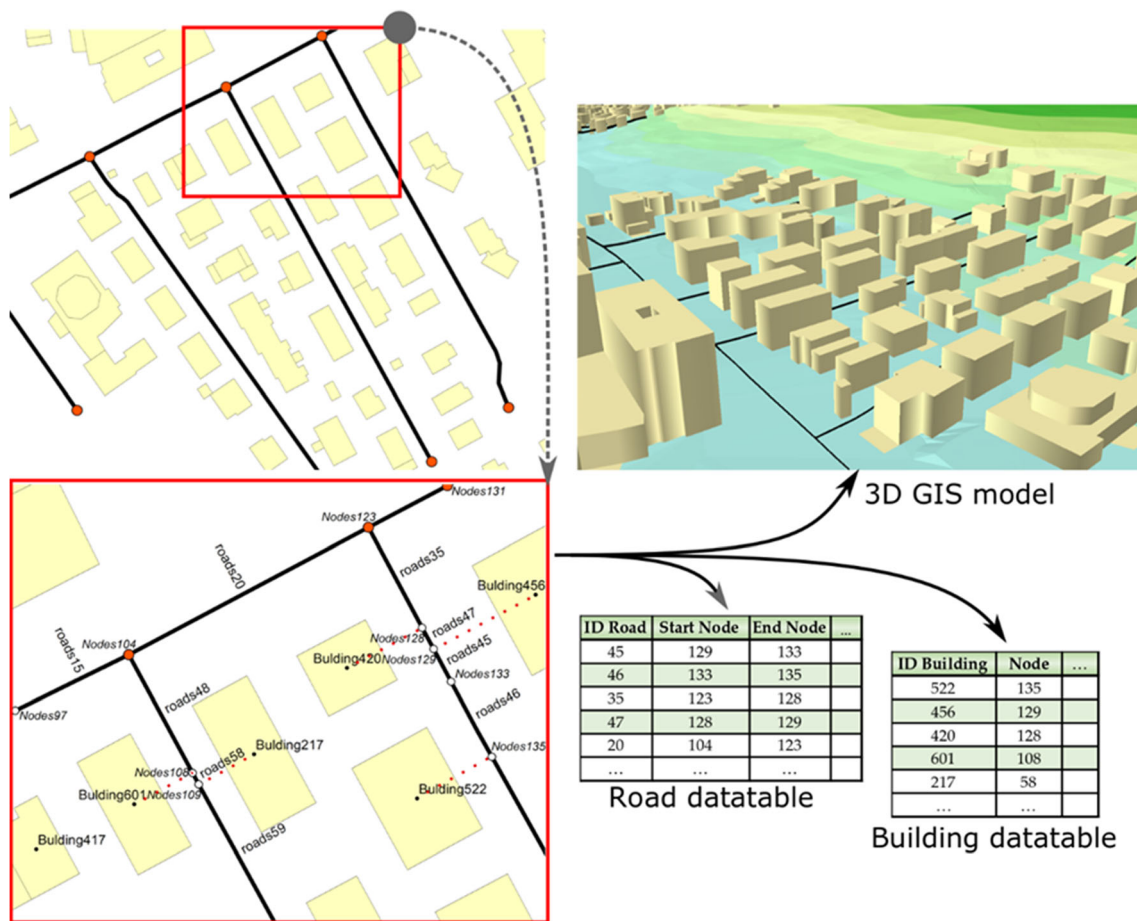
### CityGML

Nowadays there is a variety of international standards for each field of application. Open Geospatial Consortium (OGC 2019) is an international organization which develops and maintains open standards (more than 60 have been published). Among all them available, the most extended and used in city modelling is CityGML, issued also by the ISO/TC 211 regulation (ISO/TC 211 2019).

CityGML is a common semantic information model for the representation of 3D urban objects that can be shared over different applications. It is an open source data model and eXtensible Markup Language (XML)-based format for the



Fig. 2 Methodology workflow



**Fig. 3** Operations carried out from the original data, at upper left. A zoom view, in bottom left, shows ID roads nodes, ID buildings and ID roads (linked by a dashed line). The assigned ID codes collected in datatables, in down right. Upper right: the final 3D GIS model

creation and the exchange of city models. The geometry is stored using the Geography Markup Language version 3.1.1 (GML3), which is also XML-based format and is usually employed in geographical information archive.

Furthermore, CityGML enables lossless information exchange between GIS software and users. It defines classes and relations regarding their geometrical, topological, semantical and appearance properties (Kolbe and Nagel 2012). It is applicable for large areas and small regions and can represent the terrain and 3D objects in different levels of detail (LOD) simultaneously. For example, simple models without topology and few semantics in one LOD, instead of very complex detailed models with topology and fully semantical, can be represented in different LODs.

LODs in CityGML indicate the accuracy of geometries and the potential elements that are included in the model. LODs range from 0 to 4. Let us use building objects for illustrating them: LOD0 is the coarsest model, and it is mainly a 2D model with a DTM; e.g., building would be represented as a 2D polygon laying on the DTM. LOD1 includes buildings with its height. LOD2 defines the structure of roofs and building installations. LOD3 represents the real geometry with

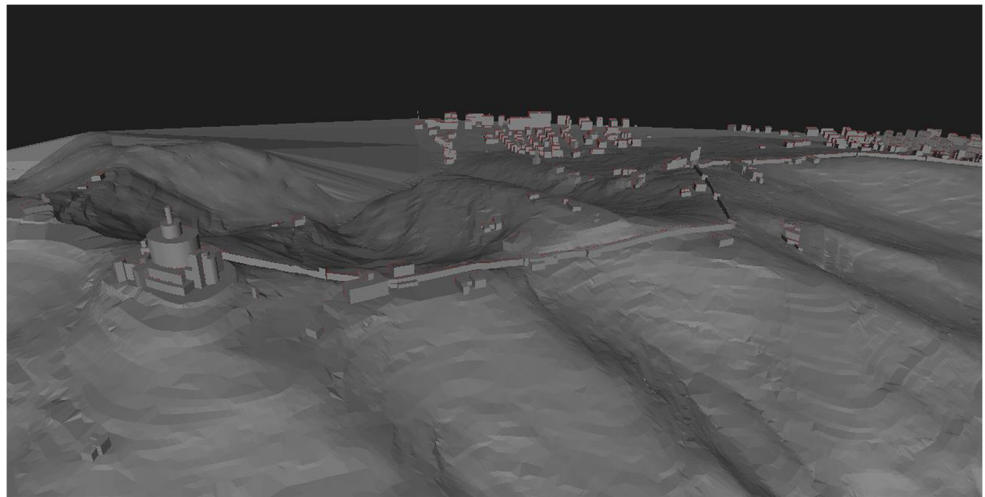
accuracy, and LOD4 is the realist model in which all details of the building are modelled. CityGML specification (Kolbe and Nagel 2012) defines for each object model the information needed for each LOD.

As has been previously indicated, CityGML is chosen because it is widely used in GIS city modelling and represents an open recognized standard and well-defined ontology.

In this project, the CityGML model was constructed based on the 3D GIS model using FME software. 3D building and road shapefiles are imported into FME, and they are transformed to LOD1 Building and LOD0 Road objects in the CityGML model. The TIN is also transformed to a 3D relief and incorporated to CityGML model. FME has a set of tools that carries out this conversion with only specifying its input data. In this way, this ETL allows the designer to easily create the CityGML model. A previsualization of CityGML geometric result from FME interface is shown in Fig. 4.

The main problem of this data is the way in which semantic connection between these independent elements is created. One way to create semantic relationships among objects in CityGML is through geometry, but it would require a full topologically consistent model with high LOD. This is rarely

**Fig. 4** CityGML viewed in FME Data Inspector



achievable. For example, many building and road shapefiles available do not share any point between them. Consequently, topological connections cannot be carried out and both kind of objects remain independent. Other problem is that FME does not automatically obtain geometrical relations between buildings: if two buildings share a wall, this geometric element is written in both buildings CityGML model. For these reasons, other ontological domain is needed to fill this gap and IndoorGML semantics adjusts to it.

### IndoorGML

IndoorGML (IndoorGML 2019) is a OGC standard, XML-based, and it is implemented for defining routes inside buildings, linking different rooms through doors and so, navigable spaces. Among all OGC standards, it is the only one that establishes an ontology for navigable routes and has a possible connection with CityGML. In this research, a slightly different interpretation of IndoorGML has been developed to make connections between CityGML elements. In this case, IndoorGML acts as linking network between outdoor and indoor spaces. An improvement of its ontology could deal with any real space because the paths can be used for several case studies, such as wifi connectivity (IndoorGML 2019). The proposed general model could have a central navigation module with some predefined extensions in indoor and outdoor environments, but this development is not the scope of our research. The aim of this work is to use IndoorGML ontology with others existing standardized to create connections between buildings, identified by a point in their centre of gravity, and the nearest point belonging to road lines (nodes).

CityGML and IndoorGML have some similar features, in particular: geometry, existing in both models, and CellSpace object in IndoorGML which corresponds with Buildings object in CityGML. In Fig. 5 is shown a schematic diagram of

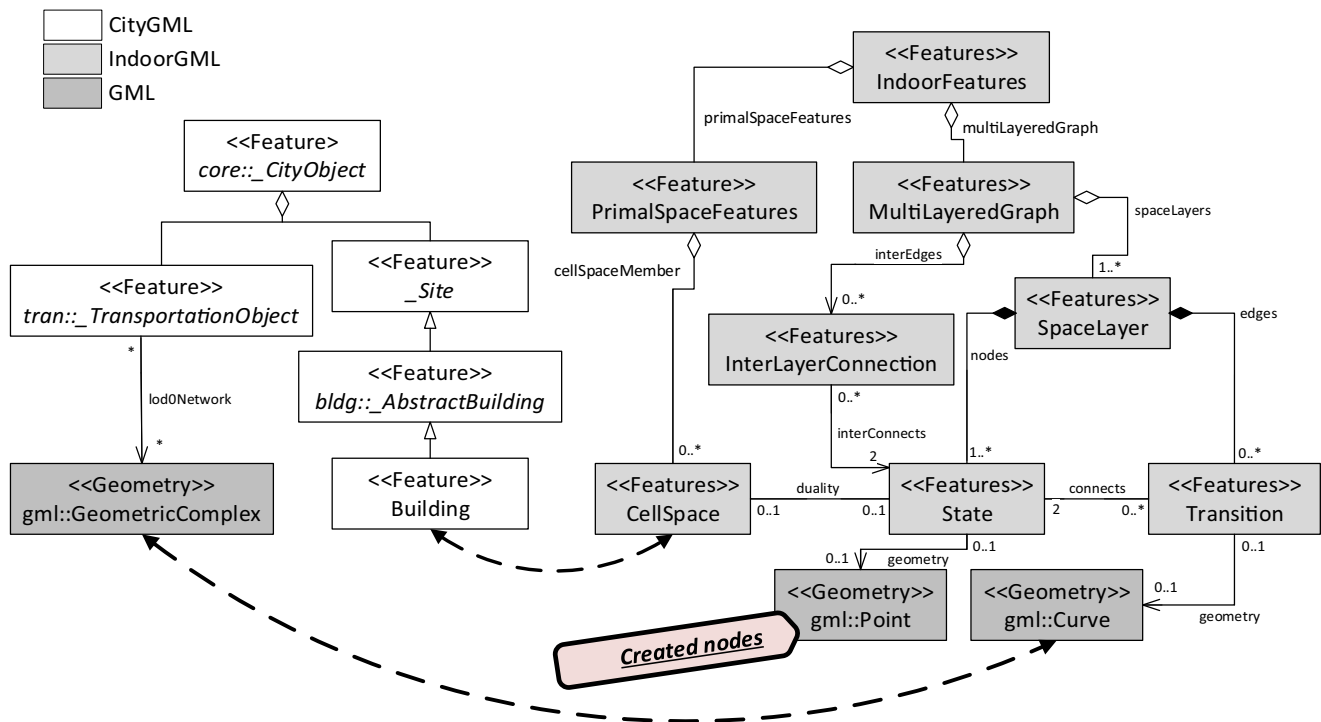
CityGML and IndoorGML considered and the proposed connections in geometry and CellSpace.

An useful software to easily create an IndoorGML has not been found. FME specifications indicate that this XML format has already been incorporated, but there is not a direct processing to generate IndoorGML data. Therefore, a new workflow was created using FME software to write an IndoorGML file, although any other software (or even a script) could be used. This workflow takes as input data building–road connections as specified in “3D GIS”. in CSV format (created from Roads datatable and Building datatable represented in Fig. 3).

The following explanation illustrates the process of how the UML (Unified Modelling Language) diagrams of CityGML and IndoorGML were interconnected using FME while creating connections between the two shapefiles, output of the 3D GIS data, previously created.

To elaborate the proposed procedure some considerations have to be taken into account. Every building is considered as a unique SpaceLayer element of IndoorGML ontology because they are unrelated. Inside this layer, all buildings are defined as CellSpace and assigned its corresponding ID as is possible through GML specifications. On the other hand, roads form another SpaceLayer element and are defined by their geometry (line entities for Transition elements and point entities for State ones). Connections among elements from different SpaceLayer are made with InterLayerConnection feature.

Furthermore, Transition feature can connect two States features using nodes created in “3D GIS”. It is therefore possible to use for roads connections between stretches of road or buildings and roads. The connection of nodes with road segments and buildings creates a network between elements. It achieves the objective of having one interconnected model where you can navigate from one feature to others. The connection between IndoorGML CellSpace and CityGML



**Fig. 5** Simplified UML diagrams of CityGML and IndoorGML. The dashed line indicates the connection between both structures and the label “Created nodes” references to the nodes created in section 4.1

Buildings and geometry connections are carried out through its ID, which has to be the same in each element of both models to connect them.

The importance of such process will be better understood in the next paragraphs, where the use of a graph database will better explain how these connections can be used to improve an information model through queries and enriching the data. As a result we obtain the model in GML containing all necessary links for the graph database.

### Implementation of graph database

The use of open source standard facilitates end users to understand every model. But these standards are usually disconnected among them as is the case of CityGML, IndoorGML or IFC. As previously expressed, some solutions exist and the approach used here is useful to create a common system where an entire model can be incorporated. This common system is represented by a graph database with RDF triples.

Graph databases belong to NoSQL databases. These types of database are useful when they have to store unstructured information. Furthermore, they allow to carry out fast transversal queries. Nowadays, there are a variety of software in this field such as ArangoDB (ArangoDB 2019; Fernandes and Bernardino 2018), MongoDB (MongoDB 2019; Fernandes and Bernardino 2018), or Neo4j (Neo4j 2019; Fernandes and Bernardino 2018; Hor et al. 2018). We chose to use

ArangoDB for its speed carrying out traversal queries, but any other could be used.

These databases have multi-model structure because they can collect information in different ways. This methodology is based on the use of documents and graphs (ArangoDB 2019). Documents store information, whereas graphs define relationships between data. Both are used by ArangoDB.

ArangoDB uses JSON (JavaScript Object Notation) format to store information in documents. Each one of them can contain different type and quantity of attributes. Documents are stored into collections assigning them, automatically or manually, a univocal key value. There are two types of collections: vertices and edges. The main difference between them is that edge collection has two special attributes that vertex collection does not have: *\_from* and *\_to*. These two attributes are used to create relations among documents of any vertex collection stored in the database.

The diagram of Fig. 1 shows that these graph databases can be defined graphically, with vertices and edges, where vertices are documents and edges are relations. It allows to use RDF graphs triples to define the data in ArangoDB where Predicates (from RDF) represent edges, whereas Objects and Subjects are vertices (or documents).

### From GML to JSON

Both CityGML and IndoorGML are GML formats, but currently, ArangoDB only imports JSON or CSV formats.



Consequently, GML models must be converted to one of these formats. The simplest way to create objects is to transform in JSON rather than in CSV.

Indeed, there is an encoding for the OGC CityGML data model called CityJSON (CityJSON 2019). CityJSON mainly describes the geometry, attributes and semantics of different kinds of 3D city objects. While we need to manage only the geometry of the objects and their geographical position to define the spatial relationships, we opted to use the GeoJSON format (GeoJSON 2019; Butler et al. 2016), mostly used for the road map management (Ferdinandus and Setiawan 2016). GeoJSON supports geometry types as point, polygon and multipoint. All things, which are supported by JSON, are also supported by GeoJSON. The difference between JSON and GeoJSON is that the key naming of each array element in GeoJSON has to follow certain guidelines. It is because the structure of GeoJSON follows the international standard published by OGC. GeoJSON has a specific function to support geographical data with a standardized format. It defines several types of JSON objects and the manner in which they are combined to represent data about geographic features, their properties and their spatial extents. GeoJSON uses a geographic coordinate reference system, World Geodetic System 1984 (WGS 84). As just explained, GeoJSON is a JSON format to store and exchange spatial data so can be imported in ArangoDB.

As an RDF graph database is used, it requires to define both nodes (subject and object) and the linking edges (predicates). First of all, GML format is constituted by elements between one start-tag and one end-tag (Fig. 6). Tags represent the kind of element that is being saved between them: for example they are the ontological names of the element. An element could be empty, have information concerning the element or have other nested elements (called child elements). An attribute is a name–value pair placed after the start-tag.

In order to create ArangoDB–RDF graph from GML formats (CityGML and IndoorGML) the main hypotheses are listed below:

- An empty initial node is created to which all main elements are linked. It helps to have a connection between all main elements of the model.
- Each tag is translated as an edge that links two nodes.
- Nodes store the information concerning the element or its attributes. If it only has nested elements, it will be an empty node.
- The attributes of a tag are saved in the document or node that acts as object (child node).
- If the attribute “gml:id” exists, it is used to define the document identifiers through the ArangoDB keyword: “\_key.” Every document in a collection has a unique identifier.

- If the attribute “xlink:href” exists, it is translated as an edge that links the parent element to the object at which it refers.
- The geometry is translated to GeoJSON format and saved in the child node.

In order to import data in ArangoDB, data must be translated to JSON format. While any useful software has not been found to carry out this transformation, a script in Python (Python 2018) has been developed. It uses XML library for reading the file and OGR library (GDAL 2019) for translating GML geometry to GeoJSON. Its outputs are two collections in JSON format for each model transformed: the edge collection and the document (vertex) collection. In this way, ArangoDB can read them as individual documents and assign them a unique identifier (if it is not defined with “gml:id” attribute). The script is based on schema less conversion. For example, it only reads the data model without checking the schema structure. However, it can become tedious and prone error when GML has many abstract type elements contained in the schema. XML schema defines that abstract types cannot be called from models or other schemas. So, they will not be defined neither in the data model nor in the graph database, while they will be shown in UML diagram. For instance, “\_AbstractBuilding” does not appear in the graph diagram in Fig. 7.

As previously said, the initial node is empty and refers to all CityGML model. In this example, only one building defined by one geometric and one generic attribute composes the model (Fig. 7, right). The predicates, or edges, are the names of the schema objects that are not GML abstract elements. Thus, following this representation, firstly appears the predicate “Buildings,” which defines the element type. The object of the first triple contains the building ID (i.e., the vertex after the edge with predicate “Building”). The geometry, inside the building element, is determined by other triple. The UML diagram (Fig. 7, left) shows different ways to define the geometry. In this case, there is a “lod1MultiSurface” element that can be identified in the graph. It composes the final triple of this example, where the subject is the node of the building, the predicate is “lod1MultiSurface” and the object will be the geometry itself.

Figure 8 shows the base schema of process used by the Python script. It takes into account the hypothesis shown previously and creates the JSON file following this procedure:

1. Read GML model as if it was any XML document and create the initial empty node to which all the main model elements will be linked.
2. The elements from the file are read and stored in memory. Most of them are processed following the same method, but there are some special cases where they require a different treatment. Mainly, these are included in IndoorGML and are listed below:

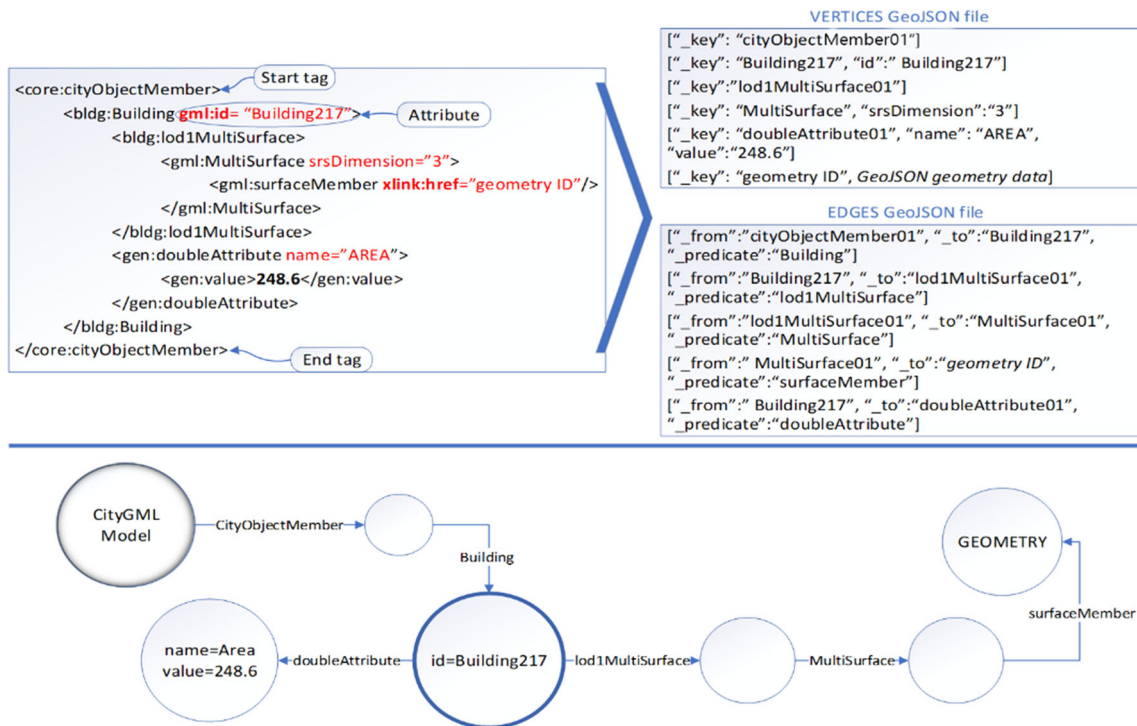


Fig. 6 One CityGML Building object with a generic attribute and its transformation to GeoJSON file. Above the definition of vertices and edges file, down the graphic representation of the relations

- *Cellspace* element. It is part of IndoorGML model, and it is referred to a CityGML building through its ID.
- *Duality* element. It is part of IndoorGML and it creates a connection between a Cellspace and a State from a linking element using its ID. In this case, it is a vertex from the roads.
- *Attributes* elements. They are part of CityGML generic attributes and allow creating special attributes. They are saved in a document, linked to the parent element using

the tag-name and saving the data in the document using the property name “value.” If tag-name attribute is defined, it is also stored.

If there are not any special cases, the elements pass the same process. Tag attributes (if they exist) are saved in the document that has been previously created. Then, the script checks the existence of nested elements. If the next nested

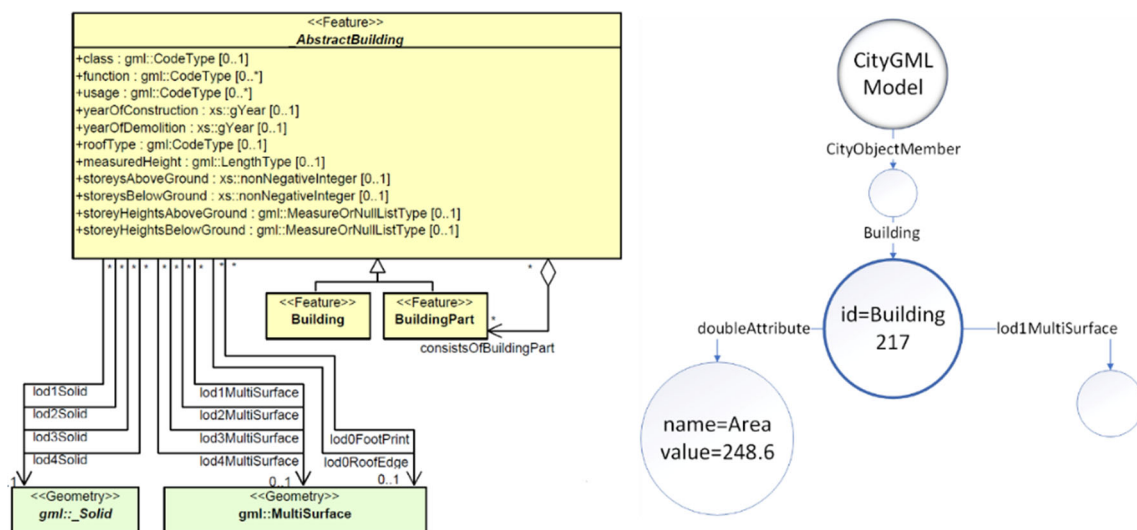
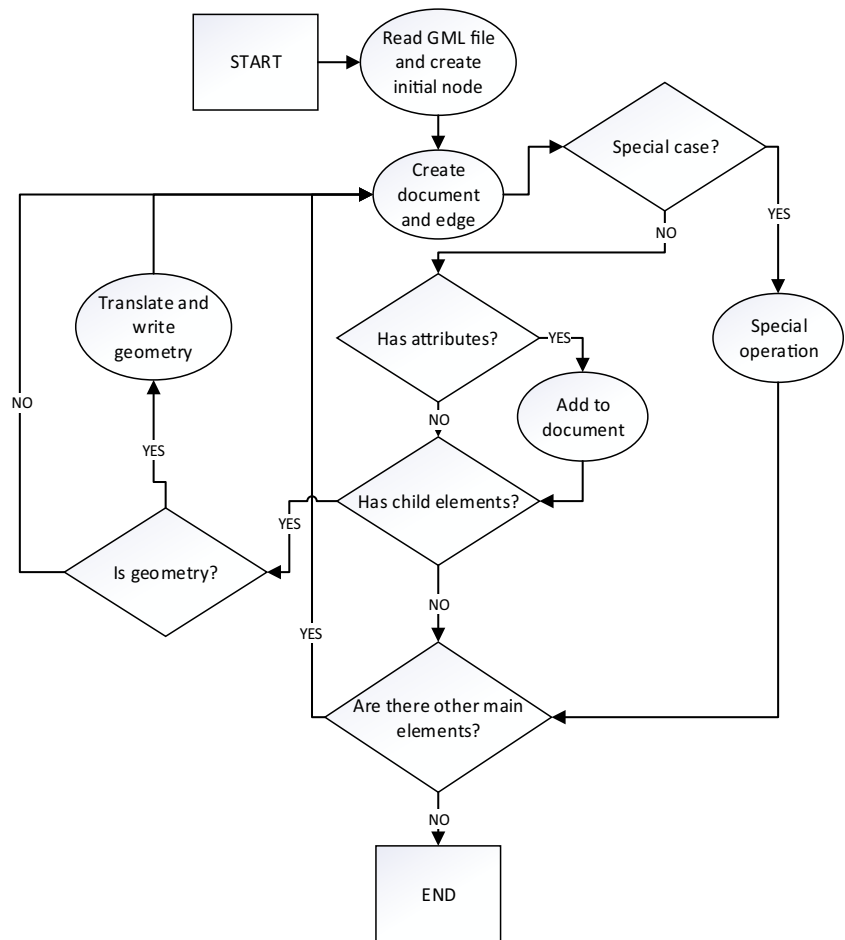


Fig. 7 Comparison between UML schema of CityGML and Graph database. The abstract element AbstractBuilding does not appear in the graph at right because it cannot be created in CityGML model due to GML language specification

Fig. 8 Script workflow



element contains the geometry, then it is translated to JSON and saved in a new document linked by an edge, which its name is the same name of the tag element. The geometry names in CityGML can be seen in its schema and mainly depends of the LOD referred to.

In order to allow linking all elements, a key value is assigned to every document created, using the “gml:id” namespace, if the actual element does not have an ID code. The edges are defined by the key words “\_from,” which is the ID of the parent document, and “\_to,” which is the ID of the actual document. The edge also stores the tag name, which represents the predicate in RDF, inside a string with name “\_predicate.” It creates a new value that is not part of any other ontology. To solve this, the predicate could be the name of the document collection. But this option would create an enormous quantity of files: one file for every kind of element. The visual management of this database would become more difficult and more complicated.

### ArangoDB graph database

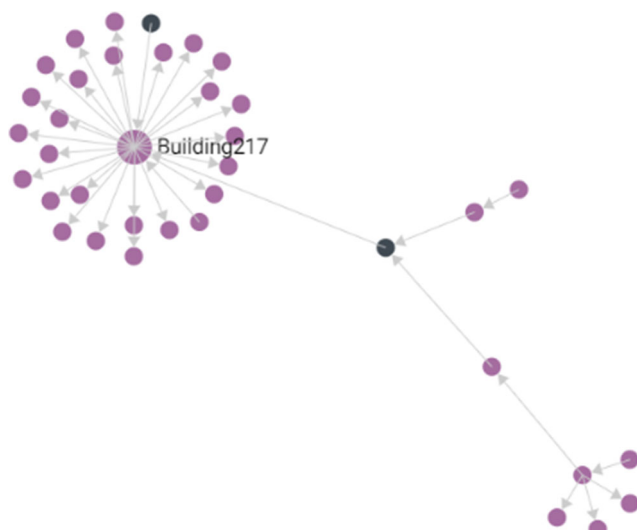
The result of the last procedure described creates two JSON files for each model that can be imported in ArangoDB. One

of them refers to a document collection, while the other one makes a reference to an edge collection.

Referring to the urban model of the case study, in order to carry out a simple and explanatory analysis, only buildings and roads were considered. The number of building features in the selected area is 876, whereas the number of road features is 84. One CityGML and one IndoorGML model were constructed based on available cartography and then encoded in JSON format. CityGML occupies 4.3 MB of disk space and IndoorGML 0.9 MB.

The CityGML model encoded in JSON creates a total of 48,831 documents in 4 MB of disk space, and 48,832 edges in 5 MB to link that documents. The documents created with IndoorGML case are 12,133 in 0.5 MB and 19,905 edges in 2.2 MB. It shows that IndoorGML is mainly used to create links among CityGML objects because of the higher number of edges in relation with documents.

A graph centred in one building is shown in Fig. 9. This graph shows in vertices the value of *\_key* field and along the edges the value of the created *\_predicate* field. At the centre of this graph is represented the vertex with Building217 as *\_key* value (which it is the same ID in the GIS model). Most of the vertices around it are of the same collection, i.e. from



**Fig. 9** Part of the ArangoDB graph representation for the CityGML included

CityGML model, but those placed at right most are from IndoorGML collection. This central node represents a building because it only acts as object in one triple considering only CityGML collection. Also is easily visible that it comes from a predicate *cityObjectMember*, which is the first element of a CityGML.

Every element in the model is always connected to others, so the graph is more extended than shown in Fig. 9. Navigating through the graph model and reaching any searched element is possible using ArangoDB Query Language (AQL). In addition, it allows to carry out queries to see, for instance, which buildings are in a determined route without using any geometrical operation.

To move from one element to another all paths must be previously defined. These paths would be based on predicates and logical relations between elements. Although there are solutions in ArangoDB to search using the shortest path, but sometimes it could report a not correct one. For instance, in this model there are always a minimum of two paths to arrive from one point to other one.

## Conclusions

Data modelling and management software, related to geomatics, urban planning, project management and many others, represent today a fundamental tool in both working and research fields. However, the exchange of data between information systems is still a huge challenge, while, if possible, it could bring enormous benefits to those who work with them.

It becomes necessary to find a solution where data interoperability is lacking, that can be replicated by everyone. This

could be possible using tools already standardized and accessible to everyone, which means basing on defined and known schemas and ontologies. For this reason, it becomes essential to go through CityGML, as it helps the designer to build a standardized ontology. Different tests, to prove the interoperability between information management system, have already been presented by other researchers moving in both directions, especially focusing from GIS to BIM. They try to integrate both spatial and non-spatial data basing on the definition of domains or ontologies, and sometimes using semantic web technology.

The work carried out in our paper aims to create a new enriched model, compared with the standard GIS one and demonstrates that it could be possible to interoperate those modified data towards a new unified data model without losing any information. To do so, we decided to create connections between different elements of the urban model in GIS allowing the interaction among them. This implicates that the formulation of queries is based specifically on these relations, and not only the geometrical elements. We adopt the use of CityGML and IndoorGML schema, but also other standardized format could be selected and added. The graph database represents a useful tool to collect all kind of data, converted in JSON formats or similar, creating RDF triples that connect all the elements and their attributes. The graph database preserves the ontologies during the various data conversions, and it turns out to be a powerful tool in order to improve considerably the performance of data management. Graph databases can query the files contained in them through the new defined ontological schema and basing on the relations created between the nodes.

The work develops the possibility to converge everything that was originally a GIS system in a graph database, based on NoSQL technology. This methodology improves the way in which the identification of affected areas by restoration works of a building is made. It allows making better analysis of operational decisions such as traffic affections, space requirements, affected neighbours and planning alternative routes. Keeping a standardized ontology helps to teammates in planning, managing and operate the building site. Or even to all stakeholders associated to the project as could be the public administration.

Moreover, this research test leaves open the possibility to import BIM data in the same graph database and so to join both GIS and BIM models in the same semantic web environment. To do so, there are many related works which show how to create a unified model which allows to link data (Volk et al. 2014; Song et al. 2017), for instance how to build a common ontology BIM-GIS and eventually how to export data in IFC format that can be successively import in BIM environment. Therefore, in

this paper we try to represent a basic approach that can be replicated in other case studies.

**Funding information** Open access funding provided by Università Politecnica delle Marche within the CRUI-CARE Agreement.

## Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Akinci B, Karimi H, Pradhan A, Wu C-C, Fichtl G (2008) CAD and GIS interoperability through semantic web services. *J Inf Technol Constr* 13:39–55 <http://www.itcon.org/2008/3>
- Anumba C, Pan J, Issa R, Mutis I (2008) Collaborative project information management in a semantic web environment. *Eng Constr Archit Manag* 15(1):78–94. <https://doi.org/10.1108/09699980810842089>
- Beetz J (2009) Facilitating distributed collaboration in the AEC/FM sector using semantic web technologies. Eindhoven University of Technology
- Bishr Y (1998) Overcoming the semantic and other barriers to GIS interoperability. *Int J Geogr Inf Sci* 12(4):299–314
- Butler H, Daly M, Doyle A, Gillies S, Hagen S, Schaub T (2016) The GeoJSON format. Internet Engineering Task Force (IETF)
- Deng Y, Cheng JC, Anumba C (2016) Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison. *Autom Constr* 67:1–21
- Ebrahim PK, Irizarry J (2015) Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services. *Aut Constr* 53:1–12. <https://doi.org/10.1016/j.autcon.2015.02.012>
- El-Diraby T, Lima C, Feis B (2005) Domain taxonomy for construction concepts: toward a formal ontology for construction knowledge. *J Comput Civ Eng* 19(4):394–406. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0000646](https://doi.org/10.1061/(ASCE)CO.1943-7862.0000646)
- Ferdinandus FX, Setiawan EI (2016) GeoJSON Web Service based road assets management system for Surabaya city using mobile GPS. In 2016 International Computer Science and Engineering Conference (ICSEC) (pp. 1-5). IEEE
- Fernandes D, Bernardino J (2018) Graph databases comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB. Proceedings of the 7th International Conference on Data Science, Technologies and Applications (DATA 2018), Porto, Portugal, 26–28 July, 2018, 373–380
- Fosu R, Suprabhas K, Rathore Z, Cory C (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 October 2015
- Hagedorn B, Trapp M, Glander T, Dollner J (2009) Towards an indoor level-of-detail model for route visualization. In *Mobile data management: systems, services and middleware*. MDM'09. 10th international conference, 692–697. IEEE
- Hor A-H (2015) A semantic Web platform for BIM-GIS integration. <https://doi.org/10.13140/RG.2.1.4176.6643>
- Hor A-H, Jadidi A, Sohn G (2016) BIM-GIS integrated geospatial information model using semantic web and RDF graphs. *Int Arch Photogramm Remote Sens Spatial Inf Sci III-4:73–79*. <https://doi.org/10.5194/isprs-annals-III-4-73-2016>
- Hor A-H, Sohn G, Claudio P, Jadidi M, Afhan A (2018) A semantic graph database for BIM-GIS integrated information model for an intelligent urban mobility web application. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci IV-4:89–96*. <https://doi.org/10.5194/isprs-annals-IV-4-89-2018>
- Karan EP, Irizarry J (2015) Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services. *Autom Constr* 53:1–12
- Karimi HA, Akinci B (2009) CAD and GIS integration. Auerbach Publications, Boca Raton
- Kolbe TH, Nagel C (2012) Open Geospatial Consortium OGC City Geography Markup Language (CityGML) Encoding Standard
- Kolbe TH, Gröger G, Plümer L (2005) CityGML: interoperable access to 3D city models. In: *Geo-information for disaster management*. Springer, Berlin, pp 883–899
- Ma Z, Ren Y (2017) Integrated application of BIM and GIS: an overview. *Procedia Eng* 196:1072–1079. <https://doi.org/10.1016/j.proeng.2017.08.064>
- Malinverni ES, Pierdicca R, Giuliano A, Mariano F (2018) A geographycal information system to support restoration activities: a methodological approach experienced upon the case study of Ascoli Satriano Fortress. *Appl Geomatics* 10(4):427–439
- Malinverni ES, Mariano F, Di Stefano F, Petetta L, Onori F (2019) Modelling in HBIM to document materials decay by a thematic mapping to manage the cultural heritage: the case of “Chiesa della Pietà” in Fermo. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci XLII-2/W11:777–784*. <https://doi.org/10.5194/isprs-archives-XLII-2-W11-777-2019>
- Matrone F, Colucci E, De Ruvo V, Lingua A, Spanò A (2019) HBIM in a semantic 3D GIS database. *Int Arch Photogramm Remote Sens Spatial Inf Sci XLII-2/W11:857–865*. <https://doi.org/10.5194/isprs-archives-XLII-2-W11-857-2019>
- Mohammad MT (2010) Understanding semantic Web and ontologies: theory and applications. *ArXiv abs/1006.4567:182–192*
- Murphy, M., McGovern, E., & Pavia, S. (2009). Historic building information modelling (HBIM). *Structural Survey*
- Nagel C, Stadler A, Kolbe T (2009) Conceptual requirements for the automatic reconstruction of building information models from uninterpreted 3D models. Proceedings of the International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 46–53
- Peachavanish R, Karimi HA, Akinci B, Boukamp F (2006) An ontological engineering approach for integrating CAD and GIS in support of infrastructure management. *Adv Eng Inform* 20(1):71–88. <https://doi.org/10.1016/j.aei.2005.06.001>
- Rinaudo F, Agosto E, Ardisson P (2007) GIS and Web-GIS, commercial and open source platforms: general rules for heritage documentation. *Int Arch Photogramm Remote Sens Spatial Inf Sci XXXVI-5/C53, 625–630*

- Sebt MH, Karan EP, Delavar MR (2008) Potential application of GIS to layout of construction temporary facilities. *Int J Civ Eng* 6(4):235–245
- Song Y, Wang X, Tan Y, Wu P, Sutrisna M, Cheng JCP, Hampson K (2017) Trends and opportunities of BIM-GIS integration in the architecture, engineering and construction industry: a review from a spatio-temporal statistical perspective. *Int J Geo-Inf* 6:I. 12
- Vacca G, Quaquero E, Pili D, Brandolini M (2018) GIS-HBIM integration for the management of historical buildings. *Int Arch Photogram Remote Sens Spat Inf Sci* 42(2)
- Van Berlo L, De Laat R (2011) Integration of BIM and GIS: the development of the CityGML GeoBIM extension. In: *Advances in 3D geo-information sciences*. Springer, Berlin, pp 211–225
- Volk R, Stengel J, Schultmann F (2014) Building Information Modeling (BIM) for existing buildings—literature review and future needs. *Autom Constr* 38:109–127
- Yamamura S, Fan L, Suzuki Y (2016) Assessment of urban energy performance through integration of BIM and GIS for smart city planning. In *International High-Performance Built Environment Conference—A Sustainable Built Environment Conference 2016 Series (SBE16)*, IHBE
- Zhang X, Arayici Y, Wu S, Abbott C, Aouad GF (2009) Integrating BIM and GIS for large-scale facilities asset management: a critical review, in: *12th International Conference on Civil, Structural and Environmental Engineering Computing*, Portugal
- ## Webs
- ArangoDB (2019) <https://www.arangodb.com/> Accessed 15 July 2019; <https://www.arangodb.com/why-arangodb/multi-model/> Accessed 15 July 2019
- ArcGIS (2018) <https://desktop.arcgis.com/en/> Accessed 12 December 2018
- CityJSON (2019) <https://www.cityjson.org/> Accessed 15 July 2019
- FME (2019) <https://www.safe.com/fme/fme-desktop/> Accessed 15 July 2019
- GDAL (2019) <https://gdal.org/> accessed on 15 July 2019
- GeoJSON (2019) <https://geojson.org/> Accessed on 15 July 2019
- IndoorGML (2019) <https://www.opengeospatial.org/standards/indoorgml> Accessed 15 July 2019; <http://www.indoorgml.net/> Accessed 15 July 2019
- ISO/TC 211 (2019) <https://committee.iso.org/home/tc211> Accessed 15 July 2019
- MongoDB (2019) <https://www.mongodb.com/> Accessed 15 July 2019
- Neo4j (2019) <https://neo4j.com/> Accessed on 15 July 2019
- OGC (2019) <https://www.opengeospatial.org/standards/citygml> Accessed on 15 July 2019
- Python (2018) <https://www.python.org/> Accessed 12 Dec 2018
- Wiki.GIS (2019) <http://wiki.gis.com/wiki/index.php/Interoperability> Accessed 1 July 2019