A BIM - GIS INTEGRATED INFORMATION MODEL USING SEMANTIC WEB AND RDF GRAPH DATABASES

ABDELHADI HOR

A DISSERTATION SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

GRADUATE PROGRAMME IN EARTH AND SPACE SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

SEPTEMBER 2022

ABSTRACT

In recent years, 3D virtual indoor and outdoor urban modelling has become an essential geospatial information framework for civil and engineering applications such as emergency response, evacuation planning, and facility management. Building multi-sourced and multi-scale 3D urban models are in high demand among architects, engineers, and construction professionals to achieve these tasks and provide relevant information to decision support systems. Spatial modelling technologies such as Building Information Modelling (BIM) and Geographical Information Systems (GIS) are frequently used to meet such high demands. However, sharing data and information between these two domains is still challenging. At the same time, the semantic or syntactic strategies for inter-communication between BIM and GIS do not fully provide rich semantic and geometric information exchange of BIM into GIS or vice-versa. This research study proposes a novel approach for integrating BIM and GIS using semantic web technologies and Resources Description Framework (RDF) graph databases. The suggested solution's originality and novelty come from combining the advantages of integrating BIM and GIS models into a semantically unified data model using a semantic framework and ontology engineering approaches. The new model will be named Integrated Geospatial Information Model (IGIM). It is constructed through three stages. The first stage requires BIM_{RDF} and GIS_{RDF} graphs generation from BIM and GIS datasets. Then graph integration from BIM and GIS semantic models creates $IGIM_{RDF}$. Lastly, the information from $IGIM_{RDF}$ unified graph is filtered using a graph query language and graph data analytics tools. The linkage between BIM_{RDF} and GIS_{RDF} is completed through SPARQL endpoints defined by queries using elements and entity classes with similar or

complementary information from properties, relationships, and geometries from an ontology-matching process during model construction. The resulting model (or sub-model) can be managed in a graph database system and used in the backend as a data-tier serving web services feeding a front-tier domain-oriented application. A case study was designed, developed, and tested using the semantic integrated information model for validating the newly proposed solution, architecture, and performance.

This research has also introduced new concepts of nodes, relationships, and property occupancy indexes to measure integration accuracy and data richness, providing deeper insights into a BIM-GIS integrated semantic model. These new parameters can be used in Decision Support Systems (DSSs) and Operation Dashboards (ODs) to select information and build conceptual sub-systems allowing practitioners to perform complex data analytics to evaluate BIM-GIS integrated models. The occupancy indexes can be used in Machine Learning / Deep Learning models in future 3D Urban designs for smart cities, intelligent utility management, and risk prediction.

بسم الله الرحمن الرحيم

Dedicated to the memory of the beloved El-Hadj Benyoucef Bayou

(1929 - 2021)

ACKNOWLEDGEMENTS

This thesis is only possible with the support of many people. I want to extend my sincere appreciation to them for their help and support throughout my research.

It would be difficult to overstate my gratitude to my supervisor, Dr. Gunho Sohn, who provided me with excellent guidance and encouragement. His academic passion inspires me to grow into a better scientist. With his valuable advice, I can accomplish my research and this dissertation.

I want to extend my gratitude to the committee members, Dr.Costas, Dr. Regina Lee, and Dr. Jadidi, who offered valuable advice, insightful discussions, and suggestions on my research. Such great help from them resulted in a considerable improvement in this dissertation.

Above all, my heartfelt and everlasting gratitude goes to my family, especially my parents, my wife, Amina Bayou, and my most lovely daughters, Myriam, Yasmine, and Nadine, for their patience, encouragement, and unconditional love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AEC	Architecture/Engineering/Construction
API	Application Programming Interface
AIM	Alignment, Merging and Integration
AOM	Application-Oriented Ontological Model
ATO	Assembled-To-Order
BIM	Building Information Modelling
BIMQL	Building Information Model Query Language
CAD	Computer-Aided Design
CEGIS	Center of Excellence for Geospatial Information Science
CityGML	City Geography Markup Language
DEM	Digital Elevation Model
DOOM	Data Ontological Oriented Model
DTM	Digital Terrain Model
ETO	Engineered-To-Order
FM	Facility Management
GIS	Geographic Information Systems
GML	Geographic Markup Language
GMO	Graph Matching for Ontologies
GPS	Global Positioning System
GVW	Gross Vehicle Weight
IFC	Industry Foundations Classes
IGIM	Integrated Geospatial Information Model
IT	Information Technology
LOD	Level of Detail
LoD	Level of Development
KPI	Key Performance Indicator
LiDAR	Light Detection and Ranging
NBIMS	National Building Information Model Standard
NIST	National Institute of Standards and Technology
OGC	Open Geospatial Consortium

OWL	Web Ontology Language
RDF	Resource Description Framework
TIN	Triangulated Irregular Network
URI	Unique Resource Identifier
USGS	United States Geological Survey
W3C	World Wide Web Consortium
WGS	World Geodetic System
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Motivation

Integration between Building Information Models (BIM) and Geographic Information Systems (GIS) has been a significant challenging problem facing practitioners and researchers from these communities. Such integration aims to meet increased demand for building systems, sub-systems analysis, urban design planning applications, disaster management, homeland security, and many other enterprise and engineering applications. These applications require 3D geometry and appearance information and complex semantic information, and the ability to perform analysis on data from both domains, thereby benefitting from integration.

GIS and BIM originated from two separate domains and were developed to suit professionals' specific needs within their respective fields. BIM systems, applications, and tools mainly focus on generating objects with full information about objects and their geometries. Geospatial systems, applications, and tools are used to analyze objects that already exist around us.

The predominant data representation standard for GIS data is the City Geography Markup Language (CityGML), while the Industry Foundation Classes (IFC) language is more commonly related and used for BIM. These two formats consist of different kinds of information at different levels of detail, leading to integration and interoperability difficulties in data sharing or conversion between them. Furthermore, there are incompatibilities in every aspect of the GIS and BIM domains (Fig 1.1). Yet, using GIS and BIM approaches separately but simultaneously in architecture, engineering, and construction (AEC) and geospatial applications motivates efforts to develop integration and interoperability between the two platforms.

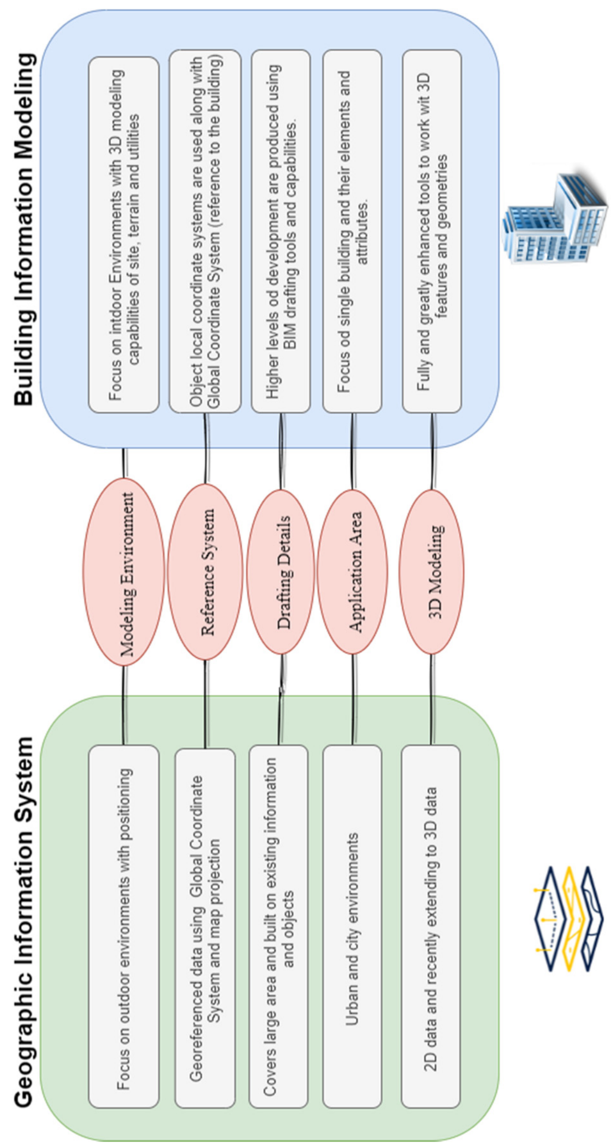


Figure 1.1 Incompatibility between GIS and BIM domains

Integration efforts by researchers in each of the GIS and BIM communities are developed as conceptual models. These models include a geometric model designed for 3D object visualization purposes or semantic models designed for various planning and engineering applications that require complex filtering queries and advanced data patterns analysis. Other trends in BIM-GIS integration efforts focus on data conversions between the two data formats. Translation algorithms are designed and coded into many BIM-GIS packages and software extensions, allowing access and use of geospatial data from within BIM and BIM data within a geospatial context. The data is read-only and subject to many limitations in most use cases. For example, GIS typically expects all features to be drawn in a pre-defined geographic coordinate system with a specific projection, unlike BIM, which does not have this property. Similarly, BIM applications cannot handle scale-dependent rendering, limiting the level of details in a map view depending on view scale.

The two approaches to BIM-GIS integration can be categorized as syntactic and semantic. The syntactic approach of data integration involves gaining access to shared data in the other domain. In contrast, the semantic approach incorporates the information from the two knowledge domains into a new information model based on a defined ontological representation with specific rules and taxonomies. Ontology is the formal naming and definition of the categories, properties, and relations between the concepts, data, and entities in a specific knowledge domain. Integration based on semantics provides the highest level of any integration between the BIM and GIS domains. The semantic approach with ontology representation is the most promising integration method that provides the footprint of this research project. Ontology representations allow structuring and organizing domain knowledge about objects so that software can automatically process and integrate a large amount of information without a predefined interface of any human interaction. The key to integration at the semantic level is to ensure the integrity between objects and relationships between the domains are maintained during the data transfer.

This study focuses on using ontologies to represent both BIM and GIS domains. It provides a robust framework for representing, sharing, exchanging, and managing domain knowledge through machine-readable, understandable descriptions that define the object taxonomies and their associative relations across domain relationships. Semantic web and data representation technologies, such as resource description framework (RDF), provide standard taxonomies and ontologies for BIM and GIS. Successful integration of the two systems depends upon thoroughly combining the strengths of both systems in the contexts of each other (Jiang et al., 2013).

This research considers the semantic web a unifying platform and the common framework for developing a semantically integrated information model that combines concepts from BIM and GIS systems. The semantic integration methods enable the bidirectional conversion between BIM and GIS. As a result, they are more flexible than any other method, with a defined ontology available for future use cases involving BIM and GIS. More importantly, this integration approach preserves the semantic information specified in both domains while enabling data integration on the semantic level. The research will demonstrate the potential of using graph theory concepts and graph database management systems to organize, manage and visualize enormous information with complex relationships of the RDF-based integrated BIM-GIS model.

1.2 Research Objectives

This study aims to use ontology engineering methodologies to integrate BIM and GIS models into a unified model according to the semantic web framework stack. Data semantics and information models are required based on resource description framework (RDF) and graph databases. A matching process is then created between BIM industry foundations classes (IFC) and the GIS City Geographic Markup Language (CityGML). The matching will be based on entities representing the same components in both models. This will demonstrate that GIS spatial (outdoor and location) and BIM non-spatial (indoor and dimension/materials) information from various sources can be integrated and analyzed within a single ontological model. This constitutes the objective of creating a semantic integration approach to GIS and BIM.

The proposed semantic integration approach presented in this work consists of four main phases:

1. The construction of BIM and GIS ontology models and semantic integration using interoperable data formats are determined and evaluated.
2. BIM and GIS models are imported into a graph model with querying and filtering capabilities.
3. The workflow and transformations pipeline is designed and developed to extract and load the IFC and CityGML RDF graph data model.
4. The model is evaluated for accuracy, data richness, and performance and then used as a data backend tier for an intelligent urban mobility web application.

1.2.1 General Research Framework

Figure 1.2 represents the different phases and workflows for a BIM-GIS integration modelling process using a resource description framework (RDF) and property graph databases. The diagram shows the interrelation between the phases proposed in this research work.

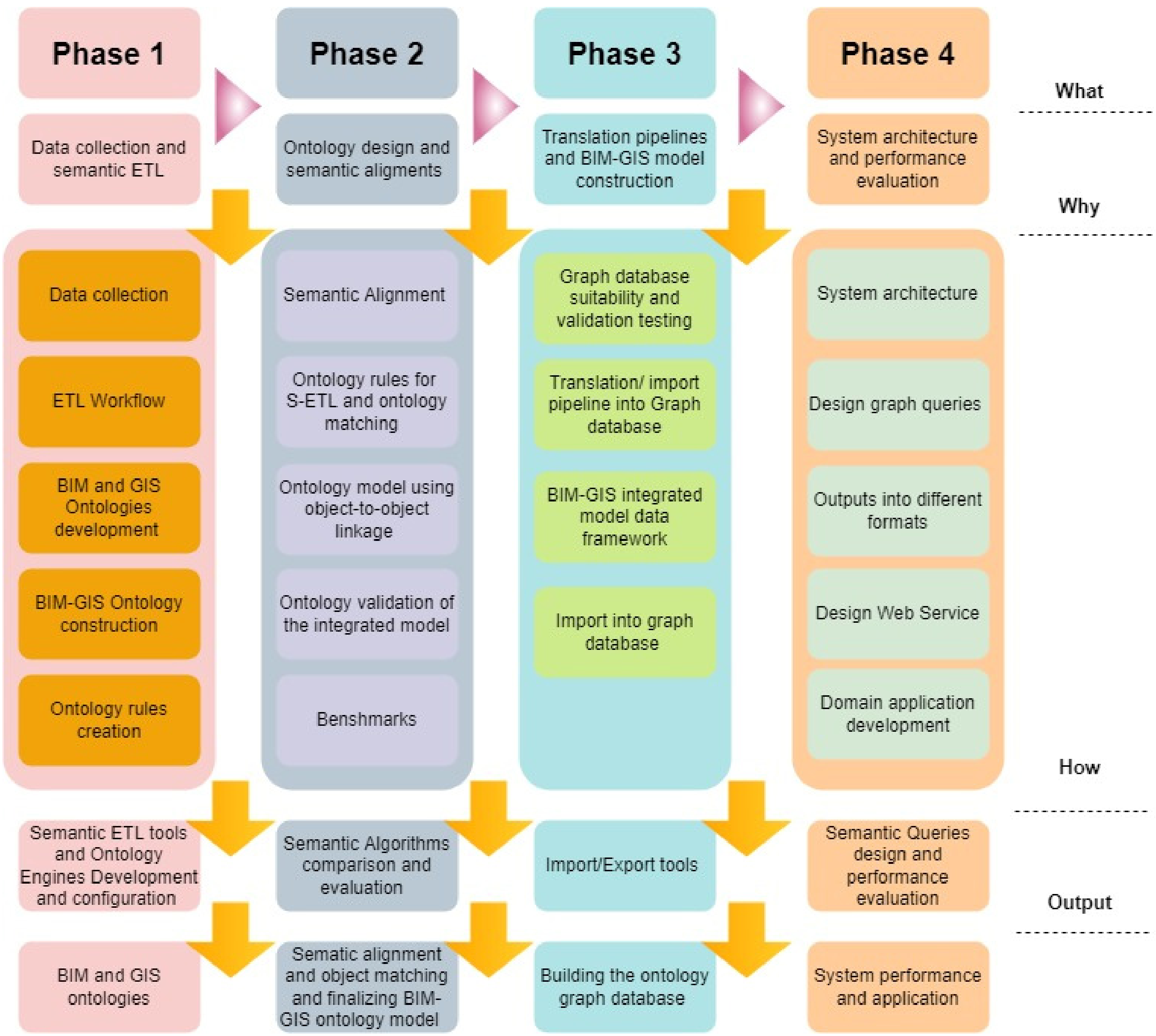


Figure 1.2 Proposed BIM-GIS semantic integration modelling using RDF property graph databases.

Phase I involves data collection and cleansing of BIM and GIS data models in their native formats, IFC and CityGML. The workflows are used as semantic Extract -Transform - Load (ETL) tools for RDF-based models. This same RDF base framework is used to develop the BIM-GIS integrated semantic model.

In the second phase, multiple ontology alignment techniques are applied to retrieve, compare and integrate objects, properties, and subjects from BIM and GIS semantic models based on predefined ontology rules and input parameters. This process consists of six main stages:

1. Comparison of multiple selected semantic matching alignments systems can be applied to BIM and GIS models.
2. Development of a set of ontological properties for the triple-store to select, store, and retrieve triples.
3. Creating benchmarks for semantic alignments requires creating a testbed and extensive testing to create and validate benchmarks.
4. Revisit ontology-based matching using graph matching for the ontologies algorithm to align BIM and GIS graphs.
5. Correlation/linkage of similar concepts, node properties, and relationships based on equivalency, common properties, and similar relationships to other objects.
6. Validation of the integrated data model based on the RDF data model.

The third phase proposes a graph database management system for the semantically integrated BIM-GIS model. It can store, maintain, and analyze the integrated model using tools and workflows from graph databases system by

- Comparison and evaluation of multiple existing graph databases system based on features and criteria related to the specifications of both BIM and GIS domains of knowledge.
- Export of BIM and GIS elements to the graph databases system.
- Development of loading and conversion workflows.
- Establishment of an ontological infrastructure for BIM-GIS integration.

Lastly, a system architecture design for implementing a BIM-GIS semantic integration RDF-based model is created in the fourth phase. The model is evaluated using the Web GIS application from Web Scene Service based on the BIM Server and ArcGIS platforms.

1.2.2 Contributions

This research uses ontology-engineering-based methodologies, a new conceptual modelling platform, and newly developed system architecture software to integrate BIM-GIS into a unified platform. It uses a reference information framework based on RDF property graph databases. It includes modules of components and workflows that contribute to a complete solution that integrates objects and elements from BIM and GIS domains. More specifically, the contributions of this study can be summarised as follows:

1. Generation of an Application-oriented Ontological Model (AOM)-based target application domain by querying the BIM and GIS resource description framework data from a graph database perspective. This requires developing RDF representations of BIM and GIS data. RDF representational structures for BIM-GIS

integrated modelling are analyzed, and the current RDF query language is studied to support graph queries for different application domains.

2. Construction of a BIM-GIS Data Ontological Oriented Model (DOOM) to represent semantic information from IFC and CityGML information, respectively. These ontologies help develop integration and filtering modules of heterogeneous attributes and spatial data and provide query language to access and acquire the data in semantic web format.
3. Evaluation of multiple semantic matching techniques for BIM and GIS objects and mapping elements to achieve the best Alignment, Merging, and Integration (AMI) techniques between candidate BIM and GIS objects based on their equivalence, attribute similarities, or functionalities within the model. The evaluation is based on scrutiny of the literature and a critical review of works from various fields and knowledge domains to provide a comprehensive overview of ontology mapping and matching techniques. The complete comparative and evaluative studies are conducted against semantic models, and results are presented and interpreted based on performance metrics.
4. Introduction of new concepts of nodes, relationships, and property occupancy indexes to measure and evaluate data richness and accuracy of BIM-GIS semantically integrated models. These indexes are introduced in this research study and offer insight into the BIM-GIS semantic strategy by providing in-depth information regarding the level of integration between IFC classes CityGML elements in the same model. They can be used as parameters in a decision support

system dashboard to compare between models using nodes, relationships, and properties in these models.

5. Creation of rule-based integration modelling workflows and rule-based selection methods using a property graph database that accounts for BIM and GIS models and the complex relationship between model entities. This method permits access to semantic information that could remain inaccessible in many cases due to closed property formats or the absence of suitable data management tools when using open IFC and CityGML standards. In addition, the rigid and complex hierarchical structure of the BIM and GIS schemas can prevent the extraction of information and requires a deep understanding of both BIM and GIS models. The separate BIM and GIS query languages have limitations, particularly for high-level IFC and CityGML object models. This research proposes graph-based data modelling, which has great potential in understanding and accessing complex and rich data. Graphs are well suited for representing and describing complex interrelationships among BIM and GIS elements. The proposed use of graph database management systems to store, manage and retrieve BIM and GIS semantic data significantly eases integration workflows between these knowledge domains and provides better tools for exploring and analysing connected data.
6. Introduction of a new flexible semantic data translation module to manage, maintain and analyse the BIM-GIS integrated model. This module provides a solid framework for storing, accessing, and manipulating complex relationships by utilizing graph data mining capabilities in BIM and GIS domains.

1.3 Thesis Outline

This thesis is organized into seven chapters. An overview of the chapters follows:

Chapter 1 introduces the thesis motivation and the proposed methods and strategies for meeting the research objectives.

Chapter 2 provides background information that aids in understanding the problem we seek to solve and a comprehensive literature review concerning BIM and GIS integration methodologies and strategies.

Chapter 3 introduces the research methodology for integrating building information modelling (BIM) and its industry feature classes to the Geospatial Information Systems (GIS) domain based on ontology mapping methods. The integration phases are presented for IFC and CityGML structures ontology models and semantic web methods for these models.

Chapter 4 proposes semantic alignment techniques to evaluate the ontology graph matching algorithm for the BIM-GIS integrated Model.

Chapter 5 introduces a semantic graph database for BIM-GIS Integrated Model.

Chapter 6 provides the validation of the integrated semantic model and an implementation of the system architecture using the new BIM-GIS graph model web platforms.

Chapter 7 presents the results and conclusions of this study, provides details of the challenges/limitations encountered during the project, and gives future work recommendations.

Chapter 2

Background and Literature Review

In the last decade, the integration between Building Information Modelling (BIM) and Geographic Information System (GIS) has mostly focused on geometry information and model visualisation. There has been little research on other integration and interoperability aspects utilizing attributes, values, and relationships from objects in BIM and GIS domains. The best integration methodology between BIM and GIS has never developed or reached its highest level. This is due to thematic modelling based on layer-based features in geographic information systems, file schema, and object structure in which BIM data is stored and presented in different systems. This chapter reviews and examines the dissimilarities between GIS and BIM data models and investigates their potential in numerous applications to understand these two important knowledge domains adequately. This chapter will show that many integration methods of the GIS and BIM domains have been developed to solve different engineering and environmental problems. It will also demonstrate that using semantic web technologies as a BIM-GIS integration platform and framework can provide a promising and complete integration solution. However, the most significant challenges are the considerable efforts required in the early development stages of each domain ontologies. We conclude this chapter by discussing the challenges for BIM-GIS integration remaining unsolved and unaddressed questions, for which we will provide our solution approaches in the later chapters.

2.1 Introduction

In the last few years, the efforts to integrate Building Information Modelling (BIM) and Geographic Information Systems (GIS) have had an enormous impact on many aspects of many applications and systems in the field of Architecture, Engineering, and Construction (AEC). This integration has become a focus of many research and development areas in academia and industry for building smart city services (Bansal and Pal, 2007; Karimi and Akicini, 2014), providing economic growth, sustainability, and citizen engagements (Bansal and Pal, 2009).

This chapter presents the literature on each domain and the most up-to-date integration efforts (Adachi, 2003; Bakis et al., 2016) in geospatial and building information modelling. This chapter is structured as follows: first, GIS and its data representation through CityGML is presented, followed by BIM and its data representation through IFC. Ontology-based integration frameworks are discussed, followed by a close examination of BIM-GIS semantic integration efforts (Adachi, 2003; Cromley and McLafferty, 2012). Then, existing integration, interoperability patterns, and domain-oriented applications for BIM-GIS integration are presented. Finally, BIM-GIS semantic integration challenges are discussed (Berners-Lee, 2006; Borrmann et al., 2006).

2.1.1 Geographic Information System and CityGML

The geographic information system (GIS) is known as a well-established decision support system and a complete information system designed to store, analyse, manipulate, and manage geospatial information (Bakis et al., 2007; Ruy et al., 2017). The features stored in the GIS system are georeferenced and contain attributes with properties associated with the spatial layer's geographic location, all of which are important for geoprocessing, temporal, and spatial analysis (Adachi, 2003; Brunnermeier and Martin, 2002). Spatial data share the same spatial reference (Karan and Irizarry, 2014) information in a single GIS system. They are linked together through relationship classes and topological rules to form one abstract model that defines the physical and behavioural structure of the GIS domain (Bakis et al., 2007). The GIS system provides architecture, engineering, and construction (AEC) practitioners a geospatial context and insight with rich tools set to design, construct, and manage buildings, utilities, and infrastructure by providing easier information access and better visualization (Hadi et al., 2016; Beetz, 2009).

In recent years, GIS systems have started to support 2D-based mapping and advanced data analysis (Agdas and Ellis, 2010; Berners-Lee, 2006; Ma et al., 2005) and provide technology and patterns for 3D applications and services. These 3D systems enhance capabilities in many areas, such as utilities, transportation, and urban design (Karan and Irizarry, 2014). One of the most popular and comprehensive standards of 3D data exchange is CityGML (Deutsch, 2011) data model. The CityGML is an XML-based file format to store, share and manage rich semantic 3D geospatial models (Lima et al., 2005; Teicholz, 2013). It was developed to support design processes (Ma and Ren, 2017) which are handled with the concept of level of detail (LODs) (Alders, 2006; Beetz, 2009), as shown in fig 2.1. The highest level of detail provides the most complex and accurate model representation (Jena, 2011; Valcik and Huesca-Dorantes, 2003).

The latest version of CityGML 3.0 (CityGML | OGC) includes many new features, attributes, mechanisms, and much richer 3D modelling, which is very close to BIM modelling (Ma and Ren, 2017). The most recent version offers time-dependent properties with versioning capabilities to administer multiple versions of building and even city models (Ma and Ren, 2017; Hbeich et al., 2020), representation of city objects using point clouds dataset, new features for traffic infrastructure, and separation of the conceptual model (Alders et al., 2006).

2.1.2 Building Information Modelling and IFC

Building Information Modelling (BIM) is a digital 3D model-based dimensional representation of buildings, bridges, tunnels, roads, and utilities (Berners-Lee, 2006). It creates and manages model representation by including its physical and functional characteristics. In a BIM, the view of the whole building and its elements are in 3D, allowing users and professionals to understand the geometries and the elements' relationships in one unified system (Azhar et al., 2011; Cromley and McLafferty, 2012).

The Building Information Modelling project is all about and around data presented at different levels of development of data and provides consistent information for everyone involved in the project. It helps with every phase of the project workflow cycle, from design and conception to construction, documentation, and maintenance (Quantum GIS, 2013; Pauwels et al., 2013). The building blocks of a BIM are object-oriented 3D objects representing building elements such as walls, doors, windows, columns, and slabs following an oriented object structure describing the elements' shapes (geometry) (Karimi and Akicini, 2014). The objects' dimensions within BIM models also store information about specifications, material, building code, manufacturer, price, warranty, etc., and relationships with other entities in the same BIM (Ma et al., 2005; Underwood and Isikdag, 2011), defining the model's behaviour. One of the BIM objects' standard representations is the Industry Foundation Classes (IFC) (Akicini et al., 2010; Cutting-Decelle et al., 2007).

IFC is a set of vendor-neutral standardized object-oriented definitions describing the representation of building components designed and developed by BuildingSmart (buildingSMART - The International Home of BIM). IFC is also a common open-source language for storing and sharing intelligent building objects. The IFC data is classified into three types (Azhar et al., 2011; Underwood and Watson, 2003). The first type is geometry data representing the 3D shape of the physical component of the building. The second type is the property data describing the object. The property data could be either an attribute or a property of the item in the model. The third type is the relationship data, which defines the link between different building components (Cromley and McLafferty, 2012).

Figure 2.1 BIM level of development

2.2 BIM-GIS Differences and Current Integration Models

The development of BIM and GIS integration methodologies depends on the perspective and the vision of each domain's practitioners and the target application during the construction of the integration data model (Bakis and Pal, 2009a; Eskrootchi et al., 2008). Indeed, the differences between BIM and GIS models are critical during the life cycle. These differences have been the focus of academic research studies and industry efforts (El-Gohary and El-Diraby, 2010), particularly during the last ten years. The integration efforts have begun by attempting to move BIM-IFC datasets into and from GIS-CityGML to use the rich information from BIM to feed the advanced spatial analysis capabilities of the GIS platform (Li et al., 2003; Ma et al., 2005). Many of these works have given good results. However, it has always been a challenge to overcome many of the mismatches between the two domains (Bakis and Pal, 2009a; Deutsch, 2011), such as:

- Coordinates systems: IFC local coordinates versus GIS global (geographic).
- Semantic and geometric representations.
- Level of details and information granularities.
- File format and data management.
- Spatial scales and application focus.

With the recent development of indoor GIS, mobile applications, and assets and facilities management systems (Crowther and Hartnett, 2001; Hijazi et al., 2011), the demand for BIM-GIS integration has reached very high-interest levels. Hence, their structural differences are increasingly frequent subjects of research and professional focus (Bakis and Pal, 2007; Ma and Ren, 2017).

2.3 Ontology-based Integration Frameworks and Patterns

Ontology is a knowledge structure built to identify objects, their attributes, and the dependencies between them through the meaning of unstructured data attributes, values, relationships, and instances (Hadi et al., 2016; Karan et al., 2014). Data modelling using ontology engineering methodologies is at the core of the process of ontology-based data integration. Ontology engineering consists of tasks and activities to develop and construct a conceptual ontology model. It provides standards and tools to preserve, manage and maintain the ontology model life cycle (Irizarry et al., 2013; Walker et al., 2005). The process will translate object definitions and attributes into vocabularies and relationships to field-specific formal ontologies (Hadi et al., 2016; Karan and Irizarry, 2014). The language specifications will allow interoperability, integration, and data retrieval from multiple data sources and fields.

In the last decade, and due to advanced research in artificial intelligence and machine learning fields (Bakis et al., 2007; Cheng and Yang, 2001), ontologies have started to be the focus of research in computer science and many other research areas like domain knowledge representation (Karan and Ardeshir, 2008), and information retrieval and integration. Therefore, the scientists are taking the integration questions and tackling challenges of ontology-based integration models such as:

- Graph-based models ontology integration.
- Artificial intelligence and Machine Learning based Integration Approaches.
- Constructor-based ontologies integration.

At the same time, semantic web theory has come with a vision of extending the world wide web by providing software programmes and systems with machine-interpretable metadata built upon two major components. This will be further explained in Section 2.4.2: The Resource Description Framework (RDF) (Karan and Ardeshir, 2008; Karan and Irizarry, 2014), which is the ontology standard model for exchanging data and knowledge about things and their relationships; and the Ontology Web Language (OWL), which permits objects manipulations and filtering queries of the RDF- based content (Karan et al., 2014).

2.4 BIM-GIS Semantic Integration

Semantic integration of BIM and GIS introduces new methodologies, standards, and data translation within a semantic web framework built on semantic web technologies, resource description frameworks, and RDF graph databases, as explained below.

2.4.1 Semantic Web Technologies

Semantic web technologies refer to platforms, systems, and applications that interact and work with linked data from different sources or platforms (Hadi et al., 2016; Hor and Sohn, 2021). These technologies provide rich, robust, distributed platforms with powerful tools to enable users to create data stores, build dictionaries, and construct vocabularies powered by rules to handle and share data from various knowledge domains (Anumba et al., 2008; Karan and Ardeshir, 2008). A semantic web (also known as web 2.0) is extended from the current web platform, but it provides tools to define information differently than regular web users.

In a semantic web, the focus is on the meaning of data or the information and interrelationship between these objects in a single semantically defined model. A new semantic data model deals very well with the four types of heterogeneities known in distributed databases systems (Karan et al., 2014; Wang and Xue, 2008):

- **Structural heterogeneity**: related to various data formats, model structures, and data schemas stored in diverse systems.
- **Semantic heterogeneity**: originated from various data definitions to different concept meanings.
- **Syntactic heterogeneity**: given different encoding and representations from various data sources.
- **System heterogeneity**: hosted application and underlying dataset stored on machines with different operating systems and specifications.

Taking these considerations into account with the fact that BIM and GIS domains have different schema, file structures, and contents (Hor et al., 2018; Pauwels et al., 2016), using semantic web technologies methods to integrate BIM and GIS is worth investigating (Karan and Ardeshir, 2008). The semantic platforms will provide a robust framework for bidirectional ontology translation between the two domains using reference ontology to define, represent and load object differences ontologies from BIM and GIS that already exists (Karan et al., 2014; Liu et al., 2017).

2.4.2 Resource Description Framework (RDF) Core Model

Resource Description Framework is the core of a semantic web platform, also known as the RDF model (RDF - Semantic Web Standards (w3.org)). The RDF combines multiple knowledge domains by flexibly and efficiently representing the information using graphical formalism to express data models (Hadi et al., 2016; Laat and Berlo, 2011; Karan and Irizarry, 2014). In an RDF-directed labelled graph network, each node in the representation is an object (or a concept) uniquely identified using a Unique Resource Identifier (URI). The RDF objects with their attributes and values information can be exchanged and then shared between different platforms, systems, and applications using various syntaxes such as N-triples, Turtle, Notation-3, and RDF/XML (Bakis et al., 2007; Karan and Irizarry, 2014; Törmä et al., 2012). The most fundamental semantic structures, such as class, sub-class, data types, and other ontological specifications, are included in RDF-Schema (RDFS) vocabulary (Karan et al., 2014). The most efficient method to manipulate and query ontological information is through the Web Ontology Language (OWL) (Hadi et al., 2016; Hor and Sohn, 2021; Karan et al., 2014), using rules and proofs to construct objects, filter complex information, and build integrated systems (Alders, 2006; Cardoso and Sheth, 2006).

Regarding BIM-GIS integration, parsing data from both BIM and GIS models into a common data model is key to reaching the best integration possible (Hadi et al., 2016; Hor and Sohn, 2021; Karan and Ardeshir, 2008). A new ontological data model for a BIM-GIS integrated system will be able to consider all structured and unstructured information from these domains and assure accuracy, scalability, reliability, and performance of processes with no loss of geometries, relationships, properties, or semantics. Most importantly, the model could add and digest new information enhancing the integration (Bakis and Pal, 2009b; Kim and Grobler, 2007).

2.4.3 RDF Graph Databases

RDF graph databases are a W3C (World Wide Web Consortium (W3C)) standard data model built using graph theory (Berners-Lee, 2006). It is designed to store and manipulate every piece of semantic information in a graph, including relationships, attributes, and particular data types (Hor and Sohn, 2021; Kim and Grobler, 2007; Karan and Irizarry, 2014). In the same way, it treats objects in the data model. The RDF, discussed in the section above, is at the core of the semantic web framework. The RDF graph database model is the best-fitted database platform for representing and manipulating unstructured metadata and complex data models (Döllner and Hagedorn, 2007). RDF graph databases organize the information into sets of triples and display it as a graph (Hor and Sohn, 2021; Bakis and Pal, 2007). The SPARQL (SPARQL Query Language for RDF w3.org) query language can extract, import, and export data from graph databases. SPARQL comes in very handy for BIM-GIS integrated sub-modelling tasks in specific applications in which a subsystem for a particular functional subsystem is needed, such as in asset management operations of Mechanical, Electrical, and Plumbing (MEP) models in a BIM life cycle phases (Bakis and Pal, 2009a; Hijazi et al., 2011; Karan and Ardeshir, 2008)

Using RDF graph databases in a BIM-GIS integrated model can offer tremendous advantages such as:

- Using Ontologies as semantic schemata.
- Integrating multiple data sources.
- Filtering workflows against diverse data schemata.
- Providing efficient data interoperability.
- Providing powerful analytical tools.

Nonetheless, designing, developing, and implementing RDF graph databases is a complex process involving many basic multiple interconnected workflows (Hijazi et al., 2011; Karan and Ardeshir, 2008), such as:

- Data extraction, transformation, and loading: including parsing and indexing.
- Query design and evaluation: from designing SPARQL queries to optimizing and fetching them.
- Data and schema updates: possibly including changing object-ontologies definitions and schemas.
- Semantic inference (Inference - W3C) processes affecting system performance, particularly during the ETL (Extraction, Transformation, and Loading) phase.

The RDF graph databases for a BIM-GIS integrated model can be enormous (billions of triples). Hence, it is imperative to have a well-defined strategy for each of the workflows below with adequate methodologies to deal with each cycle in the complete cycle, from database modelling, data loading, query design, and optimization (Hor et al., 2018; Hor and Sohn, 2021).

A semantic BIM-GIS RDF graph database BIM-GIS system can fall into any of three paradigms, i.e., unification, federation, or integration (Hadi et al., 2016; Liu et al., 2017):

- **Unified Model**: A mapping approach at the meta-model level uses a common format. Objects in this model are pre-defined for semantic equivalency (Hor et al., 2018; Hor and Sohn, 2021).
- **Federated Model**: There is no common format in this model. The federated model is dynamically created to accommodate sources’ datasets into a shared ontology model where object mapping is completed (Hor et al., 2018; Karan and Ardeshir, 2008; Karan and Irizarry, 2014).

- **Integrated Model**: This model is based on format standardization for all source models, and mapping is completed to the standard of the integration model (Hadi et al., 2016; Karan and Ardeshir, 2008).

2.5 Current Patterns and Applications for BIM-GIS Integration

A BIM-GIS integration system is triggered by the necessity to merge data between objects from the two knowledge domains. It is driven by any of the following use cases (Hor and Sohn, 2021; Ruy et al., 2017):

- Utilizing architecture, engineering, and construction (AEC) design data in geospatial workflows to assess planning, maintenance performance, and emergency response operations (Hor et al., 2018; Ergen et al., 2007, Karan and Irizarry, 2014).
- Providing BIM design teams with geospatial information to help understand and achieve efficient and reliable projects (Hor et al., 2018; Ergen et al., 2007; Karan et al., 2014).
- Increasing situational awareness within the asset management systems and projects for better communications during the life cycle of BIM-GIS projects (Karan and Ardeshir, 2008).

These cases have divided the BIM-GIS integrated applications into two major categories (Andrews, 2020):

- **Category 1**: Phase-based applications. These are applications based on a BIM-GIS integrated system for the limited phase of a project in specific fields, such as in the construction safety field (Ma and Ren, 2017), in which BIM and GIS are brought together to help with the assessment and evaluation of potential environmental hazards in construction sites for their geographical location (Adachi, 2003); (Irizarry et al., 2013).

- **Category 2:** Full-lifecycle-based applications. A BIM-GIS integrated and unified model is designed throughout the life cycle. In this category, we could find applications related to managing MEP (Mechanical, Electrical, and Plumbing) assets within big construction projects such as airports, hospitals, and universities (Hor and Sohn, 2021; Cromley and McLafferty, 2012; Karan et al., 2014).

2.6 BIM-GIS Semantic Integration Challenges

Integrating building information modelling (BIM) and geographic information system (GIS) has increasingly become a fast-developing trend and a domain of high interest for both academic research and industrial practice (Jena, 2011; Bakis and Pal, 2007). The rich semantic and geometric information from BIM models from the building life cycle and the power of GIS geo-visualization, well-established decision support system (Singh et al. 2011), and geospatial modelling motivate the integration of these two domains and have become a fertile research field. However, to date, BIM-GIS integration-focused techniques and methods lack a solid theory and methodologies. These can create a rich information model (Hor et al., 2018; Hor and Sohn, 2021), considering the dataset and layered structure in BIM and GIS systems and the attribute types and relationships between different and similar objects from BIM and GIS.

This chapter has presented an overview of the BIM and GIS domains and explored previous and current integration and interoperability efforts over the last years (Bakis et al., 2007; Goedert and Meadati, 2013; Van Deursen, 2010). It is essential to indicate that data integration algorithms, information exchange, and technology development have played a significant role in these integration efforts (Horrocks et al., 2003; Karan et al., 2014). We have discussed the semantic web and ontology engineering concepts and methods as a new approach to achieving a complete BIM-GIS integrated model (Hor and Sohn, 2021; Teicholz, 2013). It offers a 3D integration that can be used for seamless visualization applications with enriched datasets, enhancing many critical activities in various fields of applications ranging from asset management to construction site selection (Hijazi et al., 2011; Ma and Ren, 2017; Andrews, 2020).

Although a BIM-GIS semantic integration shows much potential, translating IFC local and CityGML global geometries to a common ontological model for full integration remains a significant challenge with many questions (Karan et al., 2014). The information loss and schema mismatching resulting from misinterpretation of ontological rules are still substantial concerns, especially during the data parsing phase, because:

- IFC lacks conformity, connectivity, and competitiveness standards.
- The integration method depends on the application field of application.
- BIM and GIS life cycle development differ.

This study intends to explore and investigate a new BIM-GIS integration methodology based on RDF graph databases and semantic web services (Hor and Sohn, 2021). The new platform will provide a unified ontological model that consumes both BIM and GIS into a single robust, stable, and unique model (Hadi et al., 2016; Hor et al., 2018; Hor and Sohn, 2021). The methods proposed in the following chapters are up-and-coming for this goal, given that:

- Ability to work with complex and highly connected BIM and GIS datasets
- Deals with BIM and GIS datatypes are extracted from different levels of details and developments from GIS and BIM data models.
- Flexibility in using data models behind graphs
- Integrated model scalability through sharding (or graph partitioning) used for proof of concepts, and what-if scenarios
- Transaction optimization and query processing built-in tools in graph databases provide data warehousing, real-time batch mode analytics, data discovery, and reporting.

Besides these benefits, a semantic web-based model using RDF graphs provides focused yet generalized BIM-GIS integration and open and collaborative solutions to past and newly emerging problems, such as in a smart city, smart utility grid, and digital twin platforms (Hor and Sohn, 2021).

Chapter 3

Methodology and System Overview

3.1 Introduction

This chapter will discuss the methodology for integrating Building Information Modelling (BIM) and Geographic Information system (GIS) models using semantic web framework and graph data. The process of integration consists of five phases. First, the BIM and GIS ontologies are designed and developed in compliance with semantic web and ontology engineering principles. Therefore, the industry foundation classes (IFC) and Geographic Markup Languages elements are translated to resource description framework objects, including attributes, relationships, and geometrical information. This process will provide transparency into the hierarchical structure representing BIM and GIS informational models. Second, semantic alignment tasks and the matching process was investigated and evaluated to extract appropriate graph nodes, relationships, and properties based on pre-established ontology rules and criterion based on definition similarities between IFC RDF classes and RDF CityGML elements at different object structure, whether at the node, or the relationships or the properties. This phase will provide the selection and grouping of semantic objects (along with their attributes) to create the unique BIM-GIS ontological model with only the objects satisfying the semantic similarity process resulting from the alignment and matching algorithm filtering. The third phase will extract a BIM-GIS integrated model with new objects representing one BIM-GIS model. This is followed by semantic query filtering, where only data relevant to the target domain application is kept within the unified, integrated model. The fourth phase involves designing and creating a translation workflow pipeline to import the RDF graph model from the previous phase and export it into a graph database system with analytics and data mining capabilities. This will provide a complete data framework with schema and data manipulation tools used to extract metadata and help analyze the topological and geometrical information of the integrated model. Therefore, it will be possible to look deep into the model and analyze graph patterns from nodes' interrelationships, properties, etc. Finally, the fifth phase will evaluate and optimize the BIM-GIS semantic integrated model performance. RDF-SPARQL and Cypher Neo4j database queries were designed and developed to build a new benchmark with metrics to help assess the best integration strategy.

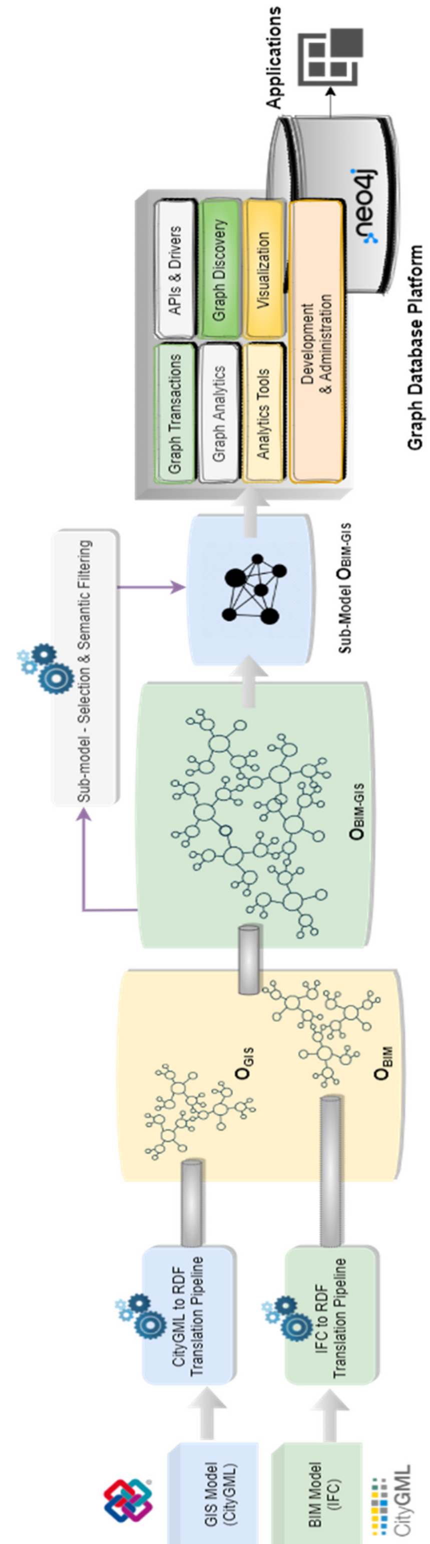


Figure 3.1 Development of the BIM-GIS integrated RDF graph-based model

This research study aims primarily to bring BIM and GIS together into a comprehensive semantic model by bridging the gap between these two domains at the semantic level by using semantic web theory (Semantic Web - W3C) of linked data, vocabularies and inferences to build a new generation of vertical applications for Architecture, Engineering and Construction (AEC) with geospatial contexts. The following sections will detail all the phases starting from object ontology-conceptualizations to the final BIM-GIS integrated model.

3.2 BIM, GIS, and Semantic Web Framework.

In a nutshell, the approach in this research is based on three components:

- RDF graph theory and Semantic web ontologies.
- Building Information Modelling (BIM).
- Geographic Information Systems (GIS).

The RDF graphs and semantic web technology are conceived and designed as the common unifying platform between the BIM and GIS models. The data from IFC and CityGML is presented in formats and structures suitable for semantic integration and to support translation workflows from native models. A description of each of these key concepts is provided below.

3.2.1 RDF Graph and Semantic Web

The semantic web (also referred to as the web of data) consists of a framework technology used to browse, represent, and publish data on World Wide Web. It consists of four main elements:

- **URIs** - (Uniform Resource Identifiers) to uniquely identify semantic objects.
- **RDF** – to represent data as a graph.
- **OWL** - (Web Ontology Language) for conceptual schema representation.
- **SPARQL**, a SQL-type language for RDF graph queries.

At its core, the semantic web defines ontologies of concepts and relationships describing a subject matter or area of knowledge and provides vocabulary formalism using a shared language. Ontology constructors are used to creating a class expression of objects and data, instances of data, constraints, and rules governing the linkage of the objects, relationships, and subjects. This set of three entities is also called a semantic or RDF triple (in most literature, it is referred to as a simple triple). It represents the atomic entity in the RDF data model. Thus, depending on the interpretation of the information representing the model, different representations of a triple are shown in figure 3.2 to figure 3.4 below.

Figure 3.2 RDF *(s,p,o)* Triple

Figure 3.3 RDF *(s,p,v)* Triple

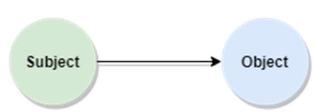


Figure 3.4 RDF *(s,p,o)* triple

In most RDF Graphs, the triples represent a relationship (predicate) between the subject and object and are referred to as statements (in some other literature assertions) of relationships. The triples with shared subjects or objects can be merged into one connected component in semantic data integration. To integrate different fields using RDF graphs, matching triples based on their similar semantic definitions, geometries, and relationships, as presented in Figure 3.5.

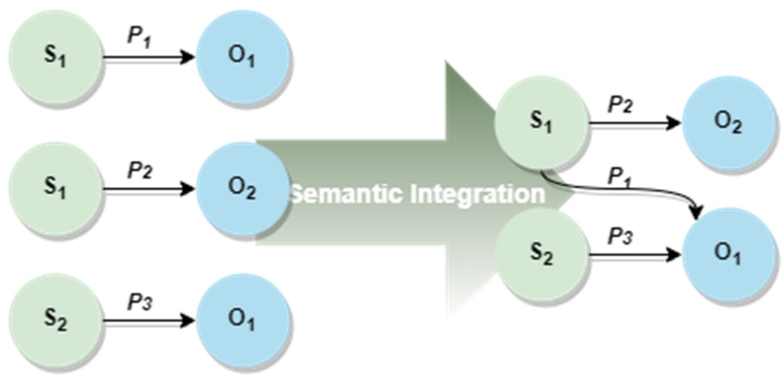


Figure 3.5 Integration and merging using RDF graphs

In the semantic web stack framework, the Resource Description Framework Schema (RDFS - Semantic Web Standards (w3.org)) (RDFs) provides resource classification and abstraction mechanisms to range domain class specification (Hor et al., 2018; Park et al., 2013; Vos et al., 2011). The Web Ontology Language (OWL) (OWL - Semantic Web Standards (w3.org)) are the language used to define semantics (Bakis and Pal, 2009a). The mathematical basis constructor's rationale defining the semantics is based on the statements' inferences, given that concepts in RDFs and OWL ontologies are expressed formally in computer-readable formats (Hayes and Gutierrez, 2004).

3.2.2 BIM and IFC

Building Information Modelling (BIM) uses Industry Foundation Classes (Industry Foundation Classes (IFC) - buildingSMART Technical) standard data schema to manage and exchange building models in Architecture, Engineering, and Construction industries. IFC was designed and developed by BuildingSMART (buildingSMART - The International Home of BIM). It is defined by neutral, open specification EXPRESS-based (IFC Schema Specifications - buildingSMART Technical) for an entity-relationship hierarchical, object-oriented model containing hundreds of entities. The entities in an IFC model can be a basic construct like IfcCartesianPoint, a geometry like IfcExtrudedAearSolid, or an element such as IfcWall (Hor et al., 2018; Bakis and Pal, 2007; Karan and Ardeshir, 2008). The IFC file can be used and exchanged in many formats (IFC Formats - buildingSMART Technical); the following are the most widely used:

- The standard for the Exchange of Data product (STEP): this is the physical format (.ifc) and is primarily used in practice (figure 3.6).
- IFC-Extensible Markup Language (ifcXML): Mostly used because of its enhanced readability by many software and application tools (Berners-Lee, 2006; Karan and Irizarry, 2014).
- Resource Description Framework-XML and terse RDF Triple Language (Turtle): these are based on Web Ontologies language (OWL) (Karan and Ardeshir, 2008; Vos et al., 2011).

Header

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('ViewDefinition [CoordinationView_V2.0, QuantityTakeOffAddOnView]'),'2;1');
FILE_NAME('Central-430-ARCH.ifc','2015-10-28T16:14:39',(''),(''),'Autodesk Revit 2013','20121221_2030(x64)','');
FILE_SCHEMA(('IFC2X3'));
ENDSEC;

DATA;

......................
......................
......................
```

Quantities

```
#267= IFCPROPERTYSINGLEVALUE('Reference',$,IFCIDENTIFIER('TRANSPORTATION & MATERIALS RESEARCH LAB 125A'),$);
#268= IFCPROPERTYSINGLEVALUE('CeilingCovering',$,IFCLABEL('APC-2/EXP/PT'),$);
#269= IFCPROPERTYSINGLEVALUE('WallCovering',$,IFCLABEL('PT'),$);
#270= IFCPROPERTYSINGLEVALUE('FloorCovering',$,IFCLABEL('VT-1'),$);

......................
......................
......................
```

Properties

```
#1041= IFCRELDEFINESBYPROPERTIES('2TMuuB0nz8wRyLVyH90gVq',#50,$,$,(#1037),#1039);
#1048= IFCRELDEFINESBYPROPERTIES('2yWZ2a_ibEGgq5mOfq3M_k',#50,$,$,(#1037),#1046);
#1056= IFCRELDEFINESBYPROPERTIES('1lMuhaAqHDTfr9MVdV7gdn',#50,$,$,(#1037),#1054);
#1076= IFCRELDEFINESBYPROPERTIES('1N0dSFU5P0qenkOPuAjqI5',#50,$,$,(#1072),#1074);
#1083= IFCRELDEFINESBYPROPERTIES('2duNcFn2T3lepP2YZio2YJ',#50,$,$,(#1072),#1081);
#1091= IFCRELDEFINESBYPROPERTIES('0Y55LKcVX9OAA75hOLDacn',#50,$,$,(#1072),#1089);

......................
......................
......................
```

Classification

```
#1898= IFCPROPERTYSINGLEVALUE('Reference',$,IFCIDENTIFIER('CAFE / SEATING 104'),$);
#1899= IFCPROPERTYSINGLEVALUE('CeilingCovering',$,IFCLABEL('ACT-6/EXP/GYP-1/PT'),$);
#1900= IFCPROPERTYSINGLEVALUE('WallCovering',$,IFCLABEL('PT/BPG-4/P.LAM-1/MTL-1'),$);
#1901= IFCPROPERTYSINGLEVALUE('FloorCovering',$,IFCLABEL('PCT-1'),$);

......................
......................
......................
```

Geometry

```
#5746787= IFCCARTESIANPOINT((-38601.2573289096,68279.7196146542,2745.));
#5746789= IFCAXIS2PLACEMENT3D(#5746787,$,$);
#8146288= IFCRELCONNECTSPATHELEMENTS('29FczoNvX7WgJIvLuBtXjT',#50,$,$,$,#156278,#1122035,(),(),.ATSTART.,.ATPATH.);
#5746791= IFCAXIS2PLACEMENT3D(#3,$,$);
#8146285= IFCRELCONNECTSPATHELEMENTS('3aQ2QbyirEoQbRxqVfXiUx',#50,$,$,$,#156238,#1121995,(),(),.ATEND.,.ATPATH.);
#5746793= IFCCARTESIANPOINT((150.,150.,101.6));
#5746795= IFCCARTESIANPOINT((-150.,150.,101.6));
#5746797= IFCCARTESIANPOINT((-150.,-150.,101.6));
#5746799= IFCCARTESIANPOINT((150.,-150.,101.6));
#5746801= IFCCARTESIANPOINT((300.,300.,0.));
#5746803= IFCCARTESIANPOINT((300.,-300.,0.));
#5746805= IFCCARTESIANPOINT((-300.,-300.,0.));
#5746807= IFCCARTESIANPOINT((-300.,300.,0.));

......................
......................
```

Figure 3.6 IFC file structure by categories

One of the challenges practitioners faces using some of the exchange standards is interoperability between them. For instance, there is no direct way to translate between OWL and EXPRESS and vice versa, making the integration even within the same field of BIM a difficult process. This is because of global versus local names of relations and data matching (Hadi et al., 2016; Hor et al., 2018; Hor and Sohn, 2021). However, this study has used OWL as an intermediate format due to its compatibility with most semantic web and linked data platforms and its expressionlessness, flexibility, and portability (Bakis and Pal, 2007; Fu et al., 2007).

3.2.3 GIS and CityGML

Geographic information datasets are represented in the standard data format of City Geographic Markup Language (CityGML). CityGML is the complete conceptual model for geoinformation representation, storage, and exchange of 3D urban and city models between different systems and applications. CityGML along the way from GML and evolved from the former version till nowadays version CityGML 3.0 (OGC City Geography Markup Language (CityGML) 3.0 Conceptual Model Users Guide (opengeospatial.org)) (Bakis et al., 2007; Niknam and Karshenas., 2013).

The CityGML defines geographic objects' topological, geometrical, and semantical properties. It provides a robust semantic-geometric modelling framework developed using ISO 19100 and Unified Modelling Language (UML) tools. Hence, it has become a powerful and unique data representation platform best suited for 3D real-world objects such as buildings, bridges, roads, tunnels, and so on (Bansal, 2011; Elbeltagi and Dawood, 2011). Table 3.1 shows differently oriented object toolsets used to design IFC, CityGML, and RDF models.

	IFC	CityGML	Semantic Web
Schema	EXPRESS	UML	OWL
Data	STEP	GML3	RDF
Identifier	GUID	ObjectID	RDF
Query		SQL	SPARQL

Table 3.1 Modelling of IFC, CityGML, and Semantic Web

CityGML models define the concepts of the level of details (LoDs) for structured 3D objects, allowing a very efficient 3D object representation for different areas of applications like facility management, urban design, data mining, and simulation for many other purposes (Akicini et al., 2010; Bakis and Pal, 2007). The simple structure, flexibility, and ability of the CityGML to represent geographic information range from a very complex level of details, with multiple scale models, to simple and single scaled models. It is best for the GIS data format for exchanging information between different geospatial platforms and the most suitable for integrating geographic information systems into Building information modelling (Adachi, 2003); (Döllner and Hagedorn, 2007; Isikdag et al., 2008). Given the CityGML and XML compatibility, and from XML to RDF/RDF-XML/OWL data, it will be possible to translate GIS models to Semantic Web Framework (Figure 3.1) formats (Hadi et al., 2016; Hor and Sohn, 2021).

3.3 Integration Phases

Given that there have been efforts to integrate BIM and GIS systems, these two systems are very distinct and different in many aspects of model structures, data types, industry standards, and mainly in expressing their levels of detail. The AEC consortiums have tried to design and create a unified model that guarantees seamless data sharing between BIM and GIS.

The Industry Foundation Classes (IFC) and City Geographic Mark-up Language (CityGML) are the most used comprehensive standards for BIM-GIS data exchange. However, much fundamental information can be lost during and after the integration process between these models when incorporating IFC into the GIS model, extending CityGML to support IFC semantics and geometries, and the possibility of conversion between BIM and GIS (Hadi et al., 2016; Karan and Ardeshir, 2008).

The new BIM-GIS integration method proposed in this study is the **Integrated Geospatial Information Model** (IGIM). It is based on RDF graphs on the semantic web platform, allowing semantic reasoning using ontology engineering concepts and practices (Hadi et al., 2016; Hor and Sohn, 2021). Ontology is a domain framework that provides machines to comprehend, interpret data meaning and identify relationships in the domain within the objects of a model. Therefore, in the context of ontologies, data interoperability at the semantic level relies on a standard set of rules for understanding the meaning of elements and the concepts of data they exchange and share. The ontology facilitates semantic interoperability between different or similar knowledge domains by defining shared conceptualization and communication specifications. In the case of BIM and GIS domains, it adds a semantic layer over a synthetic data layer to enhance reasoning for object matching and merging. Thus, an RDF/OWL model is built for the integration model using a series of workflows to transform building information IFC and CityGML objects from traditional format to RDF graphs. Since IFC is presented in the EXPRESS schema, the transformation from IFC ontology (O_{BIM}) into IFC-RDF instances is critical. It must be designed and executed accurately using semantic methods, the same case for CityGML data, stored in a database that needs to be transformed into GIS-RDF with GIS ontologies (O_{GIS}).

In summary, the transformation workflows will consist essentially of the following:

- **Constructing O_{BIM}**: Build an IFC ontology-compliant (**O_{BIM}**) model accurately representing the BIM model's hierarchy structure with all existing objects, relationships, and properties.
- **Constructing O_{GIS}**: Construct CityGML ontology compliant (**O_{GIS}**) representing all GIS objects with spatial and non-spatial attributes and any geographic area surrounding the building.
- **Ontology Mapping**: This process is related to connecting and linking ontological concepts and relationships from O_{BIM} and O_{GIS}. The linked object must share a definition, a quality, a function, or properties. The new ontology model (RDF graph) will contain concepts and relationships ontologically related and defined by an (O_{BIM}, O_{GIS}) to constitute a Data Ontological oriented model (**DOOM**) of the IGIM.
- **Querying $O_{BIM\text{-}GIS}$**: Based on the application domain, filtered data from the DOOM model will be used as source datasets for Application Oriented Ontological Model (**AOOM**) for a particular application using IGIM as a backend. The data will be presented as Web Service Modelling Ontology (WSMO), inheriting the Universal Resource Identifier (URI) concept. It supports W3C web technologies standards, including XML. It is ontology-based and uses strict decoupling, so each resource is specified independently to comply with the distributed and open nature of the web.
- **Data loading**: import/export data onto $O_{GIS\text{-}BIM}$ to form an ontological integrated

information model with a unified graph model from BIM and GIS.

The IGIM Integrated ontology model $O_{GIS\text{-}BIM}$ will consist of all the classes and elements, including attributes, values, and properties from BIM and GIS domains combined, as illustrated in figure 3.7. The datasets are translated and loaded as a complete RDF graph based on a standard format that any semantic web application can access to use the data.

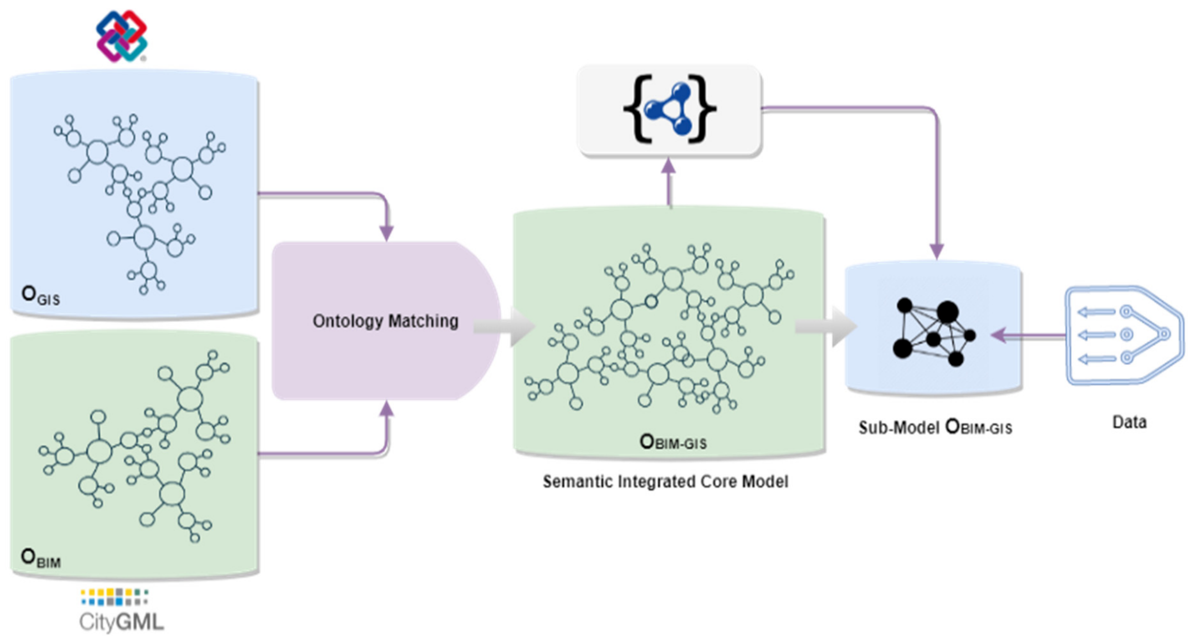


Figure 3.7 Merging O_{BIM} and O_{GIS} Ontologies

Ontology matching is a set of processes defining semantic correspondences between O_{BIM} and O_{GIS} RDF-triples (figure 3.8). Object relationships are semantically defined by matching objects, equivalency, or similar functional properties or by inheriting or enhancing properties of objects from each ontology. This information is then transformed into a semantic web format. Figure 3.8 illustrates the classification rules for matching and collecting triples from GIS and BIM ontologies, where these semantic rules define the process:

- ***Equivalent***: Object A from one domain (BIM or GIS) is Equivalent to Object B in the other domain (BIM or GIS).
- ***As-is***: Object A is taken into the resulting ontology as-is to avoid redundancy or because of the semantic richness of this object (either from BIM or GIS).
- ***Has an attribute or property***: Object A (from either BIM or GIS) has an attribute needed by the object (from BIM or GIS).

Important note: *depending on application fields such as utilities and assets management using digital twin, we can add a new rule based on a similar functional model between objects.*

Given that ontologies are represented differently because of vocabulary definitions and inference capability brought by ontology languages. We will need to outline the classification as well:

- ***Discarding***: redundant or semantically worthless triples from either BIM or GIS.
- ***Merging***: Equivalent or same triples in the same RDF graph.
- ***Inference***: Adding triples to the RDF graph with inference rules to support structural comparison, e.g., a transitive statement using *inverseOf* and *rdfs: a domain* that can be added to one unique triple

The Graph Matching for Ontologies and classification rules will be introduced in more detail in chapter 4.

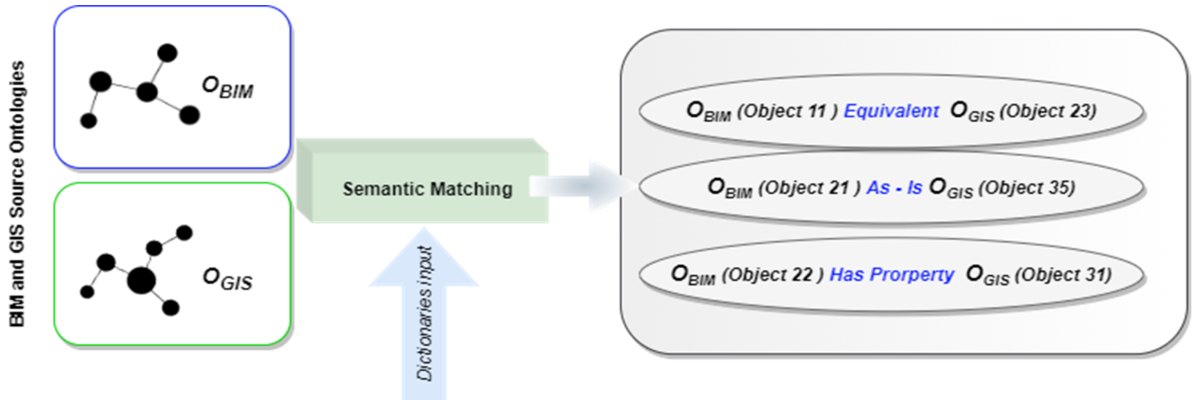


Figure 3.8 O_{BIM}, O_{GIS} classification rules

The ontological integrated information conceptual model and data translation flows are presented in figure 3.9 and consist of three tiers:

- ***Model Inputs Module consists of the following***:

  Building Information Modelling model represented by IFC along with Geographic Information System Model represented by CityGML

- ***Processing and Validation Module:***

  Parsed data from IFC and CityGML into RDF OWL format, and then the Ontology data model will be validated.

- ***Configuring and Application Module:***

  Bridging RDF-OWL from source models into an integrated Geospatial information model and data loading. The SPARQL query language identifies and selects a set of triples (subject, predicate, and objects) based on the application requirements.

3.4 Integration Processes

The BIM-GIS integration process (fig. 3.9) is based primarily on a semantic web framework and RDF graph modelling. Therefore, IFC ontology is designed and developed to describe the hierarchical structure of the BIM objects, with their associated properties and relationships. In parallel, GIS ontology is also developed by converting the data to CityGML format (if required) and then translating CityGML data into RDF format.

In the next step, ontology matching is accomplished using Graph Matching for Ontologies (GMO) between BIM and GIS semantic objects. In the process, BIM and GIS ontologies are transferred into RDF bipartite graphs and used to perform a **S**emantic Alignment Technique of Ontology Graph Matching Algorithm (SAT-GMO) to compare the data, schema, and structural similarities between BIM and GIS ontologies (details in chapter 4). SAT-GMO uses an ontology similarity matrix with pre-assigned initial values. The iteration proceeds until 1:1 matching between objects are achieved in the similarity matrix once the RDF model is formalized and BIM and GIS ontologies are merged at this step. Finally, data filtering is used to manipulate and retrieve RDF data. The result of a query can be represented as XML, RDF, and CSV. The output can also be produced in semantic web-supported standards, like Turtle, N-TRIPLE, and JSON-LD. The complete integrated BIM-GIS RDF graph will be stored and analyzed using a graph-based database model (discussed in chapter 5). The IGIM integration phases are described below.

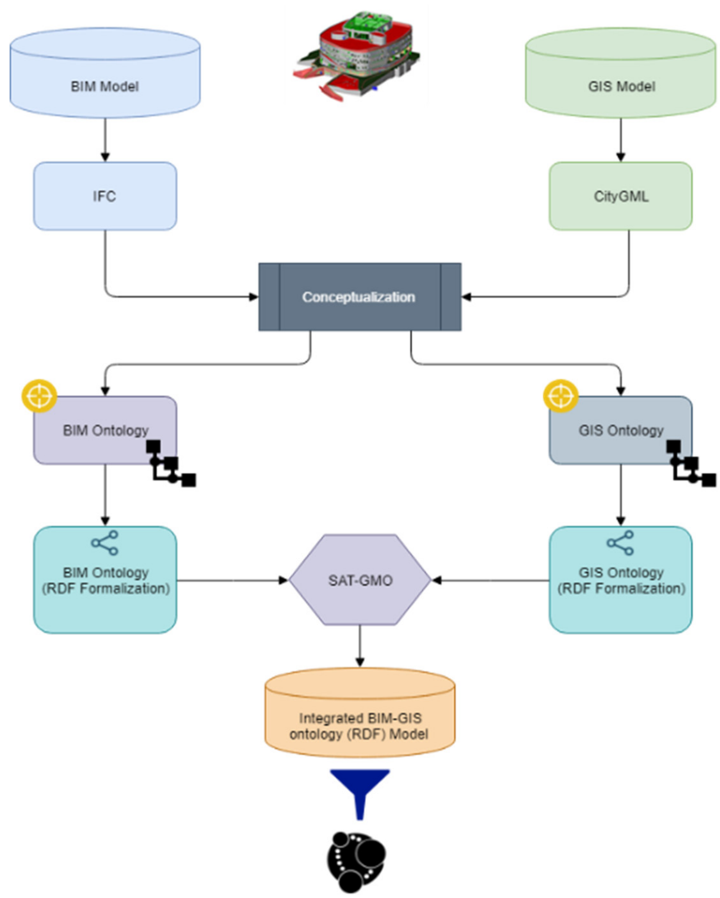


Figure 3.9 Integration Process

Phase 1: Conceptualisation of Objects and Relations from BIM and GIS

The object conceptualization is based on application ontology to represent semantic concepts from Geospatial and AEC domains. The process provides methods to examine and construct integration between GIS and BIM domains based on the object's meaning, representing an object-oriented model of concepts specifying properties relevant to each domain, such as relationships, instances, and values (Hadi et al., 2016; Hor and Sohn, 2021). The model is essentially based on two types of concepts (Karan and Ardeshir, 2008; Karan et al., 2014):

- **Primitive concepts:** represent the entity domain's natural classes where only the necessary conditions are specified, and their definitions can describe them. These concepts also correspond to the top of the hierarchical structure of the ontology representation.
- **Defined concepts**: representing subclasses of the primitive object to mimic the ENTITY construct organized into taxonomy via the Supertype/Subtype partial ordering relation as shown in figure 3.10 for IFC elements.

a. Conceptualizing Building Information Modelling

To illustrate the conceptualization of building information modelling, we can examine the *IfcWindow* entity in EXPRESS specification format (fig. 3.10).

```
ENTITY IfcWindow
 SUPERTYPE OF (ONEOF
    (IfcWindowStandardCase))
 SUBTYPE OF (IfcBuildingElement);
    OverallHeight : OPTIONAL IfcPositiveLengthMeasure;
    OverallWidth : OPTIONAL IfcPositiveLengthMeasure;
    PredefinedType : OPTIONAL IfcWindowTypeEnum;
    PartitioningType : OPTIONAL IfcWindowTypePartitioningEnum;
    UserDefinedPartitioningType : OPTIONAL IfcLabel;
 WHERE
    CorrectStyleAssigned : (SIZEOF(IsTypedBy) = 0)
                           OR ('IFC4.IFCWINDOWTYPE' IN TYPEOF(SELF\IfcObject.IsTypedBy[1].RelatingType));
END_ENTITY;
```

Figure 3.10 Representation of IfcWindow entity (EXPRESS Class)

The *IfcWindow* is defined by ENTITY and stands for an abstract super-class of mutually disjoint classes. The construct ONEOF specifies the existing disjoint relation. On the other hand, the relationship SUBTYPE OF states that *ifcWindow* is subsumed under *ifcObject*. *OverallHeight*, *OverallWidth*, *PredefinedType*, *PartitioningType,* and *UserDefinedPartioningType* are attributes, and the rule specifies a certain condition associated with the class.

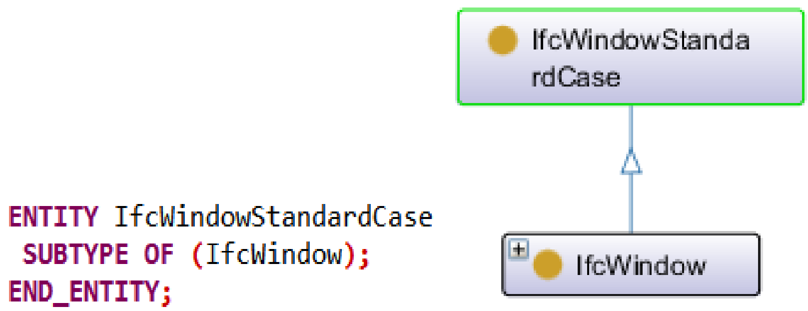


Figure 3.11 *ifcWindow – IfcWindowStandardCase*
Supertype/Subtype relationship.

To understand the primitive versus defined concepts, let's consider an individual element Wx (an instance of the primitive concept *IfcWindow).* This Wx will have the properties of a standard window in a BIM model. Therefore, it will automatically have these properties *OverallHeight*, *OverallWidth*, *PredefinedType*, *PartitioningType*, and *UserDefinedPartioningType*. The Wx can be inserted as a BIM opening, and its profile represents a rectangle within the 2D plane of the opening from IfcWindowStandardCase. We define this IFC entity as the Defined Concept, so any associated properties of the *IfcWindowStandardCase* (Fig. 3.10) are necessary and enough.

This process of conceptualizing described above using BIM IFC objects (based on their definition from EXPRESS schema) shown in figure 3.11 is used for their counterpart elements in GIS CityGML elements. As an illustration, we used OWL ontology schema and exchange file format for the building information model. BIM data are extracted to enhance the integrated semantic object in the ontology model using other properties, attributes, and values, as illustrated in figures 3.12 and 3.13.

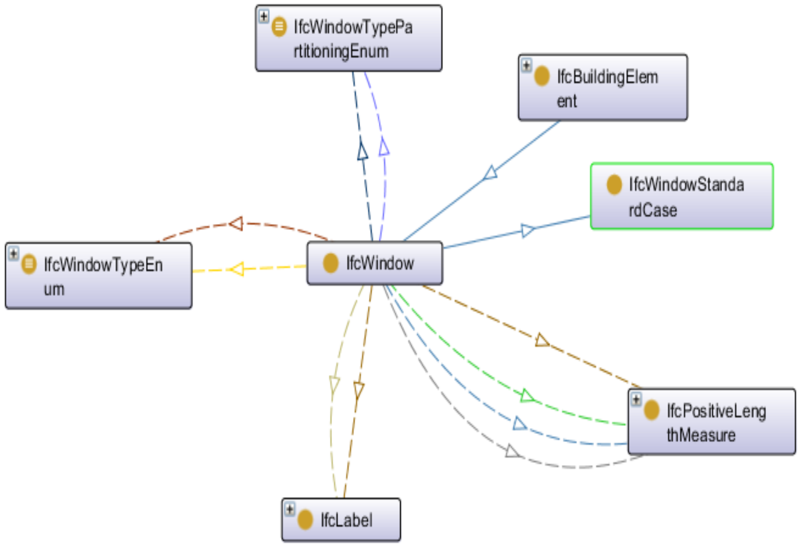


Figure 3.12 Inter-relationships between IFC entities

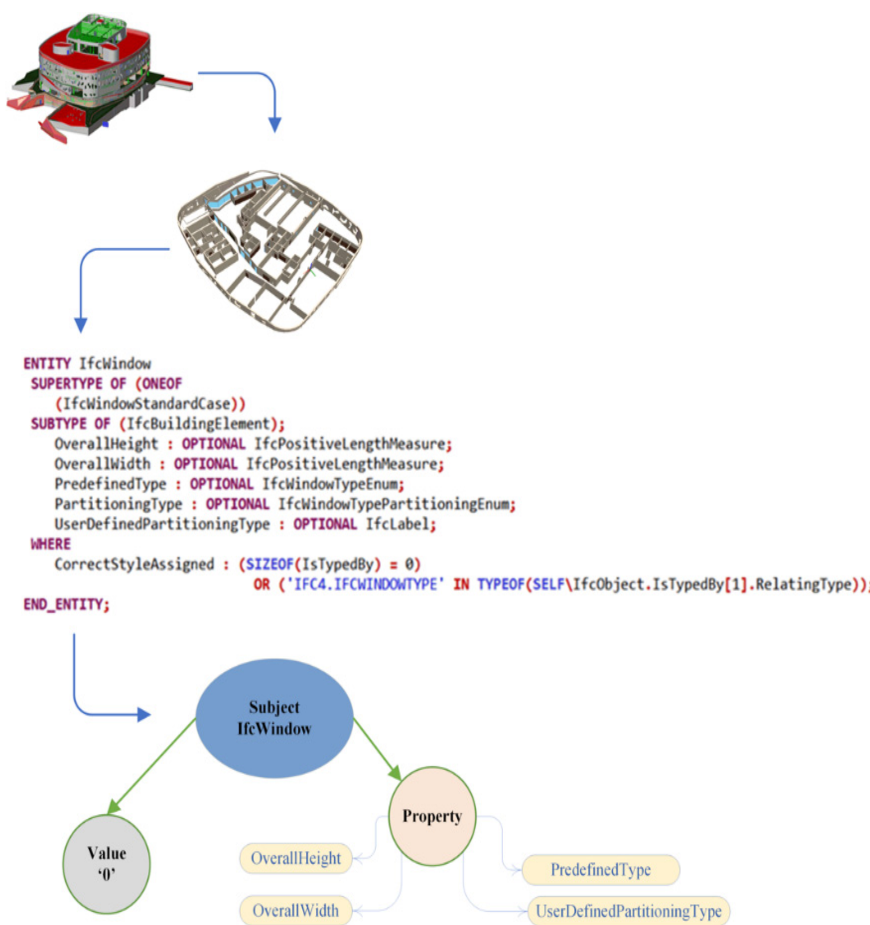


Figure 3.13 IfcWindow EXPRESS entity parameters and corresponding OWL components

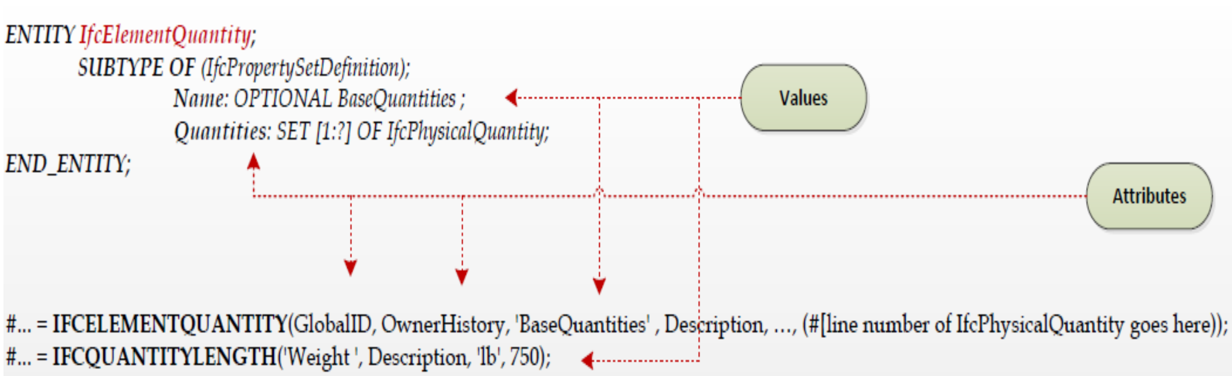


Figure 3.14 Semantic transformations of IFC Element

b. BIM Georeferencing

BIM geo-referencing plays a crucial role in BIM-GIS semantic integration. It is one of the most powerful methods to add a contextual view of any architecture, engineering, and construction (AEC) project. Since this research aims to develop a BIM-GIS integrated model using an RDF graph database, it is essential to ensure that georeferencing is implemented within the BIM during our data preparation. Many tools are available in BIM software products (Autodesk Revit 2016) that support the accurate definition of geolocation information for the BIM and export them as industry foundation classes (IFC). RDF data format using IFC to RDF packages developed during this project.

Georeferencing is the process of coordinate transformation that takes the coordinates from one system to another (Zhu and al. 2021). The changes will refer to the association of geospatial data, such as feature datasets or raster imagery, with projected or geographic CRS (Coordinate Reference System). Georeferencing refers to the exact location of the building using a coordinate reference system as defined by (Hill); georeferencing includes two steps (Zhu and al. 2021) and (Diakite and all. 2020):

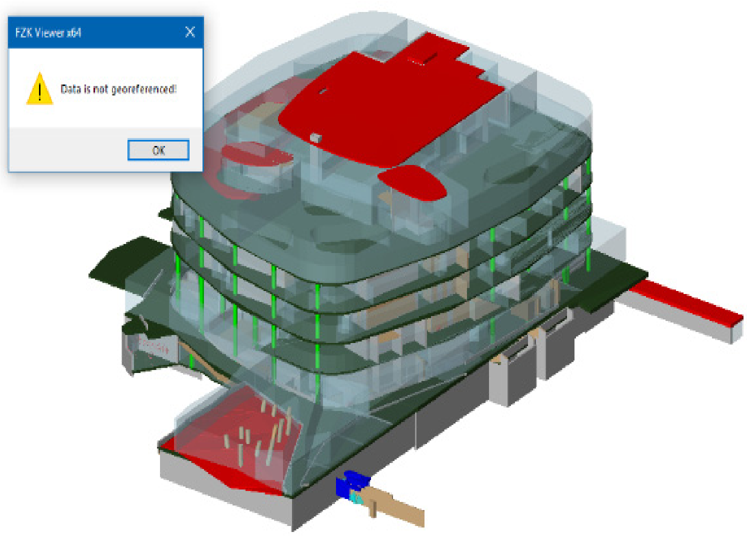
- Set up the spatial reference.
- Get coordinate transformation parameters.

The first step matters the most, as it is the premise of the second step, while the second step has been well investigated by the geospatial industry (Hill and al. 2009, Zhu and al 2021). The BIM differs from traditional GIS datasets, such as vectors and images. For traditional datasets, the spatial reference is established by selecting GCPs, in same same way as GIS systems, but BIM models have an additional way. The application is decided based on embedding the spatial reference information within them (Zhu and al. 2021) and (Diakite and all. 2020). From the established spatial reference, transformation parameters can be derived for coordinate transformation. Therefore, the geo-referencing capability of the IFC refers to its capability to accommodate spatial reference information.

In the past studies, there were many studies on the geo-referencing of BIM models. However, only a few of them focused on the systematic investigation into the geo-referencing capability of IFC, such as the study by Uggla et al. 2018, the study by Clemen and al. 2019, and the study by Arroyo et al. 2018.

The IFC standards offer adequate classes for describing the information for the georeferencing BIM in question. However, practices have demonstrated that these classes lack information or are absent during the Revit project design. Since IFC is not a native format to an existing BIM software, the IFC classes file from Revit export schemes, which allow exchange and interoperability between them (Zhu and al. 2021). If there is lost or missing georeferencing information in the IFC file, it either means that the designer did not provide it at the time of the export or the software does not support its export.

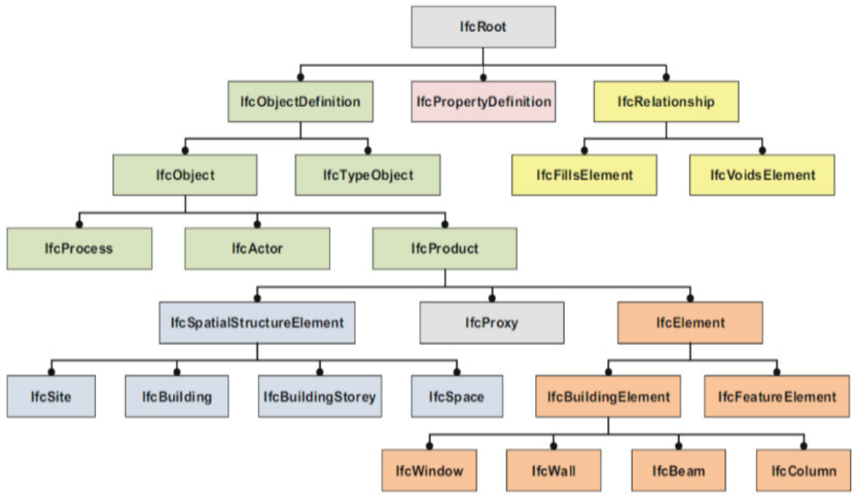
This research project investigated the tools that could offer proper georeferencing for exported IFC files (Hadi et al., 2016; Hor and Sohn, 2021). We have also looked at new GIS products, such as Esri ArcGIS Pro, that can natively deal with this important information to bridge these two domains, BIM and GIS. Once the georeferencing information (figure 3.14) is added and parsed into RDF graphs, Cipher query objects can be extracted using the location information when the BIM-GIS graph model is visualized on 3D applications like Cesium or JavaScript-based web interfaces.



AddressLines	PostalBox	Town	Region	PostalCode	Country
11 ARBORETUM LANE, TORONTO, ON M3J 1P3	?		43.7723007202148		-79.5066375732422

Figure 3.15 BIM georeferenced information from the IFC file

In this study, one of the IFC files (York University Bergeron Building) was not provided with any geographic information associated with it, whether it was coordinates or projection. This was the case with the other BIM models. The location of BIM models is stored in the *IFCSite* element, which should have at least the latitude and longitude information of a reference point in the building of its surrounding. Inspecting the IFC file reveals several inconsistencies in naming and grouping elements. For instance, some of the building elements of the *ifcBuildingElementProxy* element are organized the same as *IfcBuildingElement* without a defined meaning of the particular type of building element it represents. There were many elements (fig. 3.15) with unclear definitions.




```
ENTITY IfcSite;
    ENTITY IfcRoot;
        GlobalId                    : IfcGloballyUniqueId;
        OwnerHistory                : IfcOwnerHistory;
        Name                        : OPTIONAL IfcLabel;
        Description                 : OPTIONAL IfcText;
    ENTITY IfcObjectDefinition;
    INVERSE
        HasAssignments              : SET OF IfcRelAssigns FOR RelatedObjects;
        IsDecomposedBy              : SET OF IfcRelDecomposes FOR RelatingObject;
        Decomposes                  : SET [0:1] OF IfcRelDecomposes FOR RelatedObjects;
        HasAssociations             : SET OF IfcRelAssociates FOR RelatedObjects;
    ENTITY IfcObject;
        ObjectType                  : OPTIONAL IfcLabel;
    INVERSE
        IsDefinedBy                 : SET OF IfcRelDefines FOR RelatedObjects;
    ENTITY IfcProduct;
        ObjectPlacement             : OPTIONAL IfcObjectPlacement;
        Representation              : OPTIONAL IfcProductRepresentation;
    INVERSE
        ReferencedBy                : SET OF IfcRelAssignsToProduct FOR RelatingProduct;
    ENTITY IfcSpatialStructureElement;
        LongName                    : OPTIONAL IfcLabel;
        CompositionType             : IfcElementCompositionEnum;
    INVERSE
        ReferencesElements          : SET OF IfcRelReferencedInSpatialStructure FOR RelatingStructure;
        ServicedBySystems           : SET OF IfcRelServicesBuildings FOR RelatedBuildings;
        ContainsElements            : SET OF IfcRelContainedInSpatialStructure FOR RelatingStructure;
    ENTITY IfcSite;
        RefLatitude                 : OPTIONAL IfcCompoundPlaneAngleMeasure;
        RefLongitude                : OPTIONAL IfcCompoundPlaneAngleMeasure;
        RefElevation                : OPTIONAL IfcLengthMeasure;
        LandTitleNumber             : OPTIONAL IfcLabel;
        SiteAddress                 : OPTIONAL IfcPostalAddress;
END_ENTITY;
```

Figure 3.16 *IFCSite* Inheritance Graph

In theory, the latitude and longitude values in *IfcSite* with an optional offset and true north direction with *IfcGeometricRepresentationContext* information should make it possible to geo-reference the BIM with precision, and IFC files do fill in the requisite values in the *IfcSite* element. However, in most cases, geographic referencing information (latitude/longitude values) is generally set to zero or approximating the actual geographic location. It is highly recommended to have IFC files set to accurate real-world geographic locations, applying latitude/longitude values from the *IfcSite*. This is important in the case of geospatial integration and considering the WorldCoordinateSystem of the *IfcGeometricRepresentationContext* offsets given by the BIM model. Also, if the y-axis of the WorldCoordinateSystem in IfcGeometricRepresentationContext does not match the true north direction, the TrueNorth attribute should also be set.

c. Conceptualizing Geographic Information Systems

The GIS features' semantic terminologies must be defined and then identified to associate them with their equivalent ontological objects to convert the Geospatial model into an RDF format. This step is critical to having a starting ontological model. However, because the GIS features like feature classes, features datasets, topologies, and raster are all organized as tables (relations) managed by relational databases, these tables contain fields organized in tuples (rows) with attributes (schema). This would make each GIS table (Hor and Sohn, 2021; Hijazi et al., 2011) in the geospatial data model correspond to an RDF class represented as rdf:class with all its related attributes and properties particular to that GIS object (Hadi et al., 2016). Each object's properties are used to achieve relationships with other objects in the same model and can create a connection to different instances (like in database instances) figure 3.16. The other constraints in the GIS relational models, such as primary keys, can relate two or more objects through a direct relationship between objects, the same as relationships between GIS features in a GIS system like the ArcGIS platform (Authoritative Data & Maps for Apps (esri.com)), GeoMedia (GIS Mapping Software | GeoMedia | Hexagon Geospatial) or Bentley systems (Integrated 2D/3D GIS and CAD Software Solutions (bentley.com)) (Azhar et al., 2011; Cheng and Yang, 2001).

The definitions of the relationships in the GIS ontological model can also use GIS models methodologies to design highly connected systems. These include telecommunications and utility networks using domains and coded values such as utility pipe diameters, range of values, and flow direction, which can be expressed as the RDF rule in RDF modelling figure 3.16 and figure 3.17

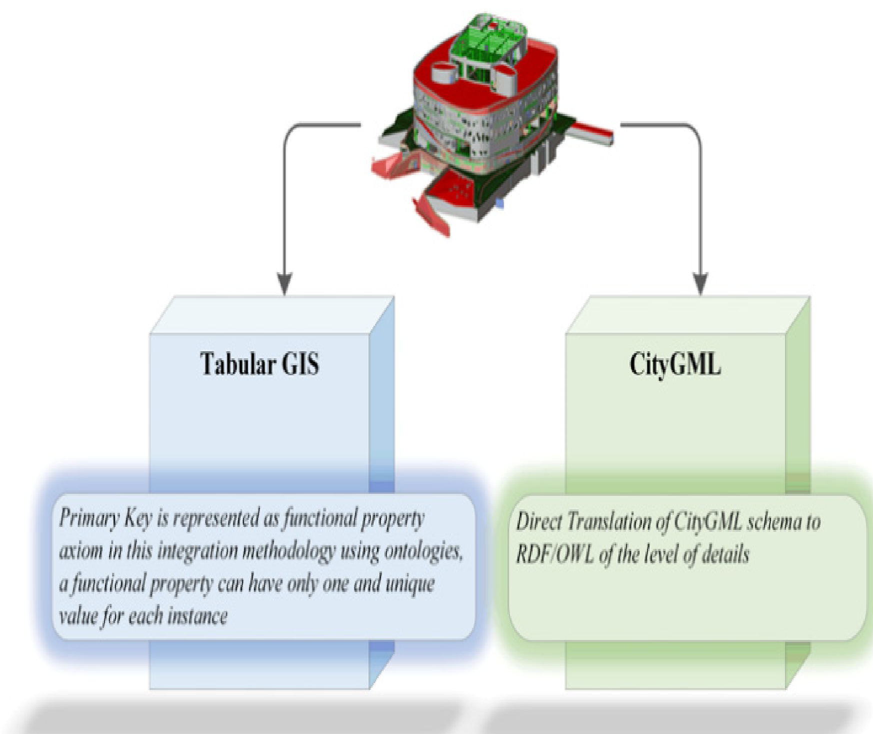


Figure 3.17 GIS representations

The GIS model will be converted from GML data (CityGML) file into an RDF file-based format using semantic Extract, transform, and load (ETL) scripts to complete the transformation process:

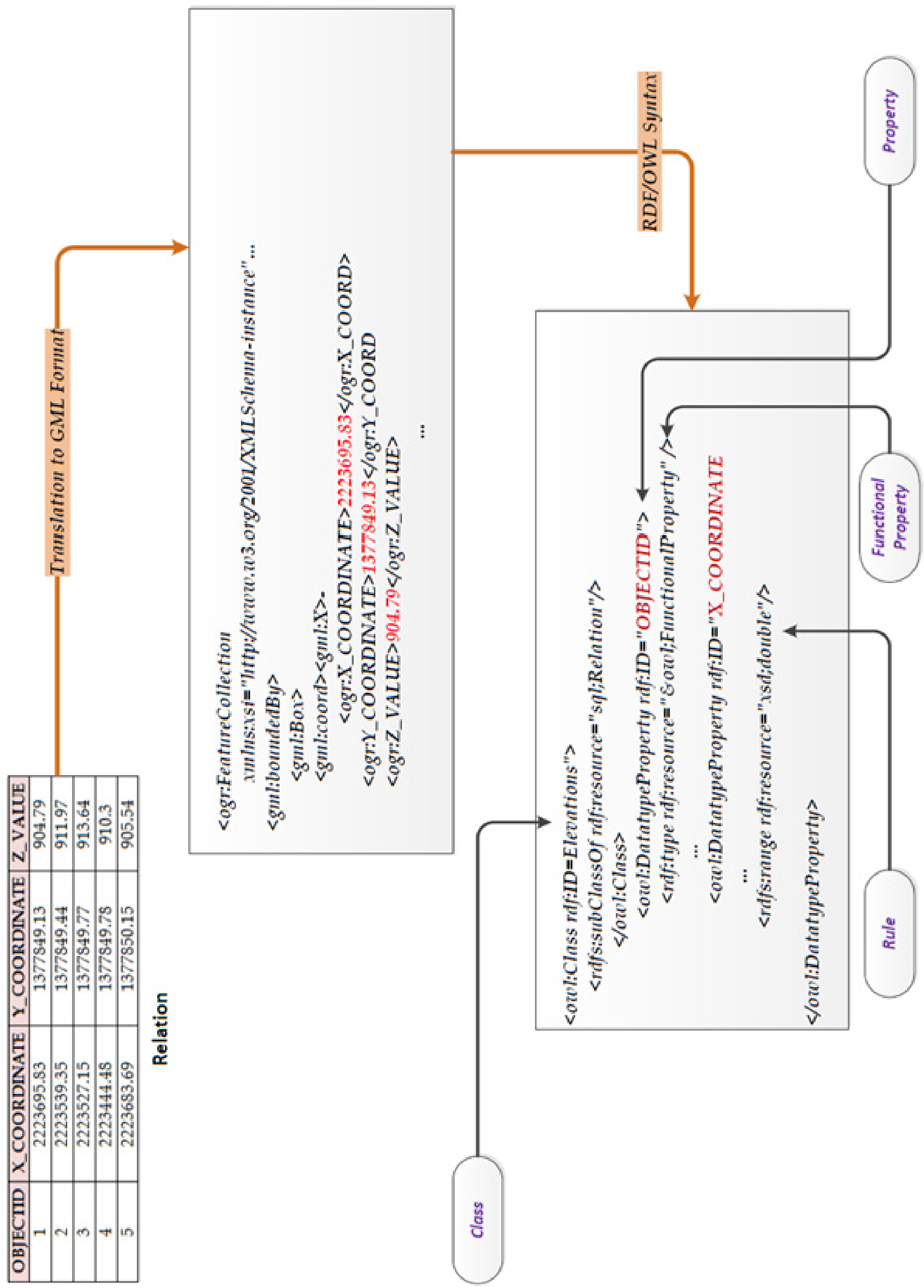


Figure 3.18 GIS relational database to RDF model

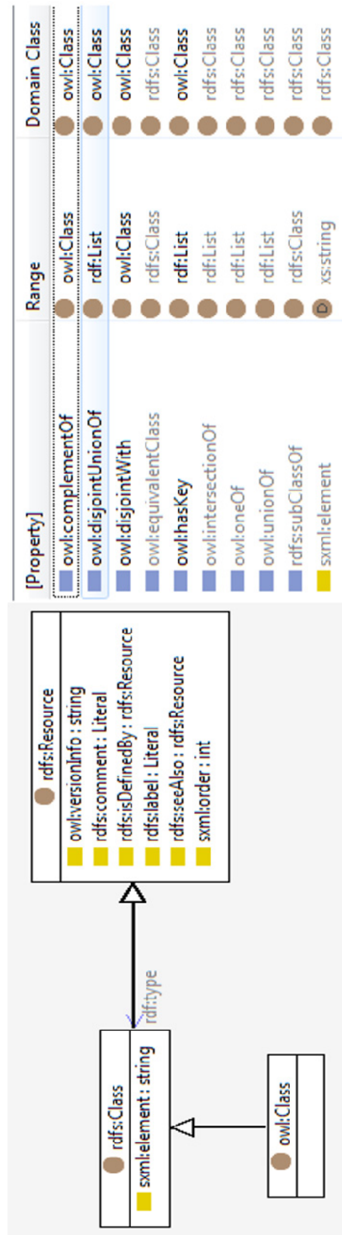


Figure 3.19 Schema and rules from rdf:Class and subclass Owl:Class

Phase 2: Ontology Concept-Mapping and Similarity

Once BIM and GIS data have been translated into RDF semantic data or RDF graphs, the following step creates a bridge between these two source models. To achieve that, we must look closely at the taxonomies of data objects in both models and their structural and syntactical differences (Hadi et al., 2016; Hor and Sohn, 2021). As ontologies have a hierarchical structure where concepts and instances can be arranged in a tree-like structure, using the GMO (Graph Matching for Ontologies) bipartite graph matching algorithm represented in Table 3.2, we can define ontologies and establish the correspondences between them by measuring the structural similarities between graphs (Hor and Sohn, 2021; Karan and Irizarry, 2014). At a high level, the GMO will compare the structures of the entities of interest to quantify the similarity of triples (subject, predicate, and object) between the BIM and GIS models expressed in OWL or RDF format (Goodchild et al., 1999).

Step 1	*Pars O_{BIM} and O_{GIS} and transform them to corresponding RDF bipartite graphs.*
Step 2	*Classify entities in O_{BIM} and O_{GIS} as classes. Properties and instances.*
Step 3	*Coordinate O_{BIM} and O_{GIS} using coordination rules (discarding, merging, inference, and list).*
Step 4	*Determine external entities for O_{BIM} and O_{GIS} and set up an external similarity matrix.*
Step 5	*Setup matrix representation for O_{BIM} and O_{GIS}.*
Step 6	*Initialize the similarity matrices.*
Step 7	*Run Iteration step with updating equation until some predefined convergence precision is reached.*
Step 8	*Find One-to-One object matching using a similarity matrix.*

Table 3.2. Ontology Matching Algorithm for BIM-GIS Ontologies integration

This research presents an efficient and very effective methodology using ontology matching called GMO (Graph Matching for Ontologies) between objects from BIM and GIS models (Akicini et al., 2010; Venugopal, 2010). The method uses the bipartite graph to represent ontologies from BIM and GIS platforms. Then, the objects are aligned based on their ontological characteristics to measure their structural similarity between their respective graphs. Graph Matching for Ontologies (GMO) takes matched pairs of objects found previously by other approaches as external inputs in the matching processes. It outputs additional matching pairs by comparing the structural similarity considering their properties and functional profile (Hor and Sohn, 2021; Isikdag et al., 2008). The matching criteria for the GMO module can be gained by variant approaches available and may have significant variance in size. In our approach, the structural similarity is designed to be independent of lexical similarity, and the effectiveness of GMOs has been tested with a variant-sized input matching (Fu et al., 2007; Karan and Irizarry, 2014). In the following sections of this chapter, we will show how the GMO can be applied to construct a BIM-GIS integrated model (Karan and Irizarry, 2014).

Knowing that the RDF graph model is foundational to the semantic web, it has the nature of a graph structure. OWL ontology can be mapped to an RDF graph. Thus, we adopt the graph structure approach to represent the integrated information model between BIM and GIS platforms to compute the structural similarity between ontology entities with semantic correspondence. Therefore, graph-based techniques are graph algorithms that consider the input ontologies as labelled graphs. The ontologies, including schema and taxonomies, are viewed as labelled graph structures (Hor and Sohn, 2021). The semantic similarity analogy between two nodes from two ontologies will be based on analyzing their positions within the graph. The logic is that if two nodes from two ontologies are similar, their neighbouring nodes (and their properties/relationships) must also be identical or have similar associated properties and relationships. Adding to this level of similarity between entities, the model-based (or semantically grounded) algorithms will handle the input based on semantic interpretation so that two entities are the same such as the IFC class *ifcDoor* class and CityGML object Door. Then they share the same interpretations. Thus, they are well-grounded deductive methods (Horrocks et al., 2003; Teicholz, 2013).

Using SPARQL (fig. 3.19) for RDF and semantic web, queries will interrogate the model to select and extract information based on a target application call/request. The results will contain a set of triples of subjects, predicates, and objects satisfying the request criterion.

```
Select DISTINCT ?subject ?property ?value
WHERE {?subject
          rdfs:subClassOf SpatialStructureElement.
          ?subject
          ?property
           ?value.
```

Figure 3.20 SPARQL typical query

The SPARQL query output will combine GIS and BIM triples in many semantic-based data formats (Hor and Sohn, 2021):

- ***Turtle***: This is the most straightforward human-readable format used by many applications.
- ***RDF/XML***: Consists of RDF format in XML.
- ***RDFa***: Consists of the original RDF file embedded into HTML attributes.
- ***JSON-LD***: Web-based format used in web development.

Recently, the IfcXML file format has been developed mainly for AEC-focused applications. The ifcXML is an XML-based format (.ifcxml as an extension) that was defined by ISO 10303-28, and it is well suitable for data integration and interoperability with tools and exchanging complete and partial building models such as MEP sub-models (Cheng and Yang, 2001; Karan and Ardeshir, 2008).

By designing and developing an integration-based ontology model, there will be no need to create links between various terminologies represented in objects and classes from models. Instead, we will only need to transfer between vocabularies in each standard of these models (Hor and Sohn, 2021).

Phase 3: Conversion and Integration Using RDF Graphs

In the third step of the Integrated Geospatial Information Model (IGIM) model construction, BIM and GIS elements are translated into a formal standard ontology language RDF/XML-OWL.

For flexibility and practicality purposes for end-user applications and data sharing, we used IFC and CityGML as our primary models (Hadi et al., 2016; Hor et al., 2018). The RDF-OWL data model representing the IGIM will comprise relationships and triples from BIM and GIS. The IFC was designed and written in a separate EXPRESS schema by BuildingSmart. Therefore, the IFC needs to be transformed into RDF by exporting an EXPRESS schema into RDF ontology to feed the IGIM, parallel to the process from CityGML to IGIM. An IFC ontology makes it possible to develop IFC-RDF instances from the IFC file used to build the RDF model (Hadi et al., 2016; Karan and Irizarry, 2014). All the GIS data stored in the database will be converted to RDF by annotating data with ontologies defining the properties of the GIS features, such as the interior space, floors, etc. As mentioned in the previous section, the GIS model was built on relational modelling. The semantic web uses graph structures with triples for storing and sharing data. Hence, we must transform the data into Geography Geospatial Language (GML) format (OGC standard). This will help define an extended markup language-based data access language to support the interoperability with GIS packages and result from the data model-based (Hor and Sohn, 2021).

The conceptual data translation pipeline and configuration are represented in the data flow process design diagram in figure 3.20.

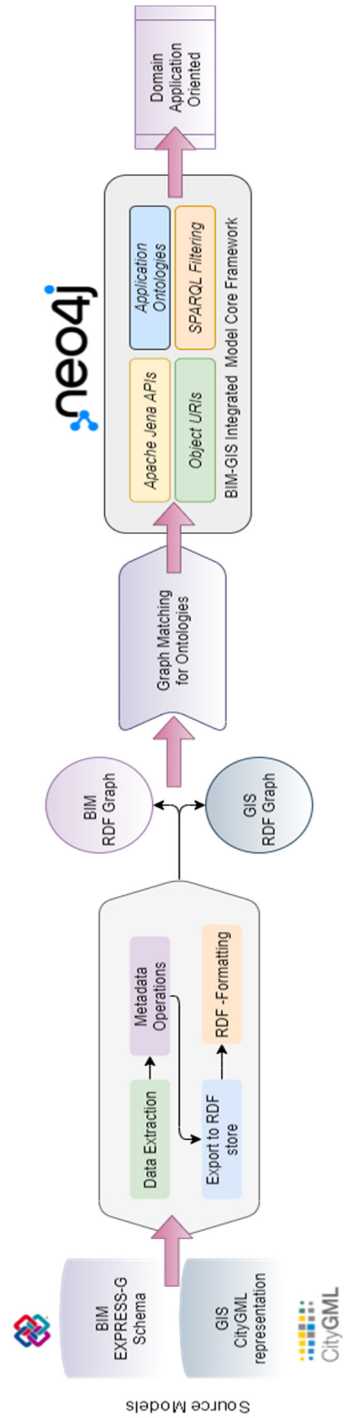


Figure 3.21 Translation pipeline

Figure 3.20 shows an end-to-end translation process using Jena (https://jena.apache.org/). This programming toolkit uses the Java language with rich ontology API and allows operations and tasks against the new integrated model. However, many other great tools provide interactions with RDF data models, such as Python RDFlib (https://github.com/RDFLib/rdflib) package that we also consider during the implementation phases.

Phase 4: Querying RDF Integrated Model

The Semantic querying language SPARQL is the standard language for RDF graphs and extracting information from semantic web models. It provides commands and constructors to retrieve and manipulate data stored in semantic models. The outputs can be presented in many formats supported by the semantic web, like CSV, RDF, or XML, and can combine BIM RDF triples to GIS RDF triples in N-Triples, JSON-LD, or simply as RDF/XML. A typical SPARQL query code is presented in figure 3.21. Using the SPARQL endpoint with SPARQL Query Service APIs (Ergen et al., 2007), data manipulation can be executed to select the sets of datasets streamed on a web platform through SPARQL Protocols. These operations also include resolving unfamiliar URI and vocabularies from BIM or GIS into the integrated RDF models and providing a single, integrated data model with one presentation view for the applications to access (Hadi et al., 2016; Hor et al., 2021).

```
## Project : BIM-GIS Semantic Integration
## Author : Hadi H. (YU)
## Description : Query to retrieve pairs of a window and a
## wall, with the condition that the window is placed in
## the wall but they are not contained in the same storey.

SELECT ?window ?wall
WHERE{ ?window a ifc:IfcWindow .
        ?window schm:isPlacedIn ?wall .
        ?wall a ifc:IfcWall .
FILTER NOT EXISTS {
                  ?wall IGIM:isContainedIn ?storey .
                  ?window IGIM:isContainedIn ?storey .
                  ?storey a ifc:IfcBuildingStorey .
                  }
    }
```

Listing a – SPARQL Query to retrieve pairs of windows and walls with a condition on the wall

```
## Project : BIM-GIS Semantic Integration
## Author : Hadi H. (YU)
## Description : Query to count load bearing walls for each
## building storey
SELECT ?storey (COUNT(?wall) AS ?H)
       WHERE{
             ?wall a ifc:IfcWall .
             ?wall IGIM:loadBearing true .
             ?wall IGIM:isContainedIn ?storey .
             ?storey a ifc:IfcBuildingStorey .
} GROUP BY ?storey
```

Listing b - SPARQL Query to retrieve the count load bearing walls for each building story

```
## Project : BIM-GIS Semantic Integration
## Author : Hadi H. (YU)
## Description : Query to retrieve spaces which have
##               window-to-floor area ratios less than 0.5.
SELECT ?space ?ratio
    WHERE{ ?space a ifc:IfcSpace .
           ?space IGIM:hasSpaceArea ?area .
         {
       SELECT ?space (SUM(?windowArea) AS ?
           totalWindowArea)
       WHERE{ ?space schm:hasSpaceBoundary ?w .
              ?w a ifc:IfcWindow .
              ?w IGIM:hasWindowArea ?windowArea .
       } GROUP BY ?space
       } BIND ((?totalWindowArea/?area) AS ?ratio) FILTER (?ratio<0.5)
}
```

Listing c – SPARQL Query to retrieve spaces which have window-to-floor area ratio less than 0.5

Figure 3.22 Example of SPARQL Queries

Phase 5: Semantic model validation process

In this phase of the project, a proof of concept was designed and developed using Bergeron Building Centre for Engineering Excellence on the campus of York University in Toronto to validate the proposed approach for semantic BIM-GIS integration. A complete 3D GIS combining another BIM (Petrie Science and Engineering Building) and in-campus Road (Wall Road) are also used as source full datasets to model the BIM models and enhancing them with surrounding GIS data. Further, we designed and developed the complete system architecture for the integrated BIM-GIS model (Hadi et al., 2016; Hor and Sohn, 2021).

At the core of the system, we have REST web services used as semantic endpoints pointing to the integrated BIM-GIS integrated model, the graph database management system will be the container of the RDF dataset feeding the application on the intelligent urban mobility web to validate the semantic integration methodology (Hadi et al., 2016, Hadi et al., 2018). The completed system will offer analytics and filtering, making it possible to run queries for data retrieval, advanced analysis and graph data mining algorithms of the BIM-GIS integrated graph model. These queries could range from complex queries like the ones used for facilities management, emergency evacuation, evaluating inventories, cost estimation, and shortest path to the simplest, like finding element information and indoor navigation paths (Hadi et al., 2016; Hor and Sohn, 2018).

3.5 Summary

This chapter introduced the theoretical and conceptual basics of building an integrated data model between BIM and the GIS domain using a semantic web framework and ontology engineering methodologies. The new approach will bring the benefits of BIM and GIS knowledge domains together using their semantic data into one integrated model, the Integrated Geospatial Information Model (IGIM). The Ontology Graph Matching algorithm (GMO) with classification rules and defined taxonomies will offer access to process datasets from GIS and BIM utilizing RDF graphs. There will be no need to establish a one-to-one correspondence between BIM objects and classes. Instead, IFC and CityGML models are transformed, translated then exported into IFC-RDF and GIS-RDF graphs to build a single integrated semantic model.

This chapter also introduced a conceptual data translation pipeline that takes input BIM and GIS models from their native formats, respectively, IFC and CityGML, and presents the translation process to the IGIM output integrated semantic model.

We also presented the integration algorithm and the concept of a similarity matrix to achieve the most accurate model based on ontology rules and object definition. In our next chapter, we look deeply at enhancing the SAT-GMO algorithm by considering more aspects of GIS and BIM domains data to maintain and optimize RDF graph generation and mapping procedure and create evaluation metrics for the RDF-generated graphs.

Chapter 4

Semantic Alignment Techniques for Ontology Graph Mapping Algorithm Evaluation for BIM-GIS Integration

The process of graph matching and semantic alignment is vital for successfully and seamlessly sharing information between BIM and GIS models. Therefore, designing an adequate matching methodology with proper alignment parameters has fundamental to achieving the finest BIM-GIS semantic integration possible. However, the core schema differences and the complexity of these two domains make it a great challenge. Also, most previous matching attempts were often erroneous, time and resources consuming. This chapter dives deeper into multiple ontologies matching algorithms and semantic alignment techniques specifically for BIM and GIS models.

This ontology-based matching methodology uses IFC (BIM standard) and CityGML (3D GIS standard) semantic RDF triples the bipartite graph approach, allowing an accurate matching between these two schemas considering the difference and richness of each. Therefore, based on their semantic models, an evaluation procedure was designed and implemented to investigate the relationships between IFC and City GML RDF graphs. The most relevant results have been achieved using an Association Rule Ontology Matching Approach (AROMA) matching technique. The results are presented and discussed for such integration employing the proposed method on multiple datasets.

4.1 Introduction

The Building Information Modelling (BIM) and Geographic Information System (GIS) domains share a common need for mutual information. The GIS can facilitate BIM applications for construction site layout selection, urban planning, and facilities management. At the same time, rich information from BIM models provides detailed models in GIS for better buildings and urban management (Azhar et al., 2011; Horrocks et al., 2003). However, the exchange and integration of these 3D data models are still challenging, such as semantics definitions, interoperability, representations, data models, accuracy, and interpretation approaches. Since IFC and CityGML are schemas in BIM and GIS domains, they are designed for different purposes (Hor et al., 2018; Agdas and Ellis, 2010). The content and data represented in these two models are structurally different. Therefore, it is hard to achieve a complete matching between them (Hor and Sohn, 2021; Alders, 2006; Anumba et al., 2008.

The methodology used in this study provides a complete and accurate matching between BIM and GIS models. A Resources Description Framework (RDF) graphs concept is applied based on semantic web technologies using measurable similarity metrics and identifying mapping rules. The main goal of ontology matching between BIM and GIS is to retrieve relations between entities expressed in BIM ontology (O_{BIM}) and GIS ontology (O_{GIS}) models. These relations are either expressed "***as equivalent to***," "***as-is***," or "***has-a***" discovered through the similarity alignment between the entities from these two platforms.

The process of ontology matching for BIM- GIS integration is an operation that takes the BIM and GIS RDF graphs structures and produces a matching between elements of these graphs that corresponds semantically to each other. Due to subgraph complexity, ontology matching is a challenging task. Some approaches to similarity-measurement ontology matching have been proposed and presented in the literature review (chapter 2). This chapter proposes an algorithm for graph-based ontology matching for BIM-GIS using bipartite graphs to investigate the structural, conceptual, and relational similarities between the ontology graphs from these two platforms. These similarities are measured and evaluated.

The Graph Matching for Ontologies (GMO) algorithm uses matched pairs of IFC and CityGML objects represented in RDF triples (Hadi et al., 2016; Hu et al., 2005). It compares their structural similarities based on a set of semantic rules. We designed the graph matching for ontologies (GMO) to be self-reliant on lexical similarity (Ergen et al., 2007). The efficiency of the GMO algorithm is proved with multiple datasets from big to small and subsets of datasets that represent elements from BIM and GIS models applicable to the target application domain. The evaluation methodology was designed, developed, and implemented to discover ontological BIM and GIS entities, relationships, attributes, and values. The results are presented in a comparative way between the different most used and trusted ontology matching techniques where the objectives were:

- Demonstrate the feasibility and efficiency of these semantic matching techniques for BIM-GIS integration modelling. Identify the best technique(s) suited for BIM-GIS models.
- Evaluate the impact of internal resources and external parameters on the matching quality.

- Evaluate semantic matching techniques' performance by measuring the time/memory ratio of each one.
- Outline future research directions for the BIM-GIS mapping algorithm and matching techniques.

The chapter is structured as follows: Section 4.2 introduces the fundamental concepts of mapping between IFC and CityGML, a semantic web framework, and RDF graphs. Section 4.3 presents the proposed mapping methodology with the mathematical development of the similarity matrices used by the graph matching of ontologies (GMO) and the transformation of the coordinate system matrix. Section 4.4 presents implementation, followed by a graph-matching evaluation. Results are discussed in section 4.5, followed by a conclusion and future work in Section 4.6.

4.2 Background

The BIM Industry Foundation Classes and its associated XML specification(ifcXML) is an object-oriented open standard initiated in 1994 by BuildingSMART for Architecture, Engineering, and Construction (AEC) application domains. To share and exchange the information AEC professionals use during a building project's entire lifecycle. IFC supports many geometric representations and can be rich in semantic information related to the physical information that describes buildings and the required information for building project management, including planning, cost estimation, scheduling, and operations (Jena, 2011; Borrmann et al., 2006). The IFC entities can be expanded to a tree structure. These entities are tightly linked to other entities by inheritance relationships and inverse relationships. There are other entities linked to themselves, such as *ifcObjectDefinition* by the inverse *decomposed by* the relationship, which adds other entities to the semantic information of the model.

On the other hand, the Geographic Markup Language (CityGML) was the first 3D-GIS schema to provide rich semantics information (Agdas and Ellis, 2010; Anumba et al., 2008). CityGML supports component-based modelling in which different building components are assigned unique IDs, names, and descriptions (Agdas and Ellis, 2010; Eastman et al., 2009). CityGML uses five levels of detail (LoDs) that vary from LoD0, a regional 2D map, to the highest LoD4, which describes interior building features, including furniture. The LoDs definitions provide different data solutions for various applications and systems (Azhar et al., 2011; El-Gohary and El-Diraby, 2009). Furthermore, CityGML Application Domain Extensions (ADE) (Eastman et al., 2011) help users use and even create their own custom "CityGML" extensions for their specific domain applications and to broaden the model compatibility with other systems (Beetz, 2009; Choi et al., 2008).

IFC and CityGML models are component-based models. The information about a single building component can be extracted separately, allowing for decoupling and offering accurate semantic and spatial analysis. This component-based mapping is the basis for the ontology-based matching proposed in this research, compared to previous integration strategies and methods.

Several conceptual frameworks were proposed to match pr map data standards from BIM and GIS domains (Bakis et al., 2007; Döllner and Hagedorn, 2007). The first integration attempt started by merging IFC and CityGML into the Unified Building Model (UBM), in which entity definitions from BIM and GIS schemas were extended according to the entity definitions in the two schemas. However, this method did not provide details for geometry transformation (Deutsch, 2011; Eastman et al., 2009) and the criteria for selecting from which domain geometry should be taken. Another approach proposed a framework to generate 3D CityGML models using IFC models where the transformation of different LoDs and semantic information were discussed (Döllner and Hagedorn, 2007). However, the authors did not consider matching geometric and semantic information rules in IFC and CityGML. They also failed to mention the details of LoDs transformation in CityGML. For example, given a building model in LoD4 in CityGML, the framework cannot translate to a lower LoDs. In another attempt (Kim and Grobler, 2007; Mitchell and Schevers, 2007), the researchers proposed conceptual requirements for generating 3D building models from un-interpreted 3D models using CityGML as the transformation medium. The matching rules were developed in this study to transform CityGML models into an IFC model. However, they did not consider semantic information matching in the process. The developed mapping was unidirectional and only allowed transformation from the GIS model to BIM. In (Eastman et al. 2011), the methodology proposed a new unified schema called City Information Modelling (CIM), containing five categories of entities: the building, transportation, city furniture, Message Exchange Pattern (MEP), and water body. Their matching rules between IFC and CityGML, on the one hand, and CIM, on the other hand, allow information exchange between IFC and CityGML. However, in (Eastman et al. 2011) (El-Gohary and El-Diraby, 2010), the research did not provide details on the matching process. Another research that looked at the IFC and CityGML matching was by (Cutting-Decelle et al., 2007; El-Diraby and Osman, 2011), which proposed a system architecture for the effective integration of BIM into a GIS-based facility management (FM) system using an Extract, Transform, and Load (ETL) method. The authors also proposed a transformation from BIM to different LoDs in CityGML, but the process was semi-automatic and manual inputs were required.

To summarise, all former IFC and CityGML integration attempts have their benefits in bringing these domains closer through mostly interoperability, but still some limitations, such as:

- No bidirectional between IFC and CityGML integration or interoperability methodologies were developed.
- The matching between IFC levels of development and CityGML levels of details was not completed.
- The IFC semantic information was partially translated.

It is important to think about a new data model and a representation framework to overcome the integration constraints. A rich class and property modelling framework is needed to be used as an integrated data model for sharing ontologies, data dictionaries, and taxonomies. Also, the framework needs to be flexible in adding data without updating schemas of data sources with core architecture (Ergen et al., 2007). We might find an appropriate description language on the web as a valid alternative to represent BIM or GIS models. The web platform contains information about any universally used concept, the language expressing this information cannot be restricted to the domain-specific schema.

Therefore, in the case of BIM and GIS integration, a generic and flexible language is needed to describe and link information from very different knowledge domains easily. The content must be well defined for these two very diverse and large knowledge domains to maintain inter-relations and partition possibilities and permit practitioners to describe concepts in separate limited scopes and still relate them to the concepts in different scopes (Elbeltagi and Dawood, 2011; Cromley and McLafferty, 2012).

Ontology engineering fundamentals and semantic web framework describe the meaning of concepts using labelled, directed graphs based on a defined logic. Each node in the graph represents an object (or a concept) in the real world. The graph's edge (also called arc) represents a logical relation between objects. Therefore, the graph represents a set of logic-based declarative sentences (El-Diraby et al., 2005; El-Gohary and El-Diraby, 2010) linkage objects and their associated relationships and properties to other objects in the same graph. Likewise, the AEC industry uses EXPRESS schemas (Bakis et al., 2007; Choi et al., 2008) to construct that logic within the Standard for the Exchange of Product model (STEP) file standard (3D model using Standard for the Exchange of Product Data - IFC schema). The Geospatial communities use the Unified Modelling Language (Welcome To UML Web Site!) within OGC standards (CityGML) for the same purpose. In contrast, the semantic web uses the Resource Description Framework (RDF) as the primary language to represent information and graph structure, referred to as the RDF graph. It can describe different levels of ontologies and bring homogeneous and heterogeneous data into a unified model using RDF representation.

The RDF graphs can promote data merging even with differences in the underlying schemas. They can facilitate changes or schema evolution by taking advantage of semantic web technologies (Hor and Sohn, 2021; Bakis and Pal, 2007; Bishr, 1998). The RDF format is considered a standard language within the semantic web domain to describe any information not limited to the web. The information becomes interchangeable between environments, whether these are complex systems of software applications or any other environment making the semantic web and RDF graph a web of linked data of all forms, shapes, or sources (Elbeltagi and Dawood, 2011). This supersedes individual applications or limitations through links between identical or related entities (Hijazi et al., 2011).

An RDF graph can be constructed by applying a logical AND operator to logical statements containing objects and relations. The output of these statements is referred to as RDF triples. Each triple consists of a subject, a predicate, and an object representing one entity, implying directionality in the RDF graph with a source and target. Also, each concept has an assigned Unique Resource Identifier (URI) (Alders, 2006; Akicini et al., 2010), thereby explicitly labelling the RDF graph. This labelling extends to every object, subject, or predicate in the RDF graph and makes it uniquely defined through this URI. If two identical URIs are found in the graph, their semantic information is considered identical with some additional properties from either. The resulting RDF graph can be converted into a web semantic framework representation following a specific syntax like RDF/XML (which provides the highest levels of expressiveness), N-Triples, Turtle, SPARQL, and Notation-3 (N3) (Hor and Sohn, 2021).

When dealing with knowledge from multiple domains, ontological representation can be the medium to carry the knowledge and facilitate integration and interoperability between domains. Depending on the level of detail (Eastman et al., 2009; Bakis and Pal, 2009a), ontologies can be categorized into top-level domain and application ontologies. The top-level ontologies express general concepts like time, space, function, etc. These concepts are applicable across many domains. The domain ontology is designed to represent and formalize concepts in the same and specific domain, such as AEC (Jena, 2011; Bansal, 2011). Therefore, the application ontology describes the semantics of focused domain-oriented applications and defines relevant concepts for applications like the BIM or GIS domains. Application ontologies play a vital role in facilitating integration between distinctive information types. As a result, this ontology will achieve integration and interoperability between BIM and GIS, as seen in many other fields like business intelligence, biomedical informatics, and functional design (Hor and Sohn, 2021; Elghamrawy et al., 2007).

Previous studies focused on ontology modelling and schema-matching design in the AEC field. The Unified Building Model (UBM) (Döllner and Hagedorn, 2007) could be the best one to represent efforts in designing and developing an ontology for matching between BIM models and GIS models. In the UBM, the relationships between building components are used to find and match objects from BIM and GIS models. However, their approach focused on the component level and not on semantic and geometry information, which could provide a better and clear relationship for matching BIM models and GIS models. At its core, semantic integration is the ability to connect meaning to regular concepts to structure a domain of knowledge and generate information to support functioning systems in continuous evolutions like BIM and GIS systems (Döllner and Hagedorn, 2007; Eastman et al., 2011; Karan et al., 2014). It brings many application-oriented data models that can have a significant impact on the integration mainly because:

- Richer class, property modelling, and inference offered by RDF and RDFs (RDF schema) compared to relational database management systems (RDBMS) and NoSQL.
- Higher flexibility and ability to develop ad-hoc RDF sub-models from BIM and GIS resources, with no need to update the global schemas, given that SPARQL syntax can include optional matching clauses working with sparse data representations.
- Leveraging shared ontologies from BIM and GIS models can enhance linkage from various sources.
- Graph theory and graph algorithmics are well established and proven to solve many complex science and engineering problems using graph data structures.

This study employs a graph ontology algorithm on RDF graphs to accomplish the most accurate integration between these two platforms. It depicts developing an integrated model that can effectively provide powerful benefits to indoor and outdoor applications. It enables filtering capabilities from BIM-RDF and GIS-RDF graphs into a semantically integrated model combining BIM classes with GIS features. Table 4.1 summarises the top semantic-based systems used for semantic matching algorithms, including description, matching process methodologies, and steps.

Matching System	Description	References
Structure-Preserving Semantic Matching (SPSM)	A matching algorithm for tree-like structures, which ensures that: (i) Nodes are related to one-to-one. (ii) Leaf nodes are matched against leaf nodes in the same way as internal nodes. The process is decomposed into two steps: a) Node matching using S-Match b) Tree Matching uses obtained results and the structure of the trees to get an approximate matching.	Giunchiglia et al., 2008
Bootstrapping Matching (BSM)	Uses the Wikipedia category hierarchy as an external resource (E). In short, it constructs a forest (set of trees) TC for each matching candidate C, which roughly corresponds to a selection of super-categories of a given class. Next, the forest TC and TD are compared to determine the kind of relation between concepts C and D. The strength of this algorithm is the use of noisy community-generated data available on the Web. However, because the Wikipedia Web Service is queried online, its performance is poor.	Jain et al., 2010
The Rule Ontology Matching Approach (AROMA)	A hybrid, extensional and asymmetric matching algorithm based on the association rule paradigm. It selects the relevant terms contained in ontologies to discover equivalence and sub-assumption relations, which are modelled as rules. The main steps are: a) Acquisition and selection of relevant terms for each concept, b) Discovery of significant implications between both hierarchies	David et al., 2006
The Mapping by Particle Swarm Optimization (MapPSO)	Especially suited for aligning large ontologies. It treats ontology alignment as an optimization problem, applying a discrete variant of particle swarm optimization (Kennedy and Eberhart 1997, Correa et al. 2006). Each particle represents a candidate alignment initialized from a set of random one-to-one mapping. At the same time, the fitness value is measured as the weighted sum of similarities from a configurable set of elementary matches.	Bock and Hettenhausen 2012
Semantic Matching (S-Match)	An algorithm that relies on semantic information encoded in lightweight ontologies to perform a two-step alignment: a) The terms of ontologies are translated into formal propositional formulas, and b) The problem is reached to propositional validity problem using external resources.	Giunchiglia et al., 2004

Table 4.1 Ontology alignment techniques based on Ontology Alignment Evaluation Initiative (OAEI) Standard.

4.3 Ontology Mapping Strategies

A Graph Mapping Ontology (GMO) algorithm is developed for BIM and GIS ontology matching alignment workflow, as presented in figure 4.1. GMO is principally for integrating entities from IFC and CityGML heterogeneous models translated to RDF format. The mapped entities are evaluated against measuring system alignment extraction techniques for quality purposes, including data and schema inspection.

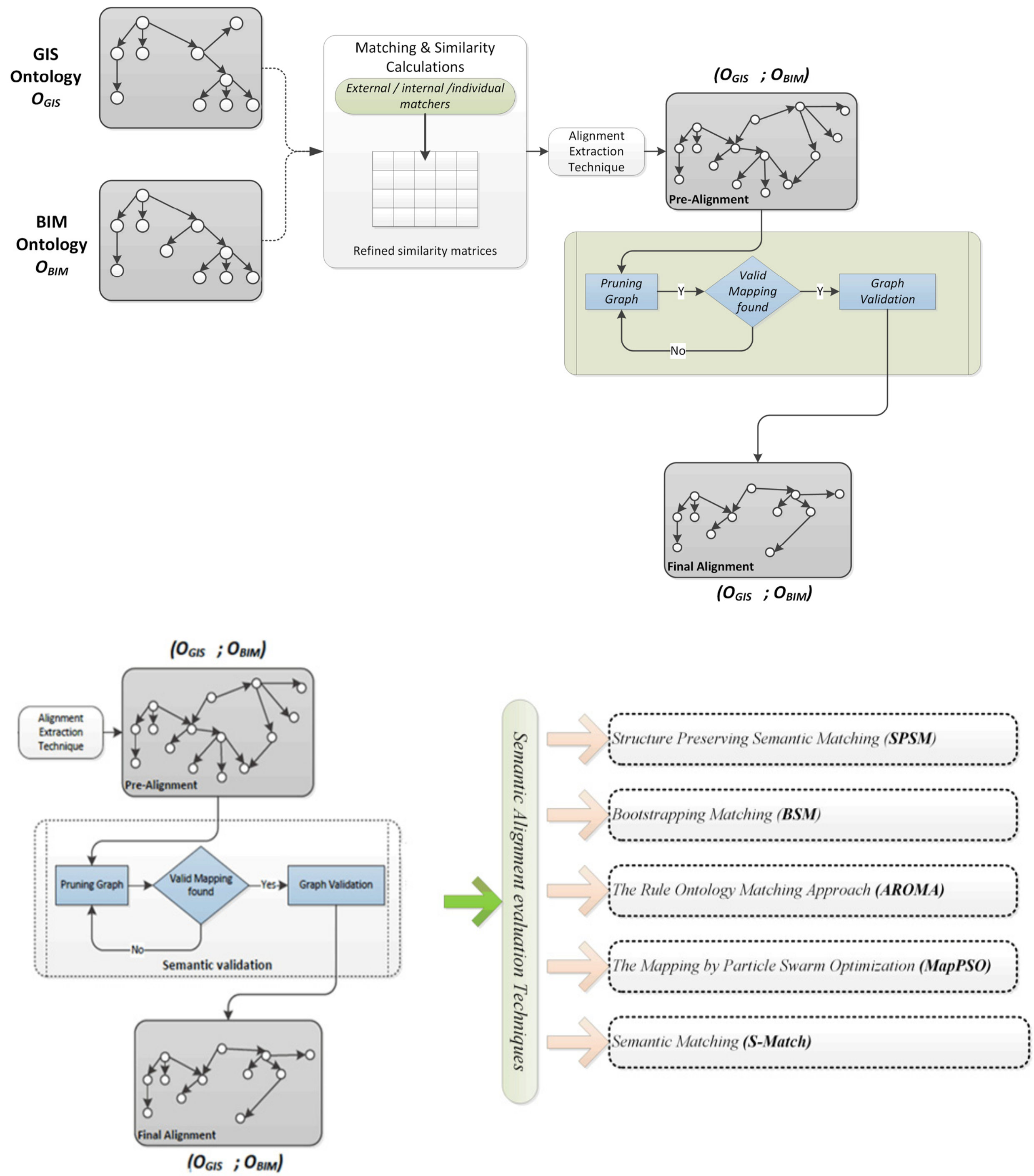


Figure 4.1 Workflow of CityGML – IFC ontologies mapping (Top) End-to-End Workflow (Bottom) Alignment technique method evaluation processes

Once the similarities between BIM and GIS graphs are identified, the system alignment techniques are selected based on availability, flexibility to customization, the possibility of automation with no user inputs, and relevancy in providing results. The diagram shown in figure 4.1 provides the main tasks for the graph matching process and the generation of the similarity matrices using alignment extraction techniques for BIM/GIS RDF graphs. A graph pruning process is applied to eliminate irrelevant triples, followed by a validation process of the new resulting graph. Iteration times can vary significantly depending on the number of triples integrated from IFC RDF and CityGML RDF graphs. Indeed, the one-to-one matching is completed when a defined estimated low similarity is reached. The most recent Ontology Alignment Evaluation Initiative (OAEI) methods have been tested for generating similarity matrices and GMO convergence tests. Moreover, for the performance evaluation of the tested algorithms, the memory and speed indicators have been considered as follows:

- **Memory**: The amount used to match BIM and GIS entities. It includes algorithm execution and the memory used by the underlying ontology management.
- **Speed**: Measured as the matching processing time (in seconds) used by the test machine to get the alignment through the execution.

The Graph Mapping for Ontologies algorithm assessment is performed during data use by ontology-based applications, following the classification introduced by (Elghamrawy et al., 2007). We framed the application into the system integration category due to the number of similarities between both BIM and GIS application fields. Accordingly, high precision and recall are prioritized, while automation and speed levels can be lower. The level of automation is a requirement and a motivation of this research, so we do not consider user intervention. Hence, leaving speed aside, the focus will be on achieving the highest possible precision and recall. Based on the above assumptions and available matching systems, those with the top OAEI results are listed in table 4.1.

The similarity threshold is the common measured factor between all techniques used by each ontology matching system cited in table 4.1. The threshold values are indicators of the similarity threshold used to trim the set of correspondences. Some correspondences are discarded when confidence measurements are less than the established threshold. Its influence on precision and recall were analyzed in the testing.

4.4 Translation and Integration Strategy

A semantic integration approach allows sharing of relevant information without requiring human intervention. A large volume of datasets needs to be transferred and automatically merged between the BIM and GIS domains using metamodels to create one unique BIM-GIS model. Given the nature of the information in BIM and GIS, the integration process brings two different data structures together rather than just the reasoning and logic of a centralized model, similar to artificial intelligence (AI) systems. To semantically integrate and query spatial and non-spatial data from BIM and GIS, we need to have a standardized set of ontologies for BIM and Geospatial domains (Horrocks et al., 2003). It should be the same as used by the semantic web to handle distributed information on the web using the Resource Description Framework (RDF). Therefore, RDF can be enhanced with RDFs (RDF schema) functionalities to provide a greater level of abstraction than RDF, allowing resources to be described as instances of general classes. RDFs, at their core, provide an essential semantic capability by making semantic languages such as DAML, OIL, and OWL usable for data retrieval and, more importantly, for assisting in developing semantic systems. For instance, the light pole is made of steel ⇔ A subject donates "**light pole**," a predicate donating "**is made of**," and an object donating "**steel.**" Any subject, predicate, or object is called a triple, such as in figure 4.2 (the *light pole is made of steel*).

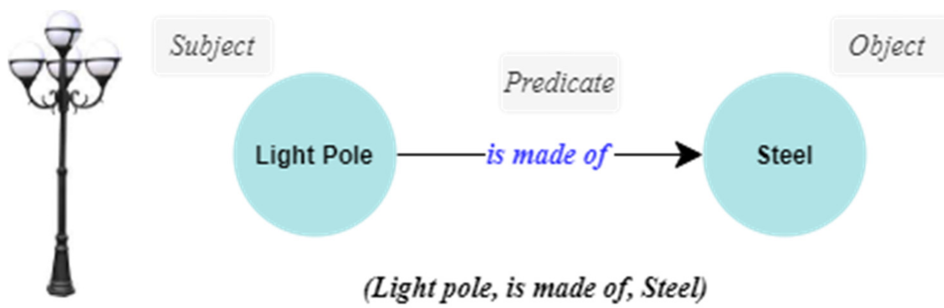


Figure 4.2 An example of RDF triples.

Every semantic concept (object or subject) or relation may be labelled with a globally unique short descriptive text (string): URI **(**Unified Resource Identifier**)**. Using a URI to prevent co-reference problems to connect data from third parties from applications and systems is essential. The URI enables the link between independent resources using the URI's properties as universal pointers.

Translating the BIM/IFC classes and the GIS/CityGML elements into a formal ontology language is essential to designing a BIM-GIS semantic integration and creating the IGIM system. RDF is an excellent complement to XML, which provides a flexible way to interchange data between applications. Hence, it is essential to translate IFC and CityGML files to XML-like; like GML3, we need to extract parameters from the IFC EXPRESS entity for geospatial data, as shown in figure 4.3.

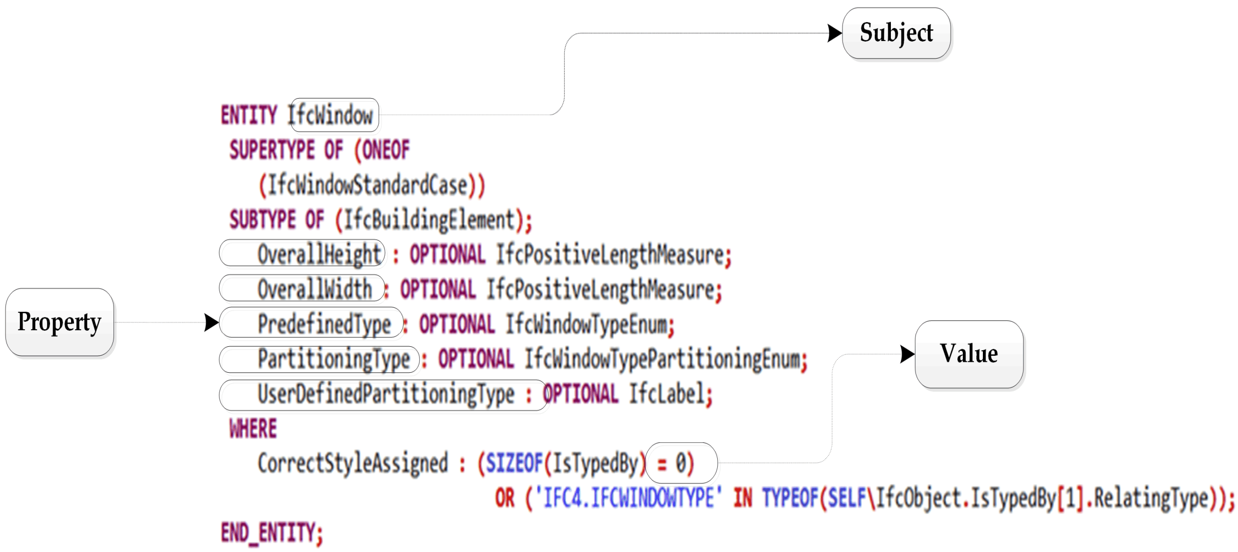


Figure 4.3 STEP EXPRESS representation of *IfcWindow*
Entity with corresponding RDF triples (subject/property/value).

Where:

- ***Subject***: representing physical elements such as a **window** in the building.
- ***Property***: Provide information assigned to the subject, like the height and width of the **window.**
- ***Value***: Specifying a property size.

Also, the same object can be represented in a UML diagram or as an OWL representation graph, shown in figure 4.4.

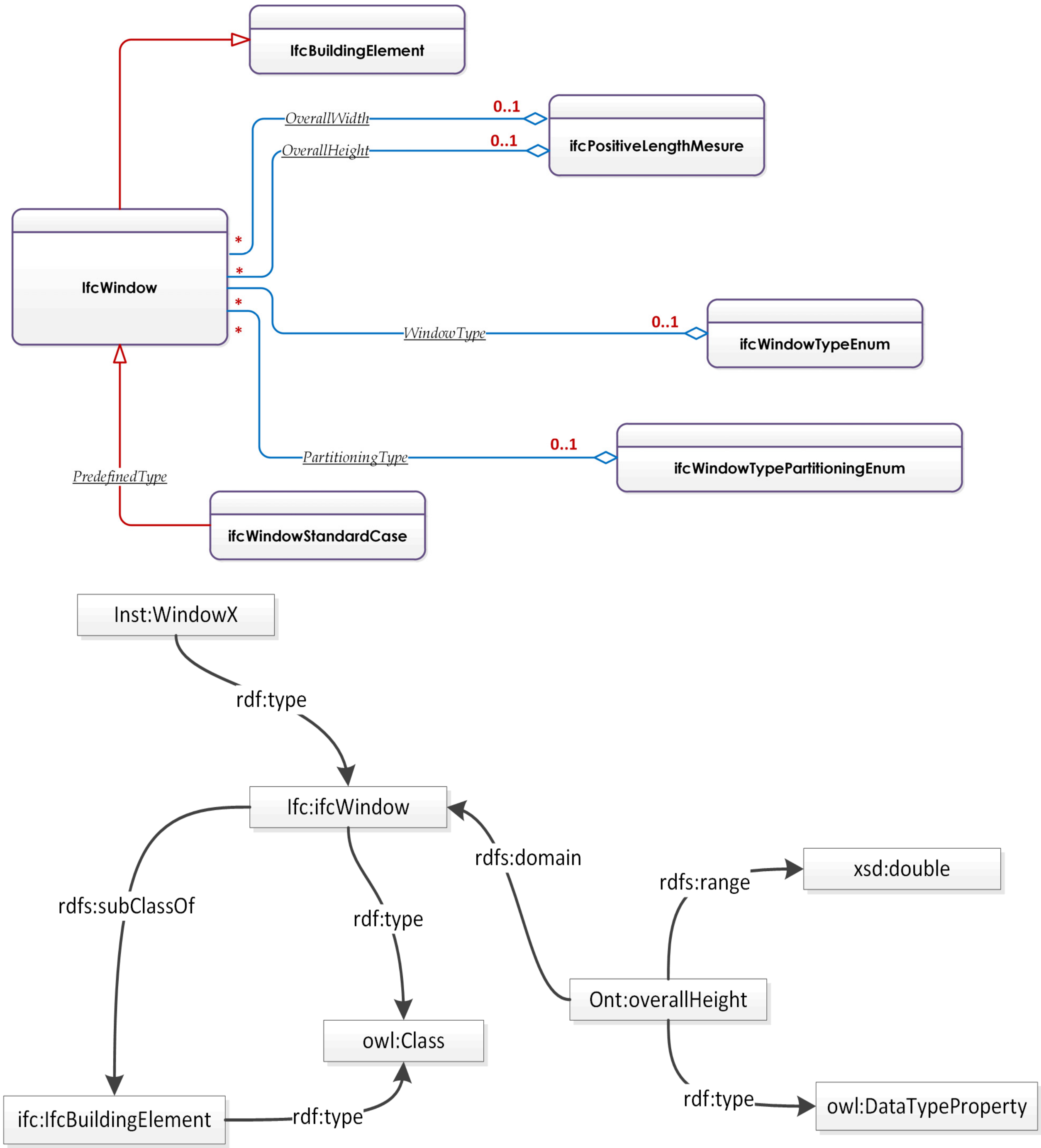


Figure 4.4 Window element Ontology in UML (left) and Ontology OWL description of *ifcWindow* class (right).

4.4.1 IFC to RDF Translation

It is critical to distinguish the four layers of any IFC model to develop an IFC to RDF semantic translation:

- Resource layer: The low-level concepts(objects) used for general tasks.
- Core layer: This layer represents the abstracted concepts defining the model.
- Interoperability layer: This layer defines shared or common concepts with other domains.

- **Domains and Applications** layer contains all objects/attributes specific to the target application.

The emergence of the semantic web framework led researchers and practitioners to design and develop methods for converting IFC schema into an ontology-based language to produce a knowledge-based system. This system would be human-readable and can be processed by the machine, with information-rich and platform-independent frameworks able to provide integration and interoperability to exchange heterogeneous data from diverse domains. The semantic web features and characteristics of RDF graphs have motivated this research work on converting the IFC EXPRESS schema into OWL and RDF-XML ontologies.

4.4.2 Integration Strategy Considerations

As indicated above, the CityGML is a standard open-source model, XML compatible, implemented as an application schema for GML3 to represent 3D-urban city objects and can be shared with multiple geospatial systems and applications where location and geometry information is needed. Likewise, the IFC allows the storage and exchange of 3D object models for different applications. However, for the integration, it is essential to consider the following:

- Multi-scale Modelling is also known as levels of details from landscape up to an interior model of the building.
- All components modelling: of semantics, geometry, topology, and appearances.
- Semantical and geometrical Coherence: direct relations between semantic objects and their geometrical representation.
- External references and appearance (textures): to be included in the modelling process.
- Application Domain Extensions (ADE): enhancing the model with custom components or sub-systems for a specific functional task.
- Generic city objects and attributes: these are the objects that are not covered in the model and lack attributes; CityGML has a class called GenericCityObject to model these 3D Objects.

4.4.3 IFC-CityGML to RDF Translation

Once the IFC to RDF translation is complete (section 4.4.1), ontology graph mapping and evaluation strategy are next steps. The RDF bipartite graph model (Fu et al., 2007) is used in this chapter to represent BIM and GIS ontologies mapping using adjacency and similarity matrices. The developed mapping approach compares the structures of entities of interest to qualify the degree of similarity of triples. The OWL ontology can be mapped to RDF graphs, given the RDF model. The spatial coordinates must be extracted and transformed from BIM local coordinate systems to correct the geo-referenced coordinate GIS system to represent the building element within the GIS context accurately. The coordinate transformation matrix was constructed to capture the correspondences between the two reference systems.

IFC uses a local coordinate placement system to determine the position of objects, despite the convenience it brings while copying entity information. However, the local placement system causes trouble mapping between IFC and CityGML.

- The IFC ifcProject class offers information about the coordinate reference system, such as:
- BIM Project coordinate system.
- The Space dimension coordinate information.
- The geometric representations precision information.
- Map conversion information project coordinate system and the geospatial coordinate system (optional).
- True north (optional).

However, as indicated above, this information is not always available with the IFC model. Therefore, to transform the local placement system to the world projected coordinate system, every *ifcAxis2Placment* class entity information will be represented into a 4x3 transformation matrix multiplying the points to the series of transformations resulting from pointing location (geographic shape) in the world coordinate system. Constructive Solid

(IfcCsgSolid) or Swept Solid (IfcSweptAreaSolid) geometries represent solid building components in IFC models. In our proposed framework, all the solid models will be broken into surfaces to represent the object's exterior. The coordinates of the surfaces are then transformed into a world coordinate system and written into CityGML to be propagated into to georeferenced the integrated model.

The following transformation Matrix was developed for the engineering building described later.

$$\begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{X} \times \boldsymbol{Z} \\ \boldsymbol{Z} \\ \boldsymbol{Origin} \end{pmatrix} = \begin{pmatrix} 0.965925826289006 & -0.258819045102752 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ -22350.9150923124 & 45244.4871754247 & 0 \end{pmatrix}, \tag{1}$$

The coordinate transformation can be verified after parsing IFC and CityGML files into a semantic data format such as RDF or a Turtle format.

The following notations to each formal definition of the matrices are used in the matching process to outline the procedures for ontological matching across IFC and CityGML:

- $\boldsymbol{O_{BIM}}$: BIM ontology.
- $\boldsymbol{O_{GIS}}$: GIS ontology.
- $\boldsymbol{G_{BIM}}$: RDF bipartite of O_{BIM}.
- $\boldsymbol{G_{GIS}}$: RDF bipartite for O_{GIS}.
- $\boldsymbol{A}$: Adjacency matrix representing a bipartite graph of ontology**.**
- $\boldsymbol{A_{ES}}$: Matrix to represent connections from external entities.
- $\boldsymbol{A_{S}}$: Matrix to represent connections from internal ontology entities.

- ***A_E***: Matrix to represent connections from statements to external entities of the ontology.
- ***A_{OP}***: Matrix to represent the connections from statements to internal entities

Each of the entities will be classified as:

  - ***Properties****:* using (rdf:type , rdfs:subClassOf) respectively and RDF type and RDFs schema.
  - ***Classes****:* this can be any defined class like Building or Terrain etc.
  - ***Instances****:* Representing single individuals and data literals (as appears on the source).

For the entities with similar class classifications and roles within the ontology (subject or predicate), these entities would have a greater possibility of possessing similar relationships. They will be utilized in the matching process between O_{BIM} and O_{GIS} with data types, build-in properties, and URI. Any two URIs indicate identical semantics in the matching process, leading to the similarity matrix representation of BIM ontology entities to GIS ontology entities. The external entities are mostly vocabularies defined in the RDF schema (RDFs), data literals (values), and built-in data types. They may also include some common vocabularies used in OBIM and OGIS models and those specified in an input matching. Consequently, separating internal and external entities is a relative matching process used in this research. Therefore, when external entities of O_{BIM} are not used as subjects in O_{GIS}, the Matrix A_{ES} is a zero Matrix.

Next, the structural similarity of O_{BIM} and O_{GIS} is measured for ontology between the O_{BIM} and O_{GIS}, followed by an entity's comparison in these two graphs for a given ontology.

The similarity between two from two different ontologies derives from the triples accumulation of similarities of triples with the same role (Subject, Predicate, Object) in the triples from O_{BIM} and O_{GIS}, including external entities of the same role in the two triples being compared.

The measurement of structural similarity is based on the similarity between directed graph vertices (Ergen et al., 2007) and applying the Graph Matching for Ontologies (GMO) approach described in (Fu et al., 2007). The ontology matching and alignment techniques detect and link concepts shared between two BIM and GIS ontologies. The primary sources of heterogeneity are classified into three categories (Elghamrawy et al., 2007):

- ***Terminological***: referring to concepts with the same names, i.e., *IfcWindow* in IFC format and Window in CityGML.
- ***Syntactic***: referring to differences of ontologies in representing data structures, relationships, and objects' connections; an example of that, the *IfcDoor* and *IfcWindow* are subclasses of *IfcBuildingElement* in IFC, while Door and Window, their corresponding classes in CityGML, are subclasses of Opening in CityGML.
- ***Conceptual*****:** referring to a different perspective of the same domain of interest in the model. This could be because of a difference in coverage (even if there is an overlap), granularity (level of details), or perspective (thematic interest). An example is an *IfcSlab* class in IFC, which would be a superclass of GroundSurface, FloorSurface, and CeilingSurface in CityGML**.**

It is possible to align BIM and GIS ontologies' structural similarities using RDF bipartite graph model by using the updating equations for the similarity matrix: (Hu, W., Jian, N., Qu, Y., & Wang, Y., GMO, 2005)

$$X_{k+1} = BX_kA^T + B^TX_k\ ,\ \ k = 0,1 \tag{2}$$

In which:

- $\boldsymbol{X_k}$ is the $\boldsymbol{n_B} \times \boldsymbol{n_A}$ matrix of entities.
- $\boldsymbol{x_{ij}}$ at iteration *k*.
- $\boldsymbol{A}$ and $\boldsymbol{B}$ are the adjacency matrices of G_{BIM} and $G_{GIS,}$ respectively.

It is proved that the normalized even and odd iterations of this updating equation converge, and limit Z_{even} is among all possible limits, the only one with the largest *1-norm.* This limit is taken as the similarity matrix: (Hu, W., Jian, N., Qu, Y., & Wang, Y., GMO, 2005)

$$\boldsymbol{A} = \begin{pmatrix} 0 & 0 & A_{ES} \\ 0 & 0 & A_S \\ A_E & A_{OP} & 0 \end{pmatrix},\ \ \boldsymbol{B} = \begin{pmatrix} 0 & 0 & B_{ES} \\ 0 & 0 & B_s \\ B_E & B_{OP} & 0 \end{pmatrix},\ \ \boldsymbol{X_k} = \begin{pmatrix} E_{BA} & & \\ & O_k & \\ & & S_k \end{pmatrix} \tag{3}$$

The structural similarity updating matrices are defined as:

$$O_{k+1} = B_S S_k\, A_S^T + B_{OP}^T\, S_k A_{OP} \tag{4}$$

$$S_{k+1} = B_E E_{BA} A_E^T + B_{ES}^T E_{BA} A_{ES} + B_{OP} O_k A_{OP}^T + B_S^T\, O_k A_s \tag{5}$$

If the limits of normalized even of iteration with:

$$O_0 = \mathbf{1}\ and\ S_0 = \mathbf{1},\ \ \mathbf{1}\ denotes\ vector\ or\ matrix\ where\ \mathbf{1}_{\boldsymbol{ij}} = \mathbf{1} \tag{6}$$

The limit of O_k is taken as the structural similarity matrix of ontologies O_{BIM} to O_{GIS}. However, the structural similarity formulation, equations (4) and (5), differ from the one in (1) in these aspects:

1. A directed bipartite graph is used instead of the directed graph.
2. Nodes are categorized differently.
3. External entities' semantic similarities are kept unchanged during the updating process.

These aspects will impose a structural similarity refinement process. In most cases, the entities in each ontology are classified as properties, classes, and instances (individuals and data literals). They have a similar classification (*RefLatitude* and *geo. latitude* is class entities) and role (e.g., *subject* or *predicate*), which increases the chance of having a similar relationship or concept. The build-in properties, datatypes, and URIs used in BIM and GIS ontologies are considered when mapping between two domain ontologies, and any two identical URIs result in identical semantics.

After successful classification, we can refine the matrix representation form to the following:

$$A_{ES} = \begin{pmatrix} A_{EPS} \\ A_{ECS} \\ A_{EIS} \end{pmatrix} \tag{7}$$

$$A_{S} = \begin{pmatrix} A_{PS} \\ A_{CS} \\ A_{IS} \end{pmatrix} \tag{8}$$

$$A_{E} = (A_{EP},\ A_{EC}, A_{EI}) \tag{9}$$

$$A_{OP} = (A_{POP}, A_{COP}, A_{IOP}) \tag{10}$$

Where:

- $\mathbf{A_{EPS}}$, $\mathbf{A_{ECS}}$, and $\mathbf{A_{EIS}}$ represent connections from external properties, classes, and individuals to triples.
- $\mathbf{A_{PS}}$, $\mathbf{A_{CS}}$, and $\mathbf{A_{IS}}$ represent the connections from internal properties, classes, and individuals to triples.
- $\mathbf{A_{EP}}$, $\mathbf{A_{EC}}$, and $\mathbf{A_{EI}}$ represent the connections from triples to external properties, classes, and instances (including data literals).

- $\mathbf{A_{POP}}$, $\mathbf{A_{COP}}$, and $\mathbf{A_{IOP}}$ represent the connections from statements to internal properties, classes, and instances, respectively.

The same refinement can also be applied to ontology O_{GIS}:

$$B_{ES} = \begin{pmatrix} B_{EPS} \\ B_{ECS} \\ B_{EIS} \end{pmatrix} \tag{11}$$

$$B_S = \begin{pmatrix} B_{PS} \\ B_{CS} \\ B_{IS} \end{pmatrix} \tag{12}$$

$$B_E = (B_{EP},\ B_{EC}, B_{EI}) \tag{13}$$

$$B_{OP} = (B_{POP}, B_{COP}, B_{IOP}) \tag{14}$$

The similarity matrix of external entities and the structural similarity matrix of ontologies have a diagonal structure: (Hu, W., Jian, N., Qu, Y., & Wang, Y., GMO, 2005)

$$E_{BA} = \begin{pmatrix} EP_{BA} & 0 & 0 \\ 0 & EC_{BA} & 0 \\ 0 & 0 & EI_{BA} \end{pmatrix} \tag{15}$$

$$O_k = \begin{pmatrix} P_k & 0 & 0 \\ 0 & C_k & 0 \\ 0 & 0 & I_k \end{pmatrix} \tag{16}$$

Where:

- EP_{BA}, EC_{BA}, and EI_{BA} represent the similarity matrices of external properties, classes, and individuals.
- P_k, C_k and I_k represent the similarity matrices of inner properties, classes, and individuals, respectively.

The updating equations for the structural similarity matrix are all refined as follows:

$$P_{k+1} = B_{PS} S_k A_{PS}^T + B_{POP}^T S_k A_{POP} \tag{17}$$

$$C_{k+1} = B_{CS} S_k A_{CS}^T + B_{COP}^T S_k A_{COP} \tag{18}$$

$$I_{k+1} = B_{IS} S_k A_{IS}^T + B_{IOP}^T S_k A_{IOP} \tag{19}$$

$$S_{k+1} = B_{EPS}^{T} EP_{BA} A_{EPS} + B_{ECS}^{T} EC_{BA} A_{ECS} + B_{EIS}^{T} EI_{BA} A_{EIS} + B_{EP} EP_{BA} A_{EP}^{T} + B_{EC} EC_{BA} A_{EC}^{T} + B_{EI} EI_{BA} A_{EI}^{T} + B_{POP} P_k A_{POP}^{T} + B_{COP} C_k A_{COP}^{T} + B_{IOP} I_k A_{IOP}^{T} + B_{PS}^{T} P_k A_{PS} + B_{CS}^{T} C_k A_{CS} + B_{IS}^{T} I_k A_{IS} \quad (20)$$

From a computing resources consumption point of view, the advanced formulation of structure similarity has some advantages when applied to big RDF models, such as:

- Good computing performance due to using the matrix computation with blocks.
- Avoiding unnecessary computing of similarity between different kinds of entities, e.g., the ones between classes and properties.

4.5 Ontology Graph Mapping Algorithm Implementation and Evaluation

The implementation of the GMO process to integrate and evaluate O_{BIM} and O_{GIS} is outlined in the steps below:

Step 1: Parse O_{BIM} and O_{GIS} and transform both to RDF bipartite graphs

Step 2: Classify entities (including anonymous ones) in O_{BIM} and O_{GIS} as classes, properties, and instances

Step 3: Coordinate O_{BIM} and O_{GIS} using coordination rules to be implemented in GMO.

- ***Discarding***: Some triples within the ontology may become redundant or not needed for structural comparison (e.g., RDF - IFC file header); a rule in GMO will be designed to discard these types of statements.
- ***Merging***: Two entities from O_{BIM} and O_{GIS} could be stated to be the same or equivalent to each other, and/or has-a complimentary property to another entity from

the other graph, and/or is-a property or a relationship that completes another similar entity. All these entity types should be merged in the resulting $IGIM_{RDF}$ graph.

- ***Inference***: Adding some inferred triples to the $IGIM_{RDF}$ graph with some inference rules would be helpful; for example, adding a triple to state a range of values.
- ***List***: Instead of using RDF collection vocabularies, a list rule can be presented to express the relation between a list and its members.

Step 5: Set up matrix representation for O_{BIM} and O_{GIS}

Step 6: Initialise the similarity matrices P_k, C_k, I_k, S_k to matrix **1**

***Step** 7*: Run the steps with updating equations until some pre-defined convergence is reached. (Hu, W., Jian, N., Qu, Y., & Wang, Y., GMO, 2005)

$$P_{k+1} = B_{PS} S_k A_{PS}^T + B_{POP}^T S_k A_{POP} \qquad (21)$$

$$C_{k+1} = B_{CS} S_k A_{CS}^T + B_{COP}^T S_k A_{COP} \qquad (22)$$

$$I_{k+1} = B_{IS} S_k A_{IS}^T + B_{IOP}^T S_k A_{IOP} \qquad (23)$$

$$S_{k+1} = B_{EPS}^T EP_{BA} A_{EPS} + B_{ECS}^T EC_{BA} A_{ECS} + B_{EIS}^T EI_{BA} A_{EIS} + B_{EP} EP_{BA} A_{EP}^T + B_{EC} EC_{BA} A_{EC}^T + B_{EI} EI_{BA} A_{EI}^T + B_{POP} P_k A_{POP}^T + B_{COP} C_k A_{COP}^T + B_{IOP} I_k A_{IOP}^T + B_{PS}^T P_k A_{PS} + B_{CS}^T C_k A_{CS} + B_{IS}^T I_k A_{IS} \qquad (24)$$

Step 8: Find one-to-one by using similarity matrices P_k, C_k, and I_k

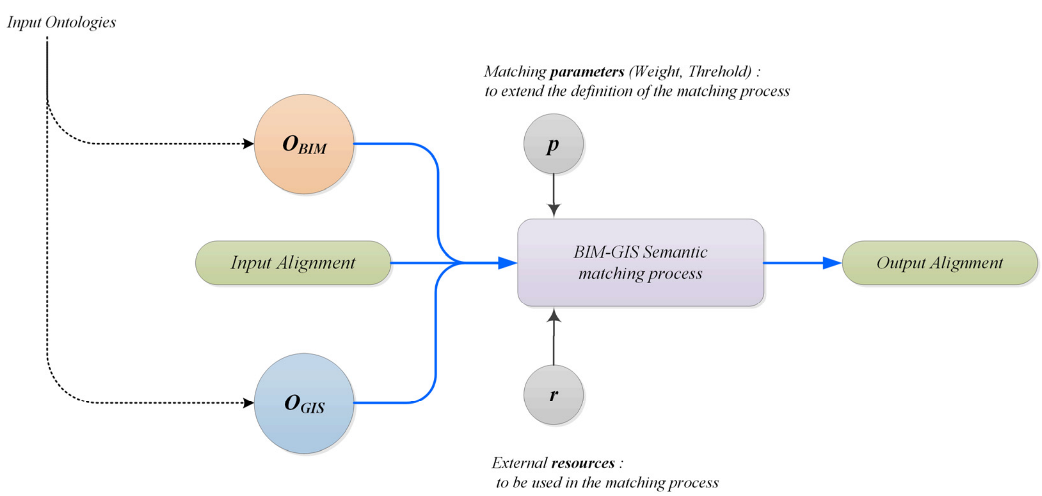
Step 9: Output additional matching pairs from O_{BIM} and O_{GIS}

This study set the iteration times in updating the structural matrices to 14. Also, the similarity matrix between external entities from both BIM and GIS ontologies E_{BA} considers as an identity matrix the following:

$$E_{BA} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (25)$$

Once the similarity workflow is completed, another process of triple matching is applied. The matching operation determines an output alignment for a pair of ontologies, O_{BIM} and O_{GIS}. Hence a given pair of ontologies, the matching task is finding an alignment between these ontologies (Hor et al., 2018). Some other parameters can extend the definition and enforce matching tasks, such as (figure 4.5):

- Input BIM and GIS ontologies.
- Matching parameters, for instance, weights and threshold.
- External resources such as common knowledge and domain-specific thesauri from BIM and GIS.



5-uple : ***(id, e_{bim}, $é_{gis}$, n, R)***

id *: Unique identifier of a given mapping element either from BIM or GIS.*

e_{bim}, $é_{gis}$ *: entity or triple (Obj, pred, Subj) – a property or class of O_{BIM} and O_{GIS} model.*

n *: Confidence Measurement holding the correspondence between e_{bim}, $é_{gis}$*

R *: Relation (same-as, equivalent to, as-is, has-a).*

Figure 4.5 Matching workflow with influential parameters

Based on many experimental tests, the one-to-one mapping was completed when an estimated low similarity was reached in updating equations (21), (22), (23), and (24). We can then get to the similarity matrix of O_{BIM} and O_{GIS}. The quality of similarity metrics can be evaluated using the following concepts:

- ***Precision***: This represents the ratio of correct found alignments concerning all retrieved pairs of entities from O_{BIM} and O_{GIS} adapted from information retrieval research from the GMO algorithm. It allows measurement of the correctness of the techniques used, defined as:

$$\boldsymbol{Precision} = \left|\frac{\boldsymbol{True\ Matches} \cap \boldsymbol{Predicated\ Matches}}{\boldsymbol{Predicated\ Matches}}\right| \quad (26)$$

- ***Recall***: Measures the ratio of the total number of expected correspondences expressed in the reference alignment. Thus, it allows determining the degree of completeness of the alignment. Like precision, these measures also come from information retrieval, and it is defined as:

$$\boldsymbol{Recall} = \left|\frac{\boldsymbol{True\ Matches} \cap \boldsymbol{Predicated\ Matches}}{\boldsymbol{Trues\ Matches}}\right| \quad (27)$$

- ***F-Measure***: Allows direct comparison between two models, which are often not comparable based on precision and recall because they usually have contradictory values. Higher precision implies lower recall and vice versa.

  The F-Measure aggregates the results of precision and recall and is defined as:

$$F - Measure = 2 \times \frac{[Precision \times Recall]}{[Precision + Recall]} \quad (28)$$

4.6 GMO Algorithm Implementation

The proposed method is applied to multiple GIS and BIM models (FZK-Hauz, FJK-Hauz, Smily-West, and Bien-Zenker Jasmin) and some sub-sets of the model, such as the case of PAN-AM Stadium and Life Science Building located at York University, Toronto, Canada. To handle BIM and GIS data and integrate them into one ontologically integrated geospatial information model, the concepts specified in RDF/XML must be converted into classes to realize the mapping of the functional tasks and combine GIS and BIM models. After writing and editing the ontology in a machine-processable ontology language, the syntax of the RDF models was validated using the RDF validation service at http://www.w3.org/RDF/Validator/. The resulting ontologies were verified and validated using the editor tool. The query results were used to ensure its quality and conformance to standards. This section describes the validation testing environment setup, testing model architecture, and results.

Table 4.2 presents the datasets' parsing results from the IFC and CityGML. These datasets can be categorized into two groups small (items 1 to 6) and significant (items 7 to 10) models, and separately, complete models (such as item 8, Bergeron Engineering Building at York University) and sub-model models (such as items 9, Life Science Building and 10, Pan-American Stadium at York University).

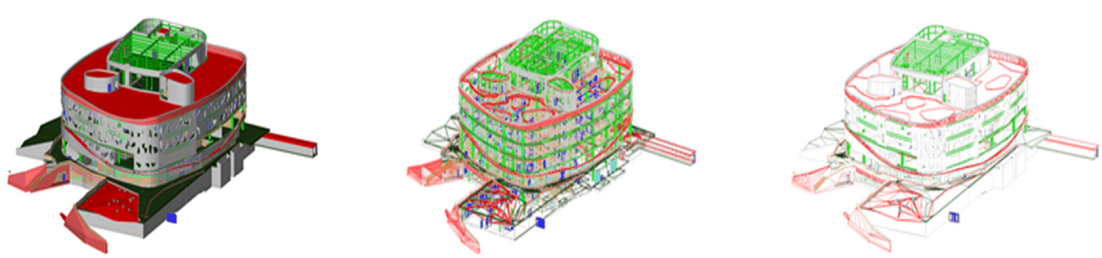


Figure 4.6 The Bergeron Engineering Building at York University completed the model with sub-models and used it in the testing process.

The selected datasets for the experiments led to conclusions because they are representative enough in the field, and they fulfil many key features:

1. ***Representative***: GIS and BIM datasets must be adopted by as many entities as possible. They must include a considerable number of different entries to achieve meaningful results.
2. ***Size***: The datasets are neither too small to be irrelevant nor too big to crash the testing system.
3. ***Reliable***: The publisher of the dataset is reliable on the international scene, and the samples should be as little modified as possible.
4. ***Operative***: Both GIS and BIM datasets and tools are fully compatible and work at least in many tests. This implies the collection of not null results with the minimum adaptation of the dataset to the tools or vice-versa.
5. ***Reproducible***: The dataset is publicly available for replaying the test and comparing the results. Besides, this eliminates any unfair advantage in conclusions we might obtain because of self-created ontologies.
6. ***Static***: When several versions of the dataset exist, one of them must be selected and pointed out, including dumping the data if needed.
7. ***Real***: Synthetic datasets help test the strengths and weaknesses of different algorithms in a well-established set of challenges but are not valid when trying to probe the robustness of the same algorithms in a real-world situation. Frequently results tend to differ in synthetic tests, making real data sets preferred.

Some converted IFC and CityGML to ontologies provided big RDF datasets that only defined eight categories without any hierarchy. These datasets were dismissed from the testing. For a complex building, such as the Bergeron and Arboleda models, the reconstruction time for the graph tree is too long because of the number of existing relationships. As a result, more than one relationship may exist between objects from models' predicted matches.

Item	Dataset		IFC	IFC				CityGML				
			Size (MB)	Entities	Relationships	#RDF Triple	Conversion Time (s)	Size (MB)	Entities	Relationships	#RDF Triple	Conversion Time (s)
1	F2K Hauz		7.02	433	273	8536782	372	19.9	212	120	212022	212
2	FJK Hauz		13.9	274	451	9268537	286	16.0	277	276	126200	272
3	Bien-Zenker Jasmin-Sun											
4	ALLPLAN Institute		4.58	122	4.06	666176	201	0.148	98	56	250721	198
5	ALLplan Smiley West		5.83	298	580	802120	552	2.98	181	201	413300	361
6	Riverside Building Washington-DC		7.98	653	1218	440422	828	1.02	184	111	200227	622
7	Arboleda Building		274	1520	2001	2852992	1521	162	623	219	1522951	1200
8	Bergeron Engineering Building-York University		147	856	553	3564928	1320	142	289	141	1691250	1108
9	F2K Hauz		196	982	653	4828186	5489	184	301	188	2421160	3210
10	Pan-American Stadium York University		83.1	1482	3503	3103291	3258	72	623	445	2664799	220

Table 4.2 Testing datasets with computational statistics

Predicted matches are the results of the method used, while the actual matches are given by the manual instance-based method. The higher the precision, the more likely the candidates in predicted matches to be the true match. The higher the recall, the more likely this method can find accurate matches.

4.6.1 Experimental Environment and Software Tools

The server machine used for the experimentation has these specifications:

- ***Operating System***: Windows 10 Pro – 2016 Microsoft Corporation
- ***System type***: 64-bit OS / x64-based Processor
- ***Processor***: Intel (R) Core (TM) Quad CPU Q9400 @ 2.66 GHz
- ***Memory***: 32.00 GB

The transformation and adaptation of the datasets require various ontology development environments since none meet all the requirements of the ontology matching tools. Several ontology frameworks and tools have been developed, but based on their popularity, robustness, and functionality, we have chosen the following:

- ***TopBraid Composer Maestro Edition version 5.0.0 (TopQuadrant)***: A commercial environment for ontology development. It includes a robust XSD to OWL importer that transforms the XSD schemas into OWL ontologies by importing the required ontologies automatically.
- ***Protégé version 5.1.0 (Stanford University – Center for Biomedical Informatics Research)***: An open-source environment for ontology development that

facilitates the edition and management of the ontologies. It is used to inspect and complete the ontologies.

- ***NeOn Toolkit version 2.5.2***: An open-source framework for ontology development that includes special plug-ins for ontology matching. This framework results from the NeOn Project of the FP6 EU Research programme that exports the alignments in the same format specified by the Alignment API.
- ***BIMserver version 1.5.63***: It is open and stable software to build BIM software tools. It provides features to compare, query, model checking, and merge BIM models. BIMserver has many open interfaces and network protocols (SOAP, PB, JSON, and RDF) and is built as a plugin framework with a flexible admin configuration GUI.

After evaluating many tools such as Solibri, BIMCollab and others, The BIMServer has been selected, the BIM Server toolsets provides many effective tools to extract data and configuration setting that allows easy integration with other systems offering possibilities to share the BIM-GIS integrated model through rich JSON APIs libraries, and with the new offered BIMCloud platform, it certainly opened the door to cloud implementation and to wider integrations such the one with smart city platform.

4.6.2 Results and Evaluation

The F-Measures were calculated under the similarity threshold taken in the interval [0,1], with 0.1 as the increment value in all the experiments. Figure 4.3 shows each alignment technique's optimal precision, recall, and F-measure. CityGML and IFC are large ontologies with complex structures, including objects and different relationships between and within these objects. They also have a high level of structural dissimilarities. This research focused on the behaviour of different ontology matching and evaluating the graph alignment using Matching systems.

Item	Model/Sub-Model	SPSM			BSM			AROMA			S-Match		
		Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure
1	F2K Hauz	0.0031	0.0025	0.029	0.002	0.002	0..001	0.35	0.057	0.01	0.0016	0.002	0.02
2	FJK Hauz	0.003	0.0026	0.029	0.001	0.002	0..001	0.33	0.06	0.011	0.0014	0.002	0.003
3	Bien-Zenker Jasmin-Sun	0.0031	0.0026	0.03	0.001	0.001	0..001	0.39	0.061	0.01	0.0014	0.0021	0.0027
4	ALLPLAN Institute	0.0032	0.0024	0.032	0.002	0.001	0..002	0.33	0.06	0.01	0.0012	002	0.0028
5	ALLplan Smiley West	0.003	0.0024	0.033	0.001	0.002	0..001	0.36	0.059	0.01	0.0013	0.0019	0.003
6	Riverside Building Washington-DC	0.0032	0.0022	0.03	0.002	0.002	0..001	0.35	0.06	0.01	0.012	0.002	0.0029
7	Arboleda Building	0.0029	0.0023	0.028	0.001	0.002	0..002	0.33	0.059	0.01	0.0011	0.0021	0.003
8	Bergeron Engineering Building-York University	0.0032	0.0022	0.029	0.002	0.001	0..001	0.33	0.058	0.011	0.0015	0.0022	0.0031
9	F2K Hauz	0.003	0.0023	0.029	0.002	0.002	0..002	0.34	0.06	0.01	0.0012	0.0023	0.0029
10	Pan-American Stadium York University	0.0033	0.0022	0.031	0.002	0.002	0..001	0.36	0.06	0.012	0.0013	0.0021	0.003

Table 4.3 IFC-CityGML model's alignments using OAEI 2011 Ontology matching techniques

Along with the F-measure, in the table above, the recall expresses the ability to find all relevant objects of a class in a data set, precision expresses the proportion of the objects our model says existed in the relevant class that were indeed relevant.

Significantly, MapPSO leads to values of 0.0 for precision, recall, and F-Measure. This indicates that structural similarity between IFC and CityGML could not find relevant matching entities. Because of that, it is excluded from the results graph presented in tables 4.4, 4.5, 4.6, and 4.7.

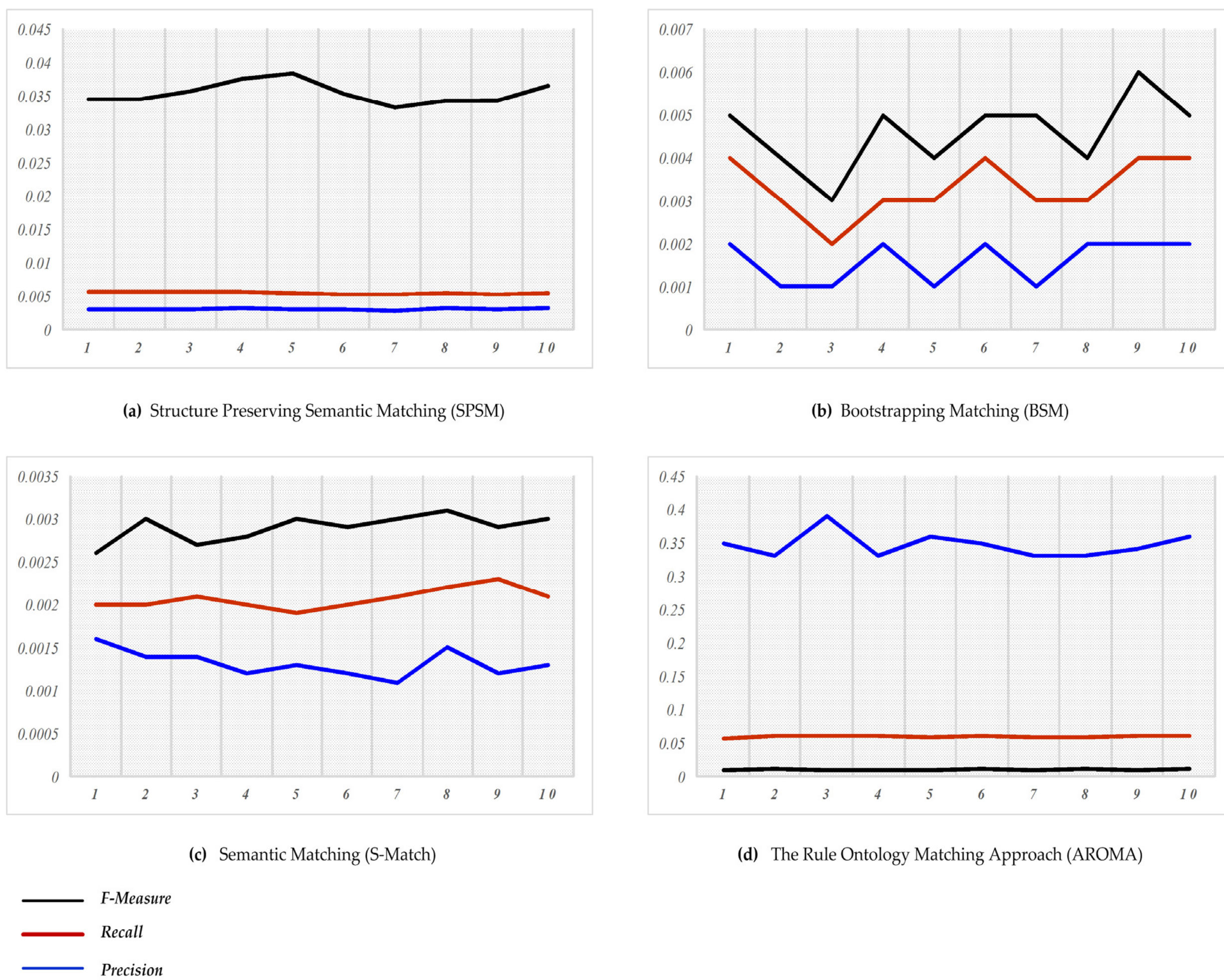


Figure 4.7 Comparative evaluation between ontologies Alignment Techniques for IFC-CityGML

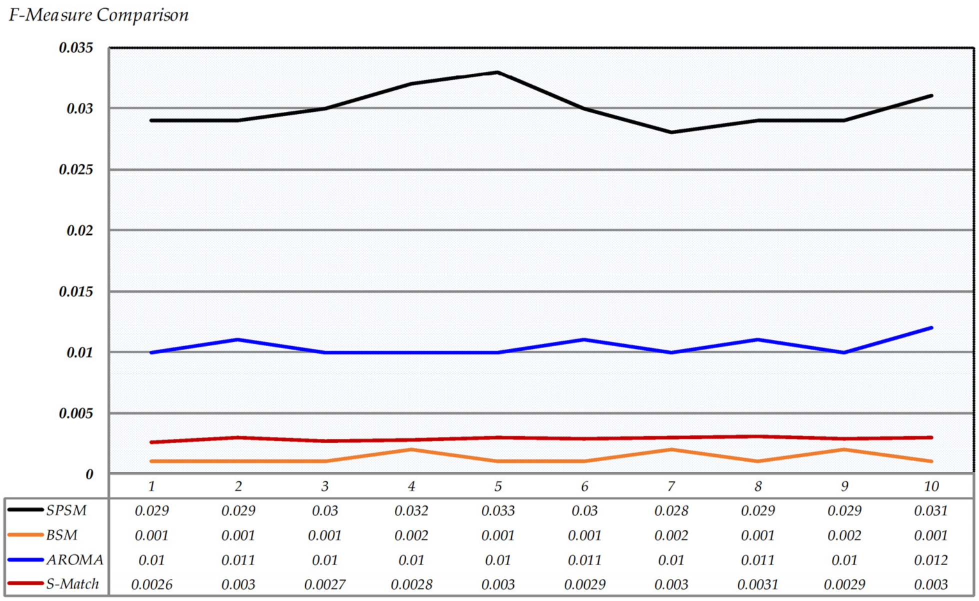
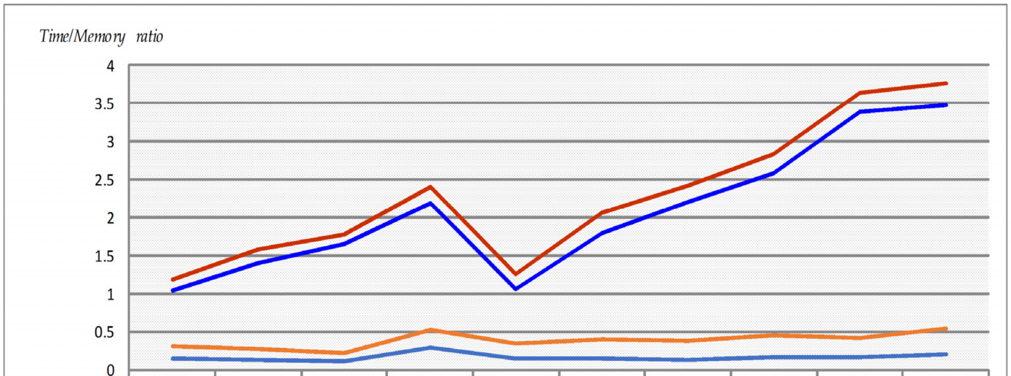


Figure 4.8 F-Measure comparison between different alignment techniques

Item	Model/Sub-Model	Time (s)					Memory (MB)				
		SPSM	BSM	AROMA	MapPSO	S-Match	SPSM	BSM	AROMA	MapPSO	S-Match
1	F2K Hauz	42	41	111	13	40	300	240	151	188	251
2	FJK Hauz	30	27	135	11	33	242	162	120	151	190
3	Bien-Zenker Jasmin-Sun	23	20	127	10	26	198	185	88	130	188
4	ALLPLAN Institute	41	40	120	9	41	140	169	72	160	171
5	ALLplan Smiley West	15	39	109	13	39	305	192	150	132	200
6	Riverside Building Washington-DC	50	44	220	12	52	322	179	156	144	200
7	Arboleda Building	48	52	300	10	45	358	210	163	108	209
8	Bergeron Engineering Building-York University	66	73	429	15	63	404	250	200	200	261
9	F2K Hauz	52	53	330	11	51	325	202	111	124	211
10	Pan-American Stadium York University	61	69	4520	10	61	300	00	153	158	208

Table 4.4 Performance (Time/Memory) evaluation of each matching technique



	1	2	3	4	5	6	7	8	9	10
S-Match	0.15	0.17	0.13	0.23	0.19	0.26	0.21	0.24	0.24	0.29
AROMA	0.73	1.125	1.44	1.66	0.72	1.41	1.84	2.14	2.97	2.94
BSM	0.17	0.16	0.1	0.23	0.2	0.24	0.24	0.29	0.26	0.34
SPSM	0.14	0.12	0.11	0.29	0.14	0.15	0.13	0.16	0.16	0.2

Figure 4.9 Performance ratio (Time/Memory) Evaluation between different Alignment techniques

GIS and BIM ontologies have many differences in their structures. This implies that the distance between their translated ontological concepts in graphs is not perfect. Both domains have been modelled and represented differently and present different points of view based on their target application and object definition. The experimental results from tables 4.4, 4.5, and 4.6 show that the AROMA ontology alignment technique has a relatively good F-measure value (high precision and high recall) compared to the other techniques despite its higher performance time/memory ratio, as shown in Table 4.7. It is noticeable that dataset (5) has almost the same performance ratio as a dataset (1). This is due to the lower correspondences between BIM and GIS ontologies for these two RDF models.

On the other hand, structural relationships are important in matching systems. For instance, *IfcWindow* (BIM) and Window (CityGML) do not share the same ontology structure, i.e., similar sub-graph or hierarchy around the classes, which supposes that the matching systems cannot establish a correspondence between the concepts. The matching is confirmed using a proposed approach that matches class, subclass, and property relationships between BIM and GIS ontologies. It is essential to know that IFC and CityGML represent many classes as object properties. To include them in the alignment process, we need to transform these object properties into classes in the resulting ontology and re-calculate the similarity matrices. These new classes or RDF triples are re-calculated into the alignment process. Another possibility is to add these classes/triples into the alignment as properties that may increase the matching process and system resources time and memory and increase the correspondence retrievals.

The results from this work are very encouraging. The proposed approach is expected to enable process integration that can improve the exchange of BIM and GIS information. The proposed approach developed in this research can be improved by overcoming a few limitations in future work. These improvements will allow better ontology matching and alignments to support BIM-GIS integration. The following are to be addressed in the future:

1. Because the RDF file contains much information about each resource's entities, translating IFC or CityGML data to RDF tends to produce substantial outputs. Keeping this huge number of triples in one big file may not be the best option to query and retrieve the data because it reads the entire file for each query. One solution is to use an RDF database management system to store, retrieve, manage and query triples.
2. Processing all data in the models is computationally intensive and time-consuming through a web portal. The ontologies are also evolving, and new concepts and

features are added to the BIM and GIS knowledge domains. Because the proposed methodology adopts ontology mapping techniques, it can overcome short-term deployment obstacles. However, the long-term effectiveness depends on developing a data framework that automatically integrates itself with the globally agreed ontologies.

3. The approach presented here demonstrates a methodology for generating RDF triples from IFC and CityGML entities from two different models. Evaluating the integrated model using the GMO algorithm to validate the accuracy of chosen entities is not fully automated. This latter has led to ongoing research on developing a fully automated translator.
4. Many matching algorithms are available now, and as the OAEI campaign indicates, no single matcher dominates others. Often, they perform well in some cases but not so well in others. It is necessary to take advantage of the best configuration of matching algorithms for both design and run-time matching. There is evidence from this research work, which is also confirmed by OAEI, that matching algorithms do not necessarily find the exact correct correspondences. By applying several competing matching algorithms to the same pair of entities from BIM and GIS, the semantic models will increase evidence of a potential match or mismatch, leading to a semantically accurate integrated model.

4.7 Summary

In this chapter, a graph ontology matching methodology is developed for integrating and validating matching results between Building Information Modelling IFC classes and the Geographic Information System CityGML elements. The proposed approach starts with developing structural similarity matrices to extract mapped elements from BIM and GIS RDF graphs and then validates the alignments between these elements by using alignment matching techniques. The proposed approach uses semantic web technologies and ontology engineering tools applied to RDF graphs to match and integrate different ontologies from different data sources and schemas. The experiment results show that the graph mapping for ontologies (GMO) algorithm and the creation of a refined structural similarity matrix play important roles in mapping triples from IFC and CityGML RDF models, validated using matching systems. Hence, this provided very representative and valid results. The quantitative comparison between alignment matching techniques by evaluating their respective precision, recall, and F-measure suggests that refinement and adaptation of matching techniques are unavoidable to improve the results. This leads to a promising research direction in ontology matching. In future work, we plan to adopt the same methodology using string-based matching results to compare results to the present work to draw new methods to integrate BIM and GIS semantic models. Also, this study has demonstrated the need in the future for applying several competing matching algorithms to the same pair of entities from BIM and GIS semantic models to increase potential matching and lead to a semantically accurate BIM-GIS integrated IGIM model.

Chapter 5

Semantic Graph Integration for BIM-GIS Information Model

Over the last decade, the semantic web platforms, with resource description framework data models, have expanded in many fields and applications, especially in what is referred to as cognitive computing. It provides a robust platform for describing information resources, simplifying semantic automation and data translation to connect heterogenous models, and creating accessible, integrated frameworks. RDF data is central in describing, retrieving, and sharing semantic information. On the other hand, graph modelling provides mechanisms to store and manipulate rich and complex data. It also offers efficient tools to express and visualize real-world data, facilitating data mining and knowledge discovery conceptually and contextually. Furthermore, RDF graph data models are powerful for simulating real-world situations for what-if scenarios that can be implemented and investigated using graph database design and analytical tools.

The graph databases are schema-less models suitable for unstructured and semi-structured data. They are characterized by efficient data retrieval and storage, allowing fast filtering of graph datasets that includes nodes, relationships, and properties. The graph database underlying the management system also includes a set of Extraction, transformation, and loading tools (ETLs), dynamic analytics algorithms, and enhanced information pattern recognition. Graph database supports Atomicity, Consistency, Isolation, and Durability (ACID) properties, transaction rollbacks, and versioning.

This chapter proposes a design architecture with implementation for IGIM on RDF graph-based database. Mainly the process would include three 3 phases—first, construction of BIM and GIS ontological models; second, semantic alignment and graph matching process. And finally, integrated model export into a graph database platform. The transformation pipelines from IFC and CityGML data from source to graph modelling are designed, evaluated then implemented in the sections below.

5.1 Introduction

As discussed in the last chapters, the integration of building information modelling (BIM) and geographic information systems (GIS) models can open the horizons for a wide number of opportunities in many areas of architecture, urban planning, facilities management, routing, and environments applications (Akicini et al., 2010; Venugopal, 2010). However, BIM and GIS came and evolved differently. Nevertheless, they can gain from others' strengths by effectively sharing their data to allow them to go above and beyond their traditional fields of applications (Bakis and Pal, 2007; Cromley and McLafferty, 2012). The BIM will bring models rich in semantics and geometries, focusing mainly on indoor environments. Then GIS will add and enhance these models' applicability to BIM surroundings and outdoor environments. Nevertheless, the design of semantic integration between BIM and GIS data will be a convoluted process given the data incompatibility and object taxonomies differences during the translation process (Hadi et al., 2016; Hor and Sohn, 2021).

The current integration between BIM and GIS enables data interoperability through a common data format between the two systems (Hor and Sohn, 2021; Karan and Irizarry, 2014). This offers users the data from the different platforms and shares it using their system's tools. It requires a good understanding of both platform tools and functionalities, not to mention that the tools do not always convey the exact meaning of data entities from the models. Hence, the need for semantic integration that takes the BIM and GIS semantic models and unifies them into a unique model by translating objects and their associated information (attributes, properties, and geometries) into an interoperable format understandable by BIM and GIS (Azhar et al., 2011; Jiang et al., 2013).

The IFC classes and CityGML elements are, respectively, the core building blocks of the BIM and GIS models. IFC and CityGML are semantically rich in data and properties describing 3D objects. This will give them a considerable advantage in using semantic web framework and RDF representation as a neutral platform for integration based on description logics principles by modelling domain terms into concepts, roles, and individuals in OWL terminology (Hadi et al., 2016; Goedert and Meadati, 2013). It corresponds to classes, relationships, and properties gathered into RDF assertions or statements. It is intuitively presented as graphs of connected nodes defining the classes linked with relationships represented as vertices. This chapter explored the possibility of using a graph database for BIM-GIS semantic integration using a unified RDF model, taking advantage of the flexibility, agility, and performance of the graph database systems and the scalability and analytical tools that come with them (Hadi et al., 2016; Hor et al., 2018; Hijazi et al., 2011).

5.2 Semantic Integration Conceptual Framework

The complete conceptual system architecture for the semantic integration based on RDF models and graph database is illustrated in figure 5.1. as shown in the diagram, the system consists of four (4) modules (Hor et al., 2018; Hor and Sohn, 2021).

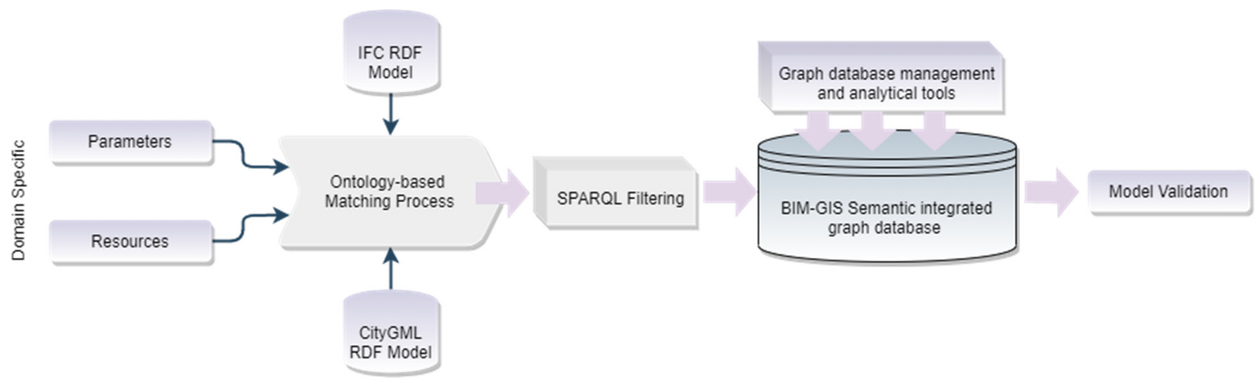


Figure 5.1 Semantic integration conceptual system design

The source BIM and GIS models are conveyed in IFC and CityGML native formats before being exported into RDF format using translation workflows (Hor et al., 2018). Subsequently, the four modules are described below:

- **Ontology-based matching module:** This module will be responsible for IFC and CityGML data translation operations using graph matching algorithm and alignment tasks prior to model export into the BIM-GIS RDF graph.
- **Integrated RDF graph model*:*** Representing data from BIM and GIS Semantic models.
- **Integrated property Database Module*:*** This represents the complete integrated BIM-GIS model, including graph databases administration, management, and development tools.

- **Model Validation Module**: The presentation tier application uses the backend BIM-GIS graph database as source data.

The new workflow proposed in this research has been designed, developed, and implemented (Hadi et al., 2016; Hor et al., 2018). The process constructs the IFC and CityGML graph models using RDF ontological models. Figure 5.2 shows the transformation steps of RDF extraction from the BIM-GIS semantic model from a triple store repository. A filtering process is illustrated, which uses SPARQL before it is imported using graph database tools into a BIM-GIS RDF graph data stored and maintained in a graph database system (Hor and Sohn, 2021; Bakis et al., 2007).

Figure 5.2 RDF transformation and loading pipeline

The procedure of loading data from RDF data into a graph database can be completed by exporting BIM-GIS integrated RDF model into a comma-separated values (.csv) file format using the import/export utility and must follow the W3C recommendation (SPARQL 1.1 Query Results CSV and TSV Formats (w3.org). The CSV file can be used for data visualizations, data mining, and knowledge discovery. The SPARQL queries are developed to pull the collection of triples selected through the matching and alignment process into a familiar tabular view (more details can be found in the chapter) (Hadi et al., 2016; Bakis and Pal, 2007).

The outputs from RDF SPARQL queries (example shown in figure 5.3) are developed and executed on the IGIM RDF graph model. The (2.2.6. Importing CSV files with Cypher - 2.2. Get started with Cypher (neo4j.com) with Cypher. This efficient Extract-Transform-Load tool converts data and directly matches source data into a detailed graph structure, supporting complex computations in merging data, relationships, and properties for large datasets. Then a data quality check task is carried out, using tools to validate the resulting dataset. There are many out of the box that can be used. Also, some custom tools can be scripted to accomplish data validation. In this study, the CSVkit python-based (csvkit 1.0.3 — csvkit 1.0.3 documentation) provides data statistics and reports that help to find inaccuracies effectively. We also used LOAD CSV from Cypher, a powerful ETL/import tool. In addition to converting data, it also permits direct mapping of input data into complex graph/domain structure, supports complex computations, and merges data, relationships, and structure on large data (Bakis and Pal, 2007; Tserng et al., 2009).

```
SELECT ?column ?window ?d
WHERE
  {
    ?column a ifc:ifccolumn .
    ?column qrw:isContainedIn ?storey .
    ?window qrw:isContainedIn ?Storey .
    ?window a ifc:ifcWindow .
     (?column, ?window) spy:distanceXY ?d .
    FILTER (?d<250)
  }
```

Figure 5.3 RDF SPARQL Query to verify the clearance before windows by finding the closest columns to windows

5.3 RDF Triple-Store Versus Labelled Property Graphs

The RDF triplestores and property graphs have dynamic and flexible ways to store data as a network of nodes with materialized links between them to explore, graphically depict linked data, and query integrated models (Hor and Sohn, 2021; Cutting-Decelle et al., 2007). Although the two representative models have a resemblance, there are some differences. However, they both are powerful and robust for representing BIM and GIS data. Therefore, this research combines aspects of the RDF and property graphs in designing and developing a graph data model that brings well-defined triplestore-managed data to the formalism of a graph model. This will produce a new rich data model with the highest level of accuracy, equipped with a large set of tools to manipulate and manage graph data. However, it is important to highlight the following concerning RDF graphs (Hor and Sohn, 2021):

- Relationship instances of the same type are not uniquely identified in an RDF graph.
- There are no instances of relationships (since there are no attributes).
- The volume of triples in a property graph that had ***N*** objects **(**nodes), with ***p*** properties for each object (per graph node), r relationships (per graph node), and ***l*** labels (per graph node), the complete RDF graph will encompass: $(p + r + l) * N$ triples.

Graph database modelling provides an efficient and flexible way of data storage, manipulation, and filtering focused on values, nodes, and relationships. It extracts information and pattern within any data set using tools and analytics methods based on graph theory. In the case of IFC and CityGML models, this means that we can capture and represent any element in the BIM along with any environmental surroundings or geospatial layers in one graph (Hor et al., 2018; Hor and Sohn, 2021; Crowther and Hartnett, 2001).

5.3.1 Graph Databases Models

Mathematically, many graph models can be effectively used to design and develop BIM-GIS integrated graph models. However, the best of these models is the property graph model, a directed multi-graph, attributed and labelled, with much richer representational information and visualization capabilities, and very efficiently deals with complex data models (Hadi et al., 2016; Karan and Irizarry, 2014). A graph database based on a directed graph model will provide an optimized processing system to refine heavy and highly interconnected dynamic datasets with a large volume of information from the dataset, its schema, or metadata (Hor and Sohn, 2021). This database graph model permits the construction of models using artificial intelligence, machine learning methodologies (adaptive and predictive models), and traditional models. Furthermore, connecting all the nodes, their associated properties, and values from both IFC and CityGML will provide fast localized transversals among the data, eliminating unrelated data and producing the most accurate integrated graph model possible (Choi et al., 2008).

Between 2004 and 2014, there have been about 80 graph processing systems were introduced to respond to the increasing demand for graph modelling and data analytics in many fields of ion science and engineering (Liu and Issa, 2012). These systems can be categorized into two major groups: graph database systems and graph processing systems (Choi et al., 2008; Goodchild et al., 1999). Since one of our main target goals from this research is to build a robust graph framework for our BIM-GIS integrated model, this study has looked at bringing the benefits of these two systems together, such as:

- Graphs provide ways to enhance data in any given model. Therefore, more information in data, metadata, and attribute or properties can always be added.
- Graph databases have built-in data storage structures for graphs.
- Graph databases possess many graph-oriented algorithms to handle graph-specific operations (constructs, procedures, functions, etc.)
- Filtering operations are based on graph type and structure, and the result of querying a graph is a sub-graph of nodes and vertices with the same structure.

Many studies have shown that graph database effectively process dense, interrelated datasets because the graphs emphasize exploring relationships between objects and entities (Hadi et al., 2016; Liu and Issa, 2012). The graph structure and design allow navigating and filtering through correlations and patterns, which is essential when integrating two different data models such as BIM and GIS. This creates a dynamic graph model that connects nodes and relationships for graph traversal fast search (tree transversal) along the edges between vertices. Since relationships and properties play an important part in IFC and CityGML data structures, it is beneficial to use graphs in building and exploring the BIM-GIS data model. In the following sections, we looked at current graph databases, testing, and evaluations for the most suitable system for our BIM-GIS integration (Hor and Sohn, 2021). The complete features and functionalities comparative study was performed to based on the six factors below:

1. Modelling capabilities.
2. APIs and analytical functions.
3. Performance and scalability.
4. Enterprise features.
5. Interoperability with other data types.
6. Integration tools.

The evaluation procedure will assess each graph data model's suitability for graph structure and type, querying language support and constraints. Our list of systems has been short-listed based on the factors above for the following systems (Hadi et al., 2016; Hor et al., 2018; Hor and Sohn, 2021):

- **AllegroGraph**: A high-performance graph database system with an extensive storage capability, it supports SPARQL, RDFS++, and REST APIs, which make it very suitable for time-series reasoning and geo-temporal data modelling
- **DEX**: High scalable and performance graph database. It is used mainly for NoSQL-based integrations and applications. DEX is based on a three-tiered architecture (Core, API, and application tier). Each tier can be extendable to provide powerful features. For instance, the application layer enhances the core layer with additional custom functionalities and uses the APIs for interfacing with more systems.
- **VertexDB**: Very efficient with web-based communication protocols and data formats such as HTTP and JSON. VertexDB server supports well memory management and garbage collection, which is very critical to processing large graph database objects.

- ***Infinitegraph*** is a highly scalable object-oriented and distributed database built on a four-tiered architecture. The main components are management and configuration and APIs tiers. Infinitegraph is the best graph database system to process enormous interconnected and complex datasets.
- ***Hypergraphdb***: This is one of the most dominant graph databases nowadays because of its modelling capabilities, graph-oriented manageable storage, relational-style queries, customizable indexing algorithms, and other powerful features. HypergraphDB is widely used for Artificial intelligence applications and considerable domain knowledge and semantic web representations.
- ***Sones***: This is the fastest, most scalable graph database, used by architects, data scientists, and engineers for its robust platform for parallel processing, database graph objects (nodes and edges), serializations/deserialization, and data abstraction capabilities. This graph database system can perform deep link analytics and high-performance complex queries.
- ***Infogrid***: This is an open-source, fully web-oriented graph database with a complete set of tools to support REST APIs, accessible to interface with any system or web-based application. Its capabilities allow rendering objects and sub-graphs for easy use and practical integration.
- ***Trinity*****:** This is a distributed graph database with an abroad set of tools to process enormous graphs, management, and configuration utilities to meet the needs of large datasets. Trinity computation platform modules possess graph stores and queries for online processing.

- ***OrientDB***: This is a NoSQL Open-source transactional graph database management system based on Java. This document-oriented database supports all schemas characterized by a robust security system built on a user-role setup. OrientDB uses the MVRB-Tree indexing algorithm, which is very efficient for fast insertions and ultra-fast lookups with APIs and supports HTTP and RESTful protocols.
- **G-*Store***: This is a vertex-labelled graph database. It exploits graph structure to extract data on an optimized desk for fast access and execution of graph queries; with its built-in query engine, G-Store supports the shortest path, shortest path tree search, and many other features.
- ***Cloudgraph***: This transactional graph database based on the .Net framework provides scalable and fast data retrieval capabilities. Cloudgraph is easy to use to maintain and manage a graph database. It uses graph query language (GQL), supporting unstructured schema-less web data and hypergraphs.
- ***Bigdata***: This is a high-performance graph database for distributed computing data-intensive processing. It is characterized by high concurrency, I/O rates, and a high-level query. It also provides SPARQL query language for fast load and query.
- ***Neo4j***: This is an open-source No-SQL-based graph database, ACID-compliant with high-performance designed java and python applications. Neo4J uses Cypher query language over HTTP and REST endpoints. In a Neo4j graph model, edges can be directed or undirected, associated with a defined type, and the object properties can be included in the node and edge. The Neo4j database system uses both single or composite property indexing, a storage manager for disk, and object-oriented APIs,

making the Neo4j graph model the best candidate for a BIM-GIS integration framework given that it is (Hor et al., 2018; Hor and Sohn, 2021; Karan and Irizarry, 2014):

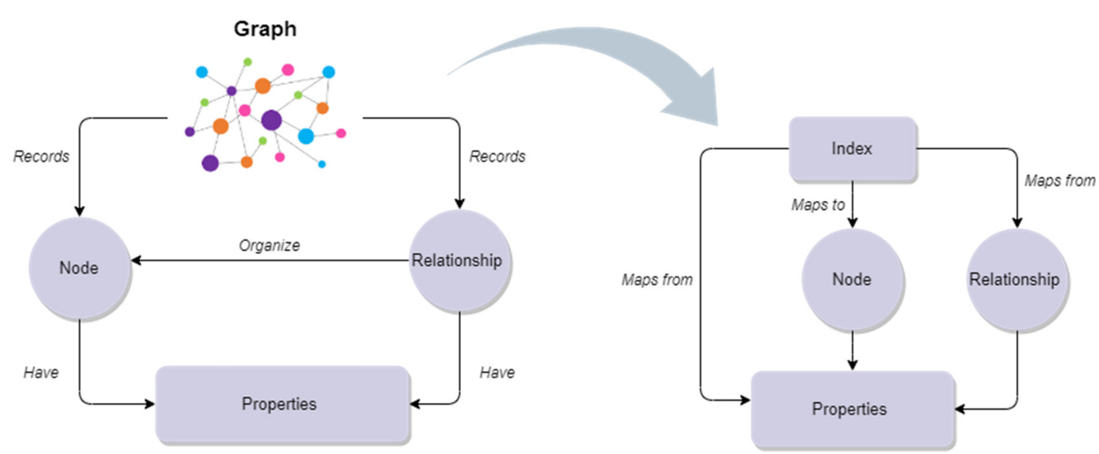
- **Intuitively reliable**: Simple graph representation with transactional and ACID properties support.
- **Embeddable and durable**: Using a native engine for graph storage
- **Highly availability and scalability**: The ability to process large data across distributed computing resources.
- **Accessibility, speed, and articulation**: its rich REST APIs, powerful graph language, and efficient graph indexing algorithms for fast query execution.
- **Cross-platform**: runs on all existing supported computing platforms, including the cloud.
- **Cypher query language**: Easy to design high-performance queries using an easy and well-documented query language.

In this research, Neo4j was adopted as the graph database management system to store, manage, retrieve, and process IFC and CityGML RDF-based graph data. The SPARQL and Cypher graph queries are designed to avoid and ignore graph data related to a node, vertices, or properties that are not connected to any other node in the target graph. This will produce a clean, rich, and accurate graph representing BIM and GIS triples conforming to the search criteria in the query constructs (Hadi et al., 2016; Hor et al., 2018; Hor and Sohn, 2021).

Like any other graph database model, Neo4j uses:

- **Property**: This is a set of data and information associated with nodes or a relationship in the graph model to define a domain's purpose, meaning, or value.
- **Node**: It is at the core of the graph model. Every node has several properties (labels) associated with it. Nodes are connected to other nodes in a graph using relationships and can be indexed for quick search or filtering tasks.
- **Relationship**: Nodes are interconnected through relationships within the same graph model. Each relationship has an origin node and a target node representing the information flow within the graph. In some cases, these relationships are unidirectional or bidirectional and can also have properties.

The most basic graph can be a single node graph with a given name and a property. In graph integration, a single node can acquire more attributes from similar nodes with similar definitions, the same function, or a specific common property, as illustrated in figure 5.4 below (Hadi et al., 2016).



.

Figure 5.4 Graph node and relationship

In addition to nodes, relationships, and property-defining objects in a graph model, and for the sake of query performance purposes, the Neo4j graph database management system maintains indices and constraints (Hor and Sohn, 2021; Bakis et al., 2007):

- ***Index***: Neo4j graph database creates a redundant copy of the database called index to search for graph entities (nodes, relationships, properties) more efficiently. Moreover, choosing what data to index (or not to index) is critical since that affects storage space and read/write operation in the graph database (Hor and Sohn, 2021; Isikdag et al., 2008). Therefore, the index type must be carefully chosen, whether a single-property or composite and applied solely for specific queries depending on the filtering operations or data modification.
- ***Constraint***: It is essential to use constraints on nodes or relationships in the graph to guarantee the object's existence and uniqueness to enforce graph data integrity. This is critical in the integration process in ensuring that the object wanted is precisely the object retuned by the filtering criteria (Hadi et al., 2016).

5.4 BIM-GIS Integrated RDF-Graph Database

5.4.1 5.IFC RDF-Graph Model

The BIM Industry Foundations Classes (IFC) are an object-based dataset designed to accurately describe and present architectural, Engineering, building, and construction data (Agdas and Ellis, 2010; Karimi and Akicini, 2014). IFCs are developed to expedite and promote interoperability between software platforms during BIM-related projects (Karan and Ardeshir, 2008). The data represented in IFC files provide a rich information model regarding components, relationships, and geometries (Hor et al., 2018; Brunnermeier and Martin, 2002). Recently, several research projects in industry and academia have proven that graph databases are efficiently suited to save, manage and visualize IFC objects given the graph hierarchy structure of the IFC data model. Each IFC class is composed of an IFC's main attributes, with defined properties as specified in an IFC schema standard code. They are also distinguished by their entity types in which the object refers to a generalized item (Bakis and Pal, 2007; Sebt et al., 2008) and the relation illustrated relationships between items. Finally, the properties qualify the characteristics of the object's type and sub-types assigned to IFC objects in the models. As shown in figure 5.5, an IFC schema class is represented as:

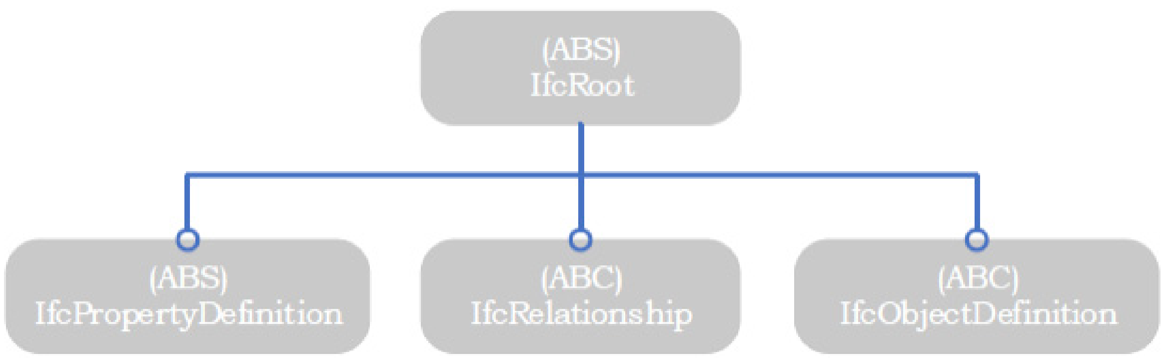


Figure 5.5 Entity types derived from IfcRoot class

- **ifcPropertydefinition**: including all aspects attached to an object; this information is common among other instances of the object in the schema and indicates the occurrences in the same BIM project.

- **ifcObjectDefinition**: represents handled processes in the context of the BIM projects; it can also refer to objects. i.e., items like slabs, doors, or others can be classified as ifcObjectDefinition.
- **ifcRelationship**: used to save properties specific to relationships and connectivity information among the objects of the same model. Each relationship could have multiple properties attached to it, as illustrated in figure 5.6

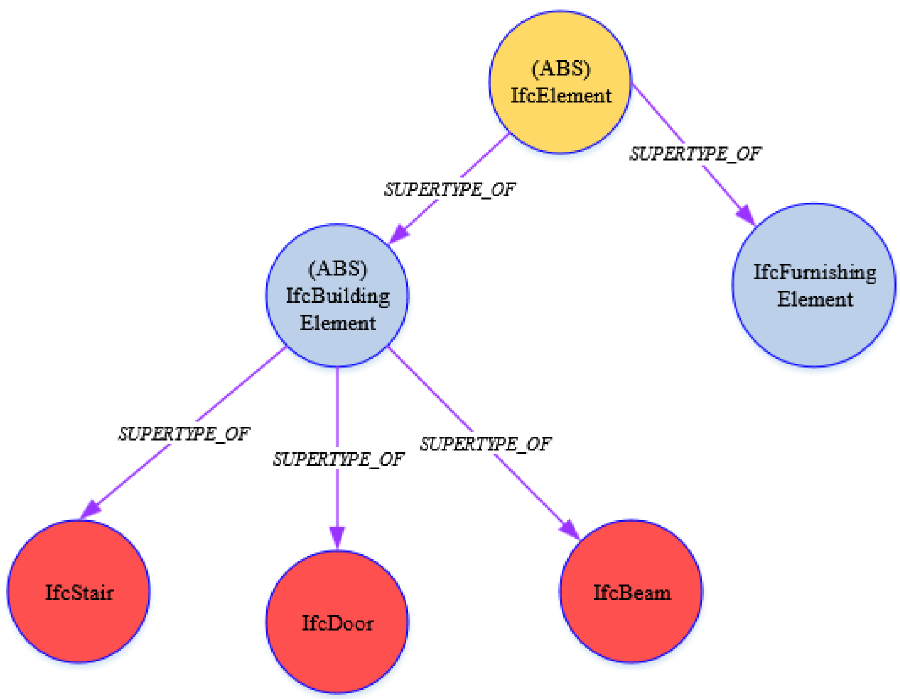


Figure 5.6 IFC Meta Graph Model diagram

The BIM Models go beyond not just the design phase of the building projects, and it covers all the phases of the building life cycle (Azhar et al., 2011; Balachandar et al., 2013). Therefore, these models contain enormous information on building objects, properties, and relationships. To unveil model patterns and explore and analyze information from the BIM model, it is necessary to execute a series of data extraction of IFC models/schemas into IFC

RDF-based object graph database through automatic workflows (Hadi et al., 2016; Hor and Sohn, 2021). These workflows will be based on graph theory methodologies to discover, analyze, and manage the BIM as a unique model by:

- Design queries for data mining and information retrieval.
- Create topology and geometrical analysis methods on BIM models to uncover patterns in the datasets for integration or connectivity to supplementary models related to the application domains, such as adding Mechanical, Electrical, and plumbing models.

The RDF models are triple-based data composed of resources of subjects and predicates. The objects in the RDF model can be just another resource or uniquely identified literals (nodes) (Bakis and Pal, 2007) that cannot be the subject of another triple in the model. For a successful mapping of the RDF graph triples to graph database objects (Hadi et al., 2016), we will need to setup the following rules:

- **Rule 1**: Each RDF graph node is labelled a resource in the model and will be mapped to an object in the graph database.
- **Rule 2**: Subjects in an RDF graph connect node to node.
- **Rule 3**: if the triple object is literal, predicates are mapped to properties in the graph database.
- **Rule 4**: if the triple is a resource, the predicates are mapped to the relationship in the graph database.

These rules will help enforce graph modelling methodologies and best practices to produce a rule-normalized RDF model inheriting the native EXPRESS-BASED IFC structural model (Bakis et al., 2007) (Irizarry et al., 2013). This new IFC/RDF model will be one source model to construct the BIM-GIS integrated information model. Therefore, it must be able to provide detailed information about any objects, their attributes, or geometrical information to applications using a graph database as a data source (Karan and Irizarry, 2014). All query results and analytical workflows output results containing sub-sets of IFC elements from the integrated model can be drawn back from the source models. The interconnection can be traced back from the relationships within the models and extended to the building's environmental surroundings from CityGML/RDF elements, including in the query construct formalism (Hor and Sohn, 2021).

5.4.2 CityGML RDF-Graph Model

The Geographic Information System's City Markup Language is OGC (The Home of Location Technology Innovation and Collaboration | OGC) compliant XML-based data format to exchange, store, and represent a 3D digital city model. The CityGML is the perfect data model for geodata integration for many systems and applications in geo-design, smart city, and urban digital twins. Furthermore, natural and man-made features like water bodies, terrain, bridges, and buildings can be defined and digitally described semantically, topologically, and geometrically using CityGML elements and at different levels of detail (LOD0 to LOD4) (Agdas and Ellis, 2010; Peña-Mora et al., 2010).

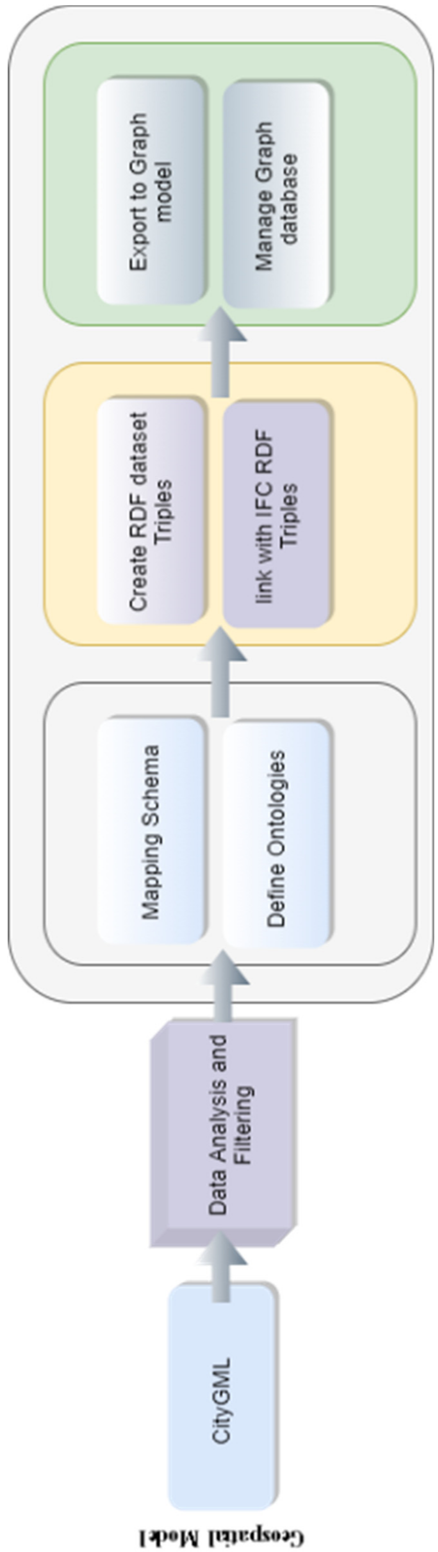


Figure 5.7 CityGML translation process

Figure 5.7 describes the complete conceptual translation process of the GIS model from native CityGML data into an RDF graph model (Hor and Sohn, 2021). The transformation pipeline is comprised of:

- Data analysis and filtering process: in this stage, the CityGML file is analyzed for suitability to the target application by ensuring that all the GIS elements needed are present with all attributes and properties. This information can be extracted using a filtering process.
- Mapping schema and defining ontologies: This stage needs a deep understanding of the target application to ensure that the existing CityGML elements with the model are available and have the required properties, attributes, and relationships to be translated to an RDF graph.
- Create RDF dataset triples and linkage to IFC RDF triples: Once all the CityGML are identified, the translation to RDF triples task is executed to reference to IFC RDF based on equivalency.
- Export graph model and managing graph database: at this stage, the sub-model is exported from an RDF graph to a fully managed RDF graph database.

It is essential to consider developing semantic queries using a graph query language to complete the processes highlighted above and perform Extraction, transform and load (ETLs) triples into the integrated model during the stages. We used RDF SPARQL query language to design data semantic properties related queries (Hor and Sohn, 2021; Beetz, 2009). However, GeoSPARQL (GeoSPARQL Ontology (opengeospatial.github.io) was implemented instead to perform processing on geospatial features with geometries and geospatial relationships between them. It defines vocabularies representing data in RDF format. For example, we can retrieve all the objects between IFC and CityGML RDF models on a specific feature type called ITEM, by embedding a simple clause from GeoSPARQL: **ITEM:hasExactGeometry** (Bakis et al., 2007)**.**

As discussed earlier in the previous chapter, the linkage process between IFC and CityGML RDF triples can be defined after matching the schema before exporting to a graph-based model (Borrmann et al., 2006).

5.4.3 IFC- CityGML RDF-Graph Model

The graph modelling of real-world scenarios has become a powerful and efficient technique. It can be applied to various fields, from architecture and engineering to medical applications. It was instrumental in describing and understanding complex patterns in data models. The advancement in graph theory and graph database systems has led to several graph modelling types with some differences and similarities in their object definitions and topologies (Hor and Sohn, 2021; Studer et al., 2007). The labelled property graph model is used most to describe and present complex data. Moreover, it consists of nodes and relationships (graphically referred to as vertices and edges) and is characterized by:

- Nodes (vertices): Can accept multiple labels and key-values pairs attributes.
- Relationships (edges): Can contain properties (and values) and connecting nodes (vertices) at the same time.

These properties are well suited to the BIM-GIS RDF graph data modelling vision and will link all the semantics and attribute information to the RDF integrated model. It will make it possible to use SPARQL queries more efficiently to extract and find specific data or information from the model.

The high-level conceptual workflow is shown in figure 5.8. The workflows contain an extraction of sub-model data into a comma-separated value file (.csv file), followed by a migration process into graph data to a single graph database system storing information from both IFC and CityGML. Each object in the graph database will carry attributes from GIS or the BIM, mostly from both models (Hadi et al., 2016; Hor and Sohn, 2021; Hu et al., 2005).

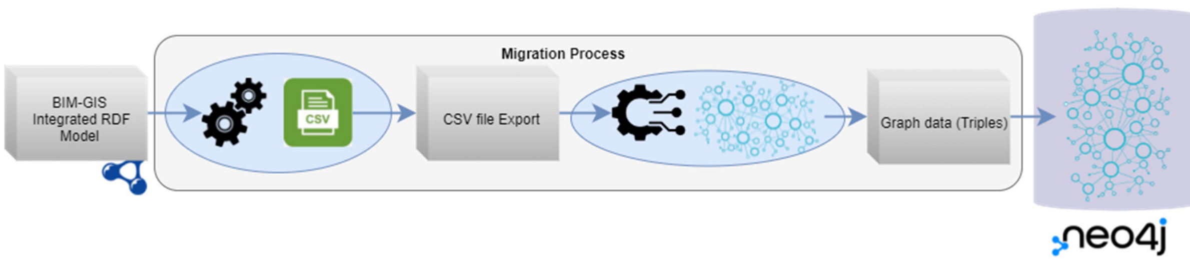


Figure 5.8 Graph database (Neo4j) conversion workflow

The graph database semantic analytics will not be strictly applied to one or the other source models. Still, these tools are applied to (nodes, relationships, and properties) from GIS and BIM together. This will open widely the BIM-GIS semantic integrated into all the RDF framework technologies tools (Hadi et al., 2016; Hor and Sohn, 2021; Hijazi et al., 2011), not for the merging objects and creating a new unified object but also to design and develop a new adaptive and predictive set of tools and intelligent analytics algorithms for data mining and discovery taking advantages of the semantic technologies and ontology engineering. A new BIM-GIS semantic Integration infrastructure is developed based on pre-defined vocabularies and relationships from the integrated ontological model within one unique semantically interconnected framework, as shown in figure 5.9.

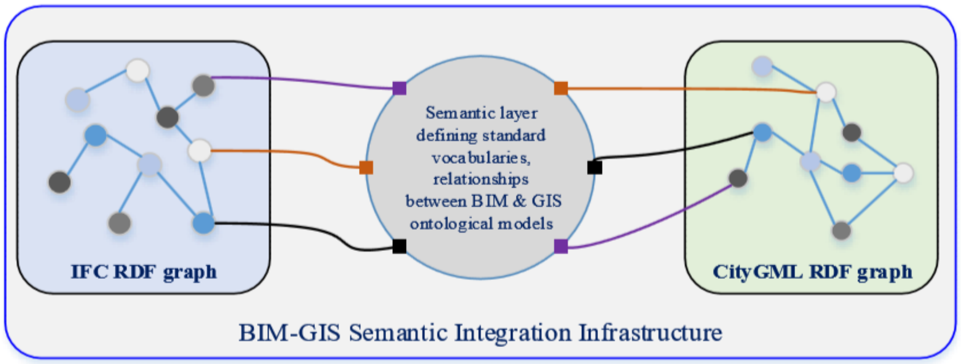


Figure 5.9 IGIM integration infrastructure (ontological model)

The resulting integrated graph model imported into the graph database will have new objects enhanced with each other's attributes and topological and geometrical properties as described for the *ifcSlab* from BIM and Slab from GIS, and can be seen in figure 5.10.

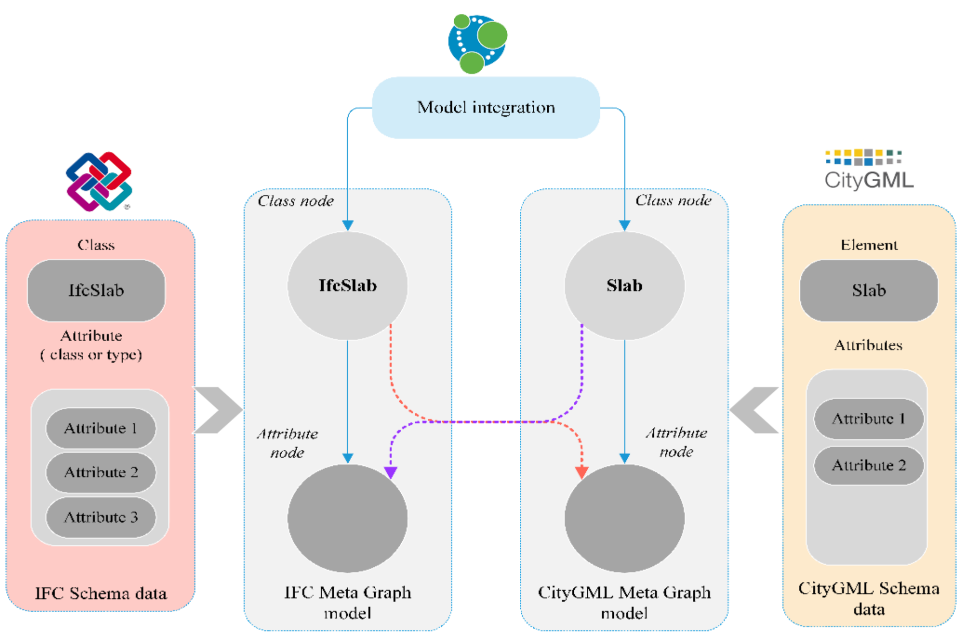


Figure 5.10 BIM-GIS information Model with enhanced objects
(Stored in Graph database)

All the BIM-GIS semantic integrated model objects stored in the graph database will be uniquely identified global ID (GUID). This is key to graph join operations through a conjuncture of attribute properties, object linkage, and, most importantly, to preserve the data model integrity during ETL operations or sub-model extraction (figure 5.11).

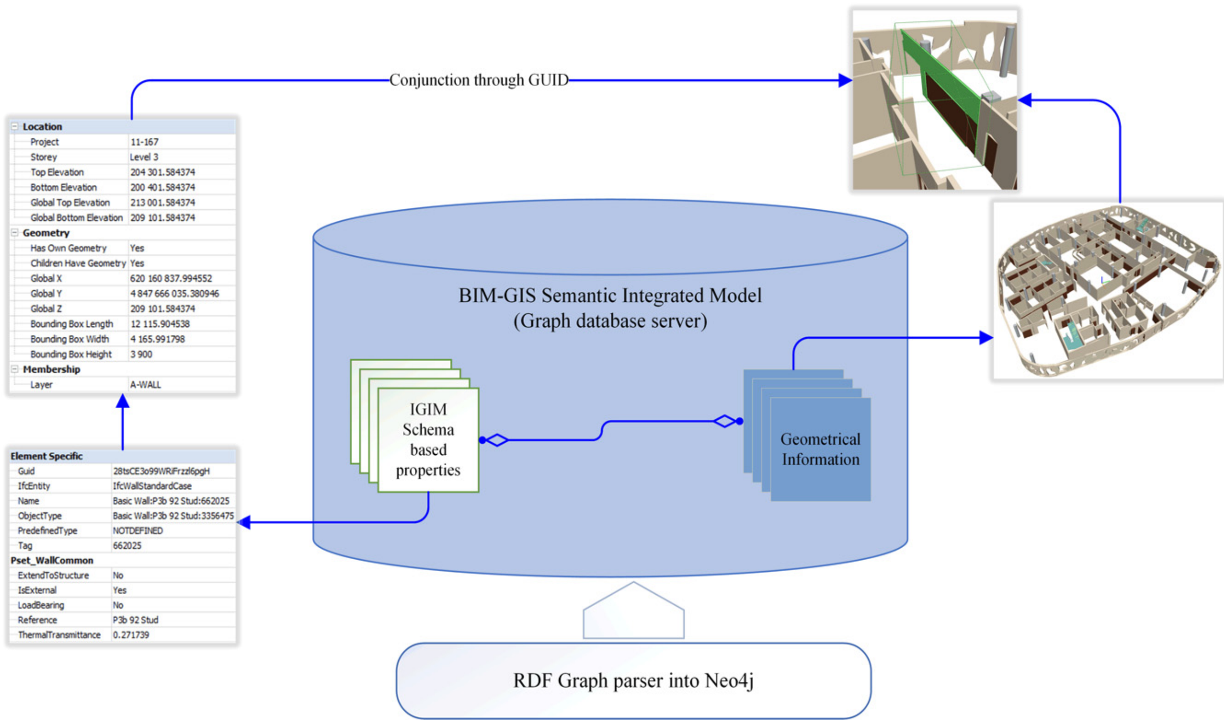


Figure 5.11 IGIM model object linkage Between properties and associated geometries

A meta-graph model of connected IFC and CityGML objects, including their attributes, properties, and relationships, is constructed by loading single or multiple CSV files into the graph database. The objects are mapped into defined nodes and interconnected to other nodes using the relationships depending on the integration criteria. For example, the has_property relationship connects a specific class with another node through the common property. The ***subtype_of*** designed make a correspondence between a CityGML element and its sub-class in the model hierarchy (Hadi et al., 2016; Hu et al., 2005).

The complete BIM-GIS integrated graph model will consist of stored objects (derived from IFC and CityGML or both) with defined relationships between graph nodes for referenced, inversed, and derived attributes created automatically from IFC and CityGML. The model will be driven by geospatial relationships from CityGML elements with no redundant relationships (Hor and Sohn, 2021; Hu et al., 2005).

The model normalization requires:

- IFC and CityGML Non-direct attributes of the same object are to be assigned as direct node attributes.
- Each object will be assigned a set of labels from a parent class.
- Queries are run to classify and normalize objects following their object type and property sets.

```
load csv with headers from 'E:\YU-Hadi\BIM-GIS-SemanticIntegration\
YorkU-BergeronEngineering_IGIM.ifc\IfcSlab.csv'
as line FIELDTERMINATOR '	'
create (u:IfcRoot:IfcObjectDefinition:IfcObject:
IfcProduct:IfcElement:IfcBuildingElement:IfcSlab
{
    model:line.model,
    ifcid:line.ifcid,
    ifcClass:line.ifcClass,
    globalId:line.globalId,
    ownerHistory:line.ownerHistory,
    name:line.name,
    description:line.description,
    objectType:line.objectType,
    objectPlacement:line.objectPlacement,
    representation:line.representation,
    tag:line.tag,
    predefinedType:line.predefinedType,
    model_id:line.model + '_' + line.ifcid
});
```

Figure 5.12 Neo4j Cypher script to load CSV files into the graph database

In this research project, we used Neo4J Cypher query language to interact, query, update, and administer the BIM-GIS semantic integrated model. Cypher provides tools and commands to work with the most complicated graph models (Hor and Sohn, 2021).

It is characterized by:

- Simple syntax.
- Declarative grammar.
- Human friendly and self-explanatory.
- Written in Scala programming language in combining oriented object and functional programming.
- Pattern matching with the ability to borrow expression approaches from SPARQL
- Uses aggregation functions, sorting, and clustering.
- Structured and inspired from DBMS -SQL Migration cycle very quickly.

Many keywords like MATCH, RETURN, and WHERE are used to construct complex queries. Figure 5.13 illustrates some examples of replacing attributes and relations in a BIM class:

```
MATCH (n:IfcPresentationLayerAssignment{Model: "IGIM-YU-Bergeron_A.ifc"})
UNWIND split(replace(replace(n.assignedItems,"(",""),")",""),",") as o
MERGE (assignedItems {Model: "IGIM-YU-Bergeron_A.ifc",IFCID: replace(o,"#","")})
MERGE (n)-[:assignedItems]->(assignedItems);
```

```
MATCH (A { Model: 'Model-A.ifc' }),(C { Model: 'Model-C.ifc' })
WHERE A.tag = C.tag WITH collect(distinct A.tag) as tags
MATCH (new) WHERE new.tag <> "" AND NOT new.tag IN tags
AND new.label = "IfcWindow"
        RETURN distinct new.label as Class,
                new.globalId,
                new.IFCID as STEPID,
                new.name,
                new.tag
    ORDER BY new.tag
```

```
MATCH (door:IfcDoor{Model:'IGIM-YU-Bergeron_A.ifc"'})-[*1..5]-(material:IfcMaterial)
MERGE (n)-[:assignedMaterial]->(assignedMaterial);
RETURN DISTINCT (material.IFCID)AS IFCID,(material.name)AS Material_name
```

Figure 5.13 MATCH-MERGE-RETURN operations on a building element

5.5 Summary

This chapter presented workflows for data translation from a semantic web RDF format model into an RDF graph-based database model. The new model shows that using graph methods to discover and explore graphically is efficient. The filtering mechanism of data and patterns depicts connected and related datasets, relationships, and values between triples. We have applied this approach to multiple IFC and CityGML schema/models, and the results have shown great possibilities for advanced data analysis for BIM-GIS integrated models.

Our approach has also considered the geometry information and the process of creating geometry objects based on the linkage between properties and associated geometries presented and explained in figure 5.11. In the next chapter, we automate the linkage process within the integration workflows, even from data sources other than GIS and BIM. This can be achieved by designing an interface between the graph database and a new BIM-GIS integrated model geometry engine.

Future development using our IGIM semantic integrated model using RDF graphs is termed Integrated Geospatial Information Model (IGIM). It will include designing and developing Cypher/Java-based procedures to automate data retrieval, schema extraction, and accessing Neo4j APIs to create an extension to support other data sources that can be brought into the model.

Finally, it is important to mention that designing queries against the IGIM semantic graph model is challenging. It requires BIM, GIS, and graph skills and should be written by BIM and GIS expert partitioners. This project is a significant first step in this direction for professionals, experts, and academics in the AEC and GIS fields.

Chapter 6

System Integration, Validation and Performance Evaluation

6.1 Introduction

Previous chapters indicate that the RDF graphs have become an increasingly crucial academic research area and attract many industry practices. Nowadays. Many systems use RDF representations to express various resources' complex information models and data associations simultaneously. The role of RDF has become paramount to investigating knowledge discovery patterns and many ML, AI, adopts it, and DL research works. However, RDF can be facilitated only if implemented with efficient tools from the graph database system to increase and enhance BIM-GIS integration and interoperability within or from new models.

This chapter used a graph database system to implement and evaluate our IGIM model. The data translation workflows and queries were designed and executed against the IGIM to evaluate integration data richness, matching accuracy, and IGIM overall performance. The new occupancy indexes for nodes, relationships, and properties are calculated and compared to provide insights into data and metadata differences between testing models used in this study.

The chapter will present the five phases of designing and implementing the IGIM model using the RDF graph database. The phases would include (1) BIM and GIS semantic models' construction, (2) semantic matching and integration alignment using interoperable representation formats, (3) Extraction, filtering, and import of data into a graph database system, (4) Design of transformation workflows and ETL pipelines, and (5) create a domain-oriented application using a web-based scene service from IGIM model. This application will validate the IGIM model; it simulates intelligent urban mobility designed on a semantic web data framework and a game engine platform.

6.2 System Architecture and Functional Components

6.2.1 Workflows Conceptual Design

The complete end-to-end IGIM semantic integrated system architecture is presented in figure 6.1. It compromises four phases, each representing the process workflows conducted during the phase design and development (Hor and Sohn, 2021); it begins with : (1) Input data IFC and CityGML models, (2) Data transformation to RDF data format (Hor et al., 2018; Hor and Sohn, 2021), in this phase the data translated from a standard native format into a semantic data compliant format, then using semantic alignment techniques process the graphs from RDF-based UFC and CityGML are matched using ontology rules, (3) Using semantic graph query filtering based on given criteria for an application-oriented domain, sub-models are extracted then exported, (4) the graph data is loaded into a graph database system (Hor and Sohn, 2021), the consumed into an application through web-based services APIs. The semantic web services, back-end graph data, and the application can all be hosed on a geospatial cloud platform (Hadi et al., 2016; Hor and Sohn, 2021). Furthermore, geospatial cloud platforms can share published and stream services to other systems, including IoT and smart city platforms (Hor and Sohn, 2021).

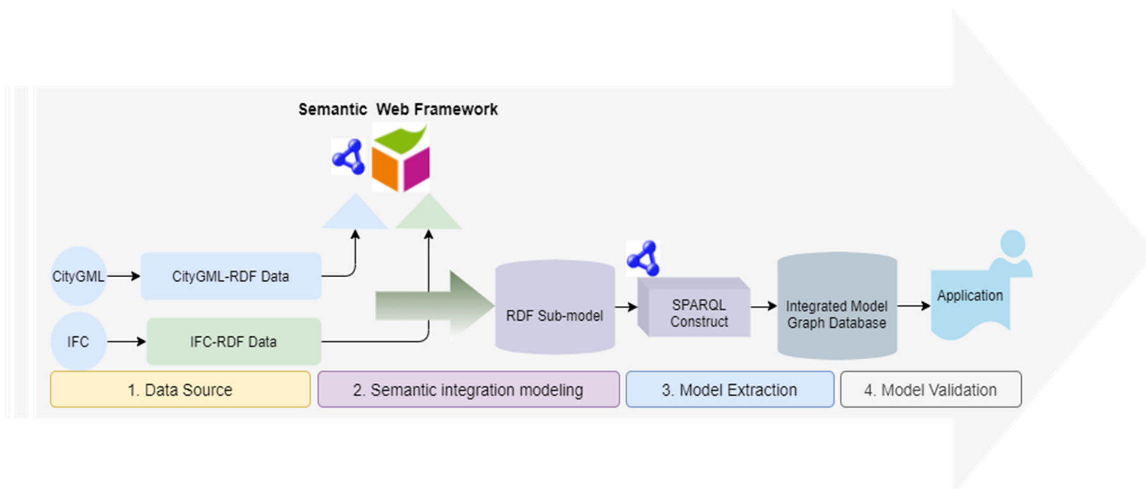


Figure 6.1 Conceptual system

Figure 6.2 represents a detailed implementation architecture diagram of the integrated semantic model with modules, sub-modules, and workflows associated with each phase. It is essential to mention that using Neo4J was determined not solely because it is a perfect triple-store repository but also its graph data analytics capabilities and querying engine, and the BIMserver component (Hor and Sohn, 2021; Hadi et al., 2016). Furthermore, this makes it possible to create an enhanced Scene layer package with embedded web services that can be published and hosted in a cloud-based GIS platform like ArcGIS for Enterprise/ ArcGIS Online used in our implementation.

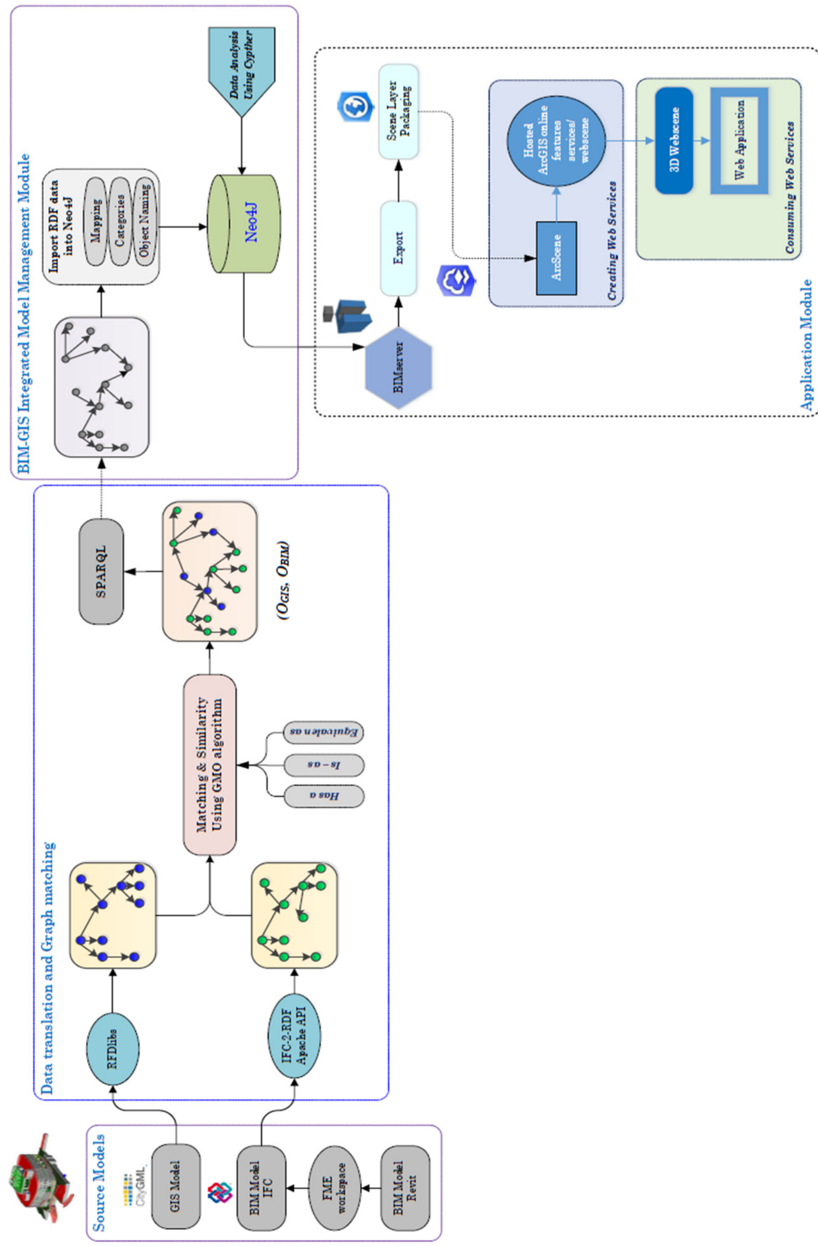


Figure 6.2 BIM-GIS Integrated model using semantic RDF graph database Neo4j

The geospatial data with its geometries and attributes transformation RDF-based model is achieved using direct matching between CityGML elements and designated ontology class. The attributes and values are used to feed the RDF target model. In our methodology, each GIS element corresponds to and matches only one class (Hadi et al., 2016; Hor and Sohn, 2021). However, there are instances in which the fundamental element classes are built from multiple classes (elements) for modelling purposes. In the case of the BUILDING element, the shape (geometry) was extracted and converted to the geometry class based on the SPARQL query to include the corresponding BIM building class (Hor and Sohn, 2021).

Since the IFC class structures are more complicated, the ETL process for the IFC data needed many steps compared to their counterpart elements in the CityGML. Therefore, we developed matching data based on one-to-one source data and the corresponding semantic model (Hor and Sohn, 2021; Hor et al., 2018; Bakis et al., 2007). However, there were some special cases, summarised in table 6.1.

<table>
<tr><th>BIM – IFC Source class</th><th>IGIM Building Ontology class</th></tr>
<tr><td>IfcBuilding</td><td>IGIM_Building</td></tr>
<tr><td>IfcStorey</td><td>IGIM_BuildingStorey</td></tr>
<tr><td>IfcSpace</td><td>IGIM_BuildingSpace</td></tr>
<tr><td>IfcSlab</td><td></td></tr>
<tr><td>(PredefinedType=Floor)</td><td>IGIM_</td></tr>
<tr><td>(PredefinedType=Roof)</td><td>IGIM_Roof</td></tr>
<tr><td>IfcRoof</td><td>IGIM_Roof</td></tr>
<tr><td>IfcWallStandardCase</td><td rowspan="2">IGIM_Wall</td></tr>
<tr><td>IfcWall</td></tr>
<tr><td>IfcCurtainWall</td><td>IGIM_CurtainWall</td></tr>
<tr><td>IfcPlate</td><td></td></tr>
<tr><td>IfcWindow</td><td>IGIM_Window</td></tr>
<tr><td>IfcBuildingElementProxy</td><td>IGIM_BuildingEquipment</td></tr>
<tr><td>IfcMember</td><td rowspan="3">IGIM_Structure</td></tr>
<tr><td>IfcBeam</td></tr>
<tr><td>IfcColumn</td></tr>
<tr><td>IfcCovering</td><td>IGIM_Covering</td></tr>
<tr><td>IfcMaterialLayer</td><td>IGIM_MaterialLayer</td></tr>
<tr><td>(LayerThickness)</td><td>IGIM_Thickness</td></tr>
<tr><td>IfcMaterial</td><td>IGIM_Material</td></tr>
</table>

Table 6.1 IFC classes to ontology classes mapping

Another example is data categorization based on BIM functional model, and the IfcSlab could be a stair, a floor, or a roof in a building model (Hor et al., 2018; Hor and Sohn, 2021). As a result, it was defined on its predefined data type and value in the BIM-GIS model. On the other hand, the information from the IfcCurtainWall class did not have adequate data to serve BIM-GIS integration. Therefore, this class is used as a reference and omitted during the integration workflows.

A- CityGML to RDF Translation Pipeline

The conversion of CityGML 3DGIS data to RDF implies representing GIS thematic properties (Tabular data with an entity-relationship model) and all associated geometry information associated with CityGML elements (Hor et al., 2018). The Geotools (GeoTools The Open Source Java GIS Toolkit — GeoTools) Software Development Kits (SDK) were used to design and develop custom scripts to convert the spatially enabled dataset into RDF triples using powerful features, methods, and java-based libraries. This aimed to manipulate well-known GIS data formats like Google KML, Esri shapefile, OGC GML, and XML, including their geographic projections, datum transformations, and coordinate systems descriptions (Hor et al., 2018; Hor and Sohn, 2021).

The process generates RDF triples with objects, properties, and relationships (Hadi et al., 2016; Hor and Sohn, 2021) for each translated element in the model. Any associated geometry information defining the geographical features of the data can be represented using GeoVocab (GeoVocab.org) libraries, providing vocabularies for geospatial modelling. The workflow pipeline parses objects into features with schema and descriptions and coordinates information from CityGML as defined by the OGC standards (Hor and Sohn, 2021; Hadi et al., 2016).

The translation and parsing process from CityGML elements to RDF triples is illustrated in figure 6.3 below. The pipeline includes parsing and then categorizing data to geometrical, which takes care of extracting and translating the coordinates reference system information. In addition, this also deals with thematical information with all the properties and values of the objects that will be stored in a build-in repository managed by an ontology engine (Stanford protégé in this case) (Hor and Sohn, 2021; Hor et al., 2018).

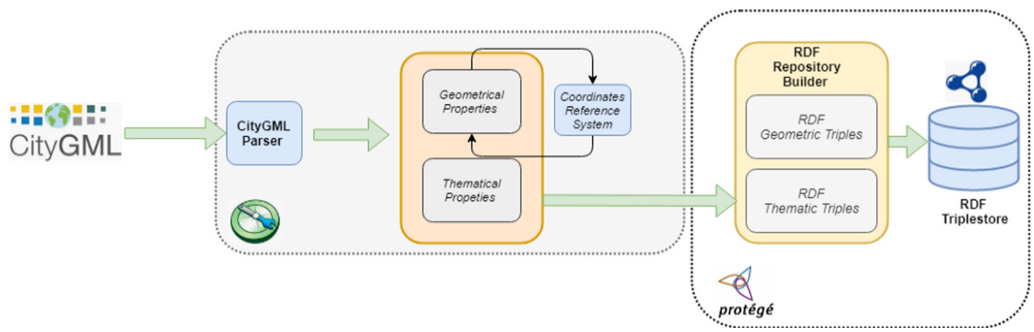


Figure 6.3 CityGML to RDF Translation pipeline

As indicated in figure 6.3, the RDF repository builder module processes geometrical and thematical properties separately to generate a new set of RDF subjects, predicates, and objects (triples) that can be stored in a triple-store (Hor et al., 2018; Hor and Sohn, 2021).

During the process, the predicates are created, take properties names are by default and acquire the dataset vocabulary. The matching operation is taken care of by the RDF platform. This study has selected the Stanford protégé ontology engine for our pipeline workspace https://protege.stanford.edu/.

B- IFC to RDF Translation Pipeline

Figure 6.4 represents IFC to RDF workflow translation pipeline; like CityGML to RDF diagram above, the data is exported in Revit native format to the RDF triple repository (managed into Neo4j database). The process parses and converts building elements into RDF data (Hadi et al., 2016; Hor and Sohn, 2021). A Global Unique Identifier (GUID) links geometrical and thematical properties and keeps the hierarchy between resulting RDF classes and relationships similar to IFC classes from the source model.

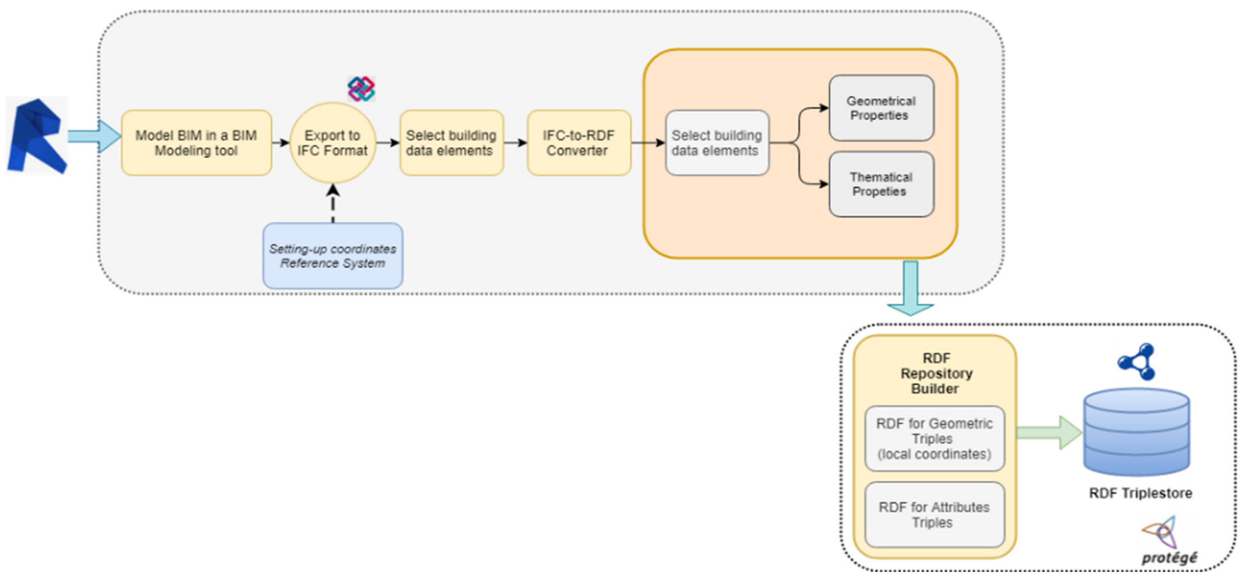


Figure 6.4 CityGML to RDF Translation pipeline

C- Filtering - SPARQL Query

Given the structure of data and geometry information in IFC and CityGML and knowing that BIM and GIS are assimilated as layers with spatial features in an integrated model, the geometry classification from either domain is critical in selecting specific information or extracting information sub-models. Therefore, it affects, on the one hand, the design of SPARQL queries, the resulting graphs, and ultimately the data structure that is used in the domain application or the decision support system. On the other hand, it can be resources demanding and computationally intensive (Hor and Sohn, 2021; Hadi et al., 2016). The semantic query language SPARQL was used to query and filter helpful information from the integrated semantic model. As shown in figure 6.5, we can construct various queries to extract information about specific elements or a set of elements and perform complex analyses on the resulting data (Hor and Sohn, 2021). In the example below, a filter has been established on all windows, including non-spatial attributes like ID, Name, Exposure, and associated material (see appendix A JSON output file) (Hor and Sohn, 2021).

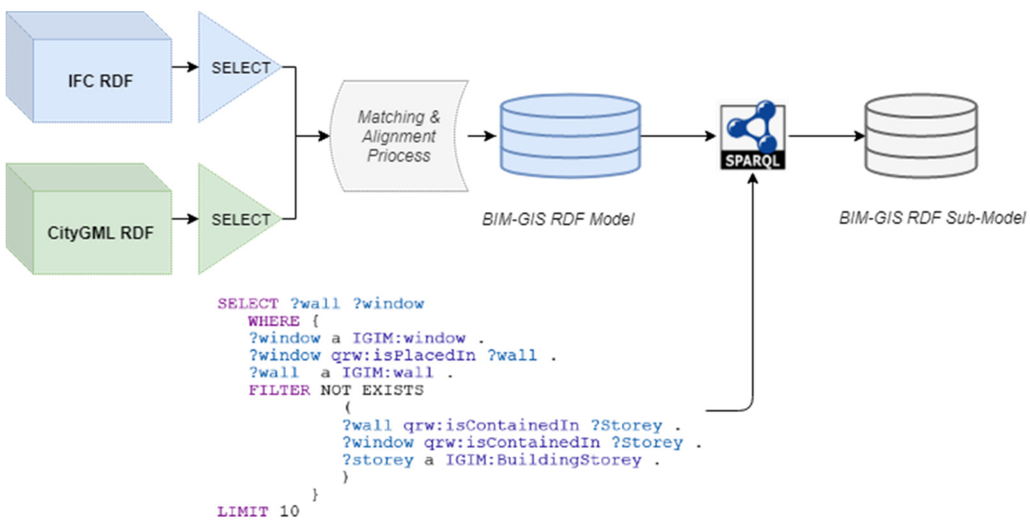


Figure 6.5 Sub-model extraction using SPARQL Filtering

D- Neo4j and Cypher Graph Queries

Once the conversion, the matching process of BIM and GIS data to semantic format, then the export and loading of BIM-GIS RDF data into a graph database system, it is necessary to develop scripts using Cypher to include or exclude specific IFC or CityGML elements from the resulting sub-model to comply with the specifications of data analytics and requirements of the target domain of the application. This data will be imported into Neo4j from a comma-separated value (CSV) file, as shown in figure 6.6.

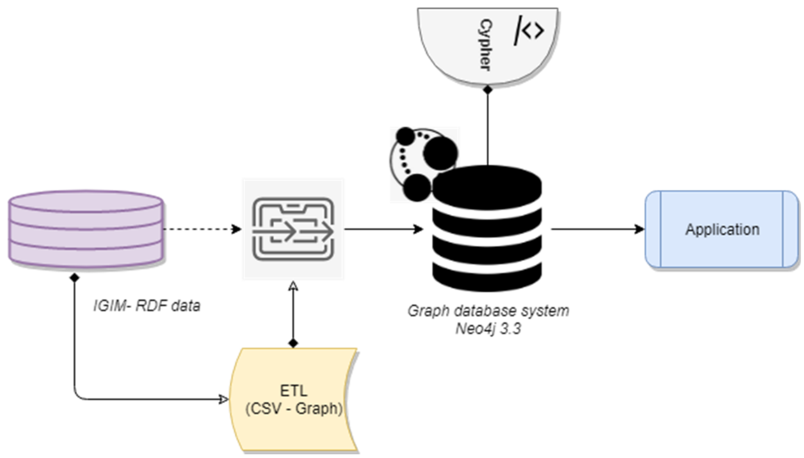


Figure 6.6 ETL process from RDF model to Graph-based database

Figure 6.7 shows a graph sub-model representing spaces associated with level 2 and level 3 from York University (Toronto, Canada) Bergeron building-integrated model (Hor and Sohn, 2021).

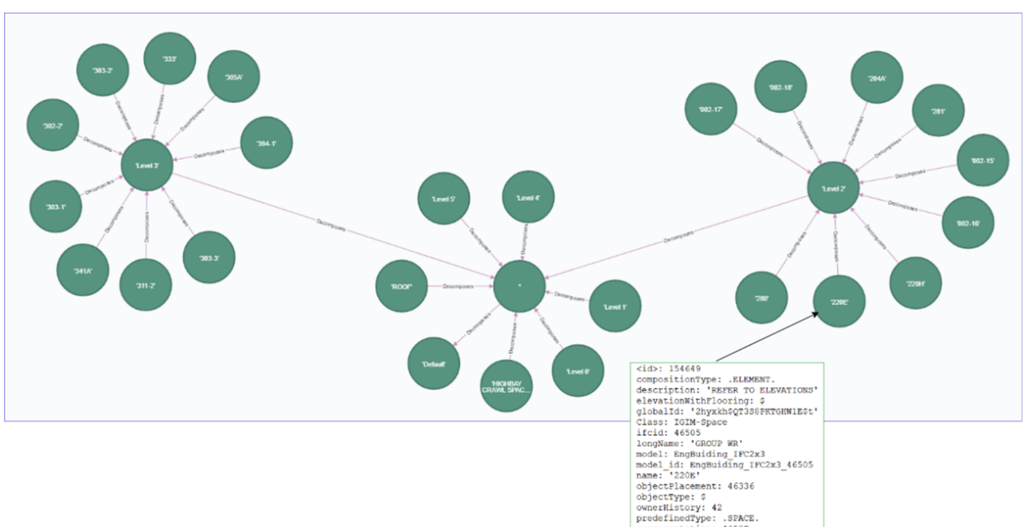


Figure 6.7 Output results on Neo4j Browser

For example, this query retrieves the assigned properties of certain objects through a global ID, and the second output shows the same query in which global ID and building Address are used to extract all associated nodes and their attributes (Hor and Sohn, 2021; Hor et al., 2018).

```
MATCH (wall:IGIM-WallStandardCase{Model:'EngBuilding',GUID:'3wP5TgnCHEPwsgi3SxIcXs'})-[rel]-(property:PropertySet) RETURN DISTINCT wall,rel,property
```

The output of the query result using the Neo4j browser will look like this (fig 6.8), the nodes are objects and relationships linking them based on the selection criteria in the query (Hor and Sohn, 2021):

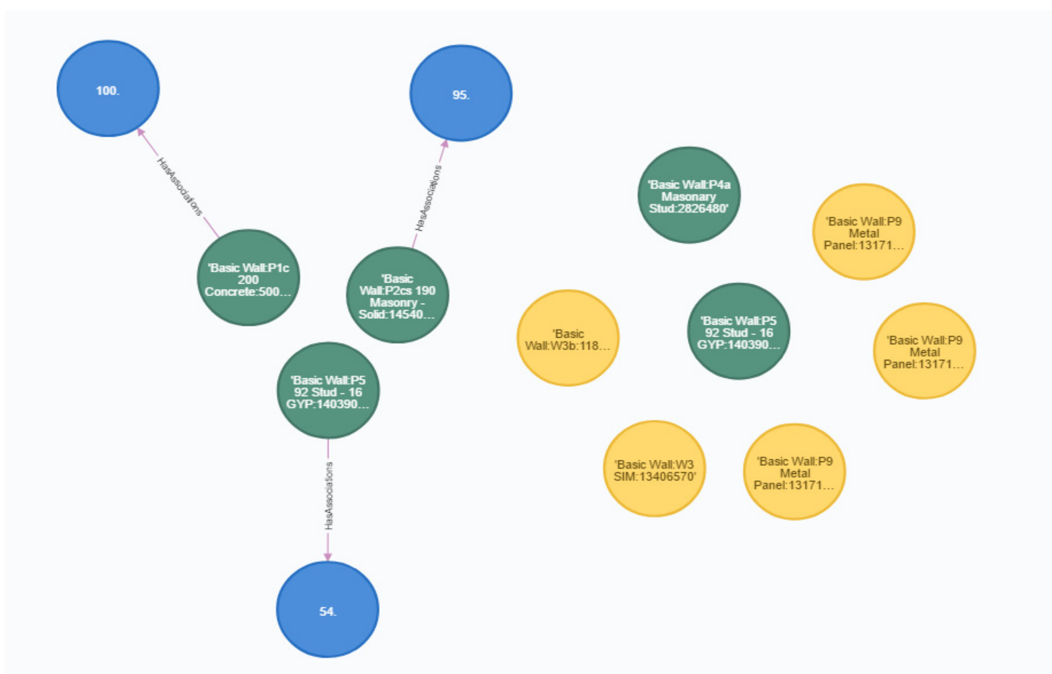


Figure 6.8 Wall properties extracted and associated properties

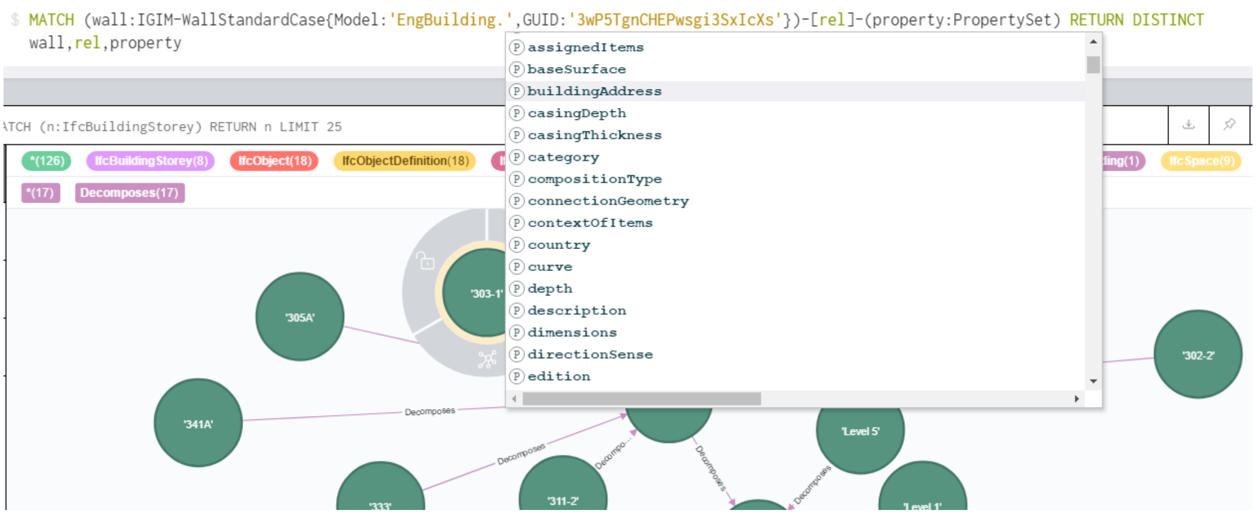


Figure 6.9

6.2.2 Testing Data

To implement our IGIM semantic integrated model using the RDF graph database, we used the following testing datasets (Hor and Sohn, 2021), as indicated in (Table 6.2):

Our testing has been conducted on all the datasets. However, more focus has been given to the real-world models with completed (semi-complete data). These models are :

1- Riverside Building (DC) from Nemetschek Vectorworks Inc., *http://www.vectorworks.net*

2- Arboleda building in Santo Domingo, Dominican Republic.

3- Bergeron Engineering building from York University (Toronto, Canada) *http://ancillary.info.yorku.ca/campus-services/facilities-development/#squelch-taas-accordion-shortcode-content-2*

4- Life Science Building from York University (Toronto, Canada): *http://ancillary.info.yorku.ca/campus-services/facilities-development/#squelch-taas-accordion-shortcode-content-0*

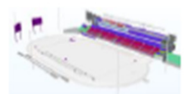
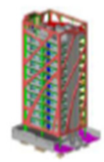
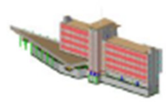
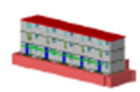
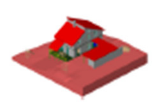
#	Name and description	Model	Size (MB)
1	Lassonde Engineering Building – York University		196
2	Pan-American Stadium – York University –		291
3	Life Science Building – York University -		22.7
4	Arboleda Building		147
5	Building Riverside – Washington DC		274
6	All-Plan Smiley West Building		4.93
7	ALLPLAN Institute Building 2008		8.83
8	Bien-Zenker_Jasmin-Sun		9.12
9	FJK Hauz		10.8
10	FZK Hauz		23.7

Figure 6.10 (a) – Complete testing datasets

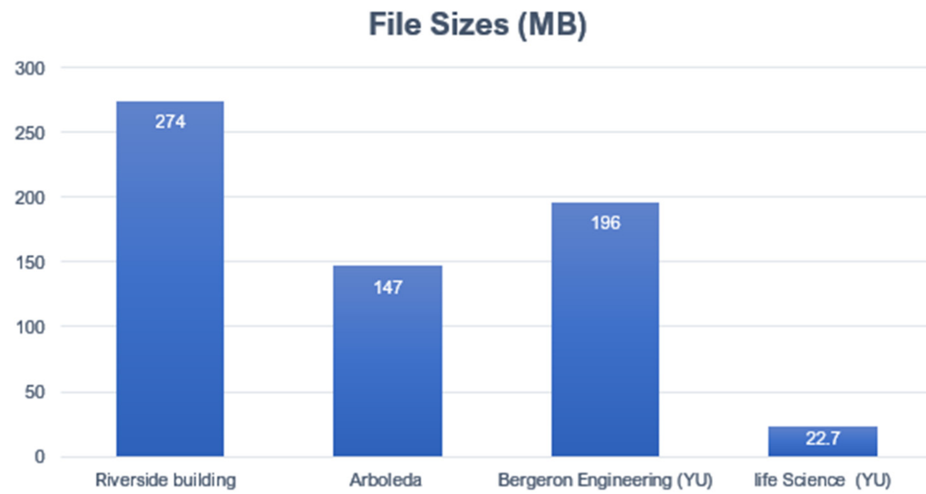


Figure 6.10 (b) Validation testing datasets

We used the first and second models of VectorWorks software to implement and validate the BIM workflow. York University Campus Services provided the other two models & Business Operations (CSBO) to test/demonstrate BIM-GIS semantic integration (Hor and Sohn, 2021; Hadi et al., 2016; Hor et al., 2018).

6.2.3 System Components and Validation

In figure 6.1, we highlighted the complete system architecture. The system includes an extraction module, taking the input source IFC and CityGML models and parsing the data into an RDF graph dataset using Apache Jena (Hor et al., 2018; Hor and Sohn, 2021) (a Java web-based open-source framework). This framework uses a rich set of Application Programming Interfaces (APIs) to construct and build RDF graphs (Hadi et al., 2016; Hor and Sohn, 2021; Bakis et al., 2007), and it includes the following:

1. **RDF API Module**: This Module is used to manipulate RDF data in memory or during execution runtime. This can be in any supported semantic format (RDF/XML, Turtle, N3 triples).

2. **ARQ Engine**: This module supports SPARQL filtering operations.
3. **TDB Engine**: This high-performance store to store, access, and manage commands and scripts using Jena API.
4. **Apache Jena Fuseki**: This integrated server provides a transactional storage layer to the Neo4j database.

Nowadays, graph databases system, including Neo4j, is equipped with advanced data management and analytics tools like advanced analytics and graph data mining algorithms. Graph databases also provide efficient data manipulation and visualization. These tools can be customized based on the domain of the application (Hor et al., 2018; Hor and Sohn, 2021). The management console is one of these tools. It is available for both on-prems or cloud implementations. It can be used as an interface to check data, run queries, and visualize results.

Users and practitioners can develop simulations using graph databases and execute complex queries for many use cases, from emergency evacuation, cost estimation, and shortest path retrieval to extracting and adding models for what-if scenarios (Hor and Sohn, 2021).

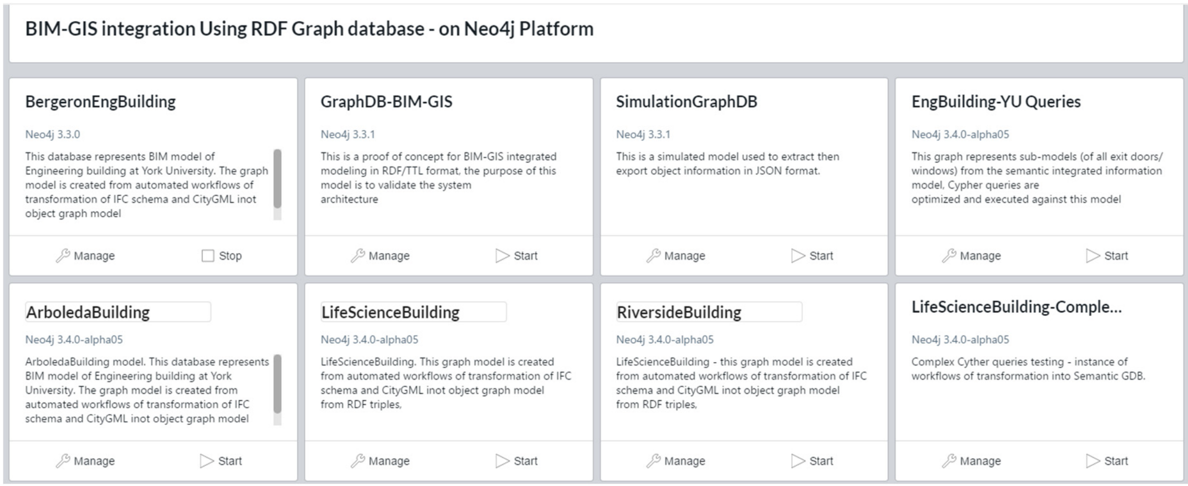


Figure 6.11 Neo4J admin console showing model and sub-models created during the system validation phase

To test and validate our BIM-GIS semantic integration approach, we created four graph database instances running the Neo4J 3.3.0 (Enterprise Server) with browser console 3.2.17. Each of these instances is used to prove and demonstrate some specific areas of the integration methodologies of this research project (Hor and Sohn, 2021; Bakis and Pal, 2009a; Hor et al., 2018).

- ***EngBuildingIFC***: This instance is dedicated to the Bergeron Engineering building and is used to test automated ETL workflows to RDF triples from parsed IFC classes.
- ***GraphBIMGIS***: This instance was developed for data import/export to different RDF data formats. It was also used to validate the design architecture.
- ***SimulationGraphDB*****:** This instance was created to test and validate the BIM-GIS linkage and conjuncture using GUID and JSON data export from BIM-GIS sub-models, as shown in figure 6.10.

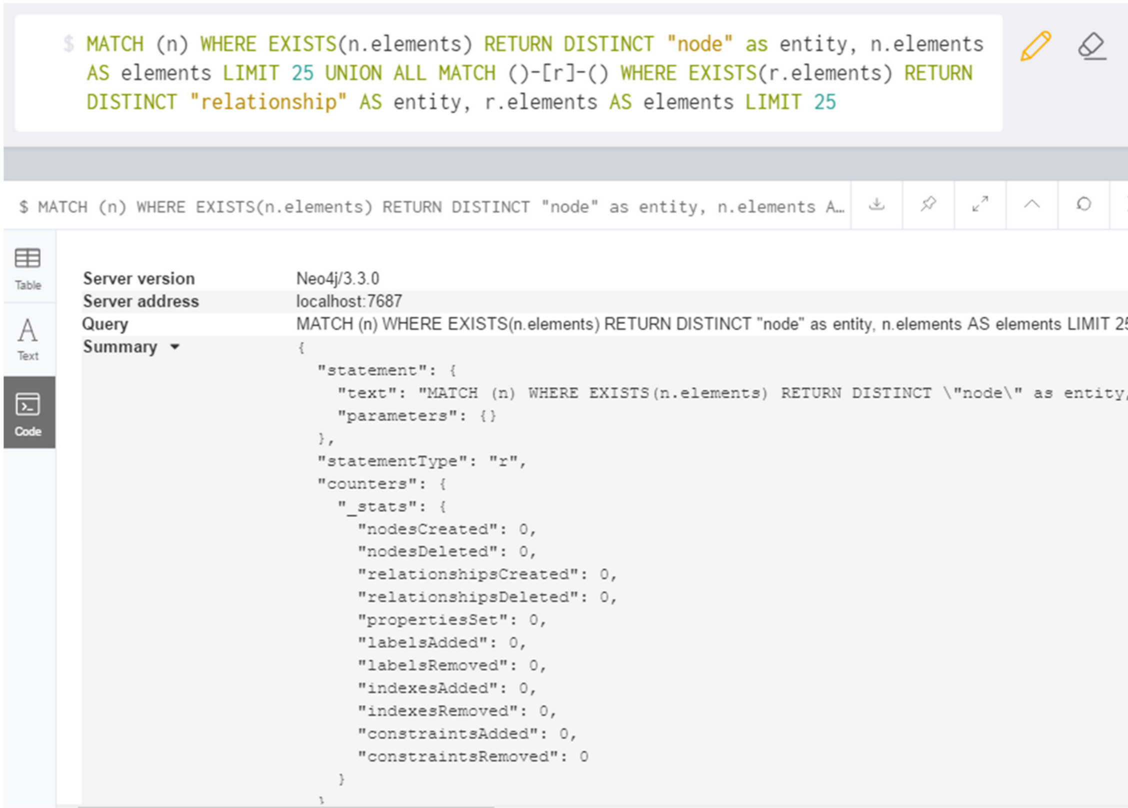


Figure 6.12 Object nodes query result exported into JSON format

- ***EngBuilding-YU-Queries***: This instance aims to validate optimization Cypher queries and their execution plans. The extracted data was in the form of sub-models.
- ***Other Neo4j GDB*** instances were created and dedicated for data validation for most complex queries during runtime. They include sub-graphs of optimization process evaluation and output data comparison, as shown in figure 6.11.

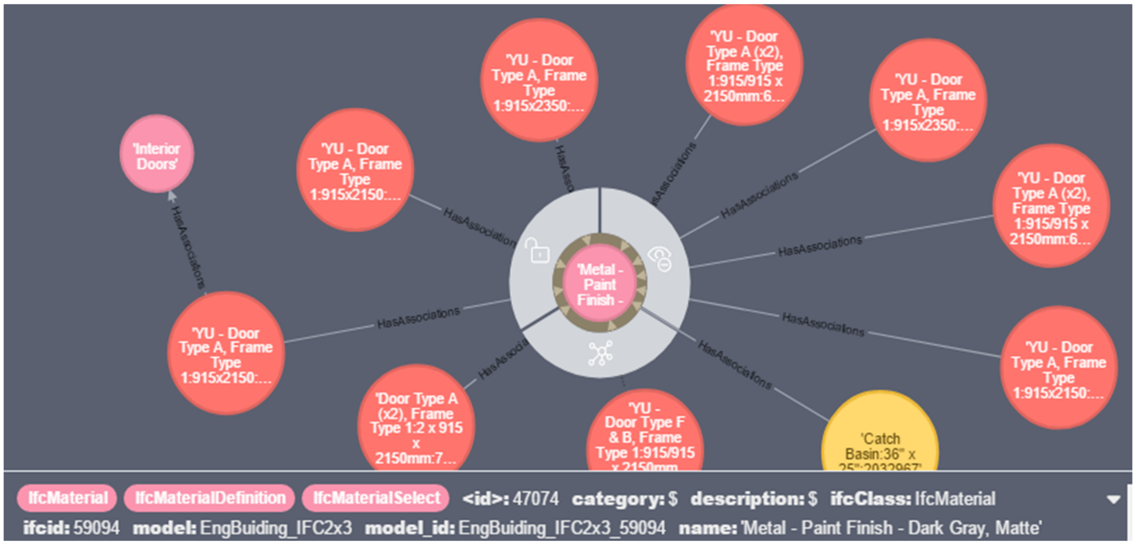


Figure 6.13 Doors elements information from the third floor of Bergeron Engineering Building

The Neo4j graph database import/export utility extracts data in various file formats JavaScript Object Notation (JSON), or Comma-separated values (CSV). With its high data transfer performance that can reach 1 million records/second, it will provide a robust extraction and loading workflow to construct the IGIM semantic integrated model through incremental data layers loading till the entire model is built and all nodes, relationships, and properties (values) as complete defined into the integrated graph model and stored in the graph database system (Hor et al., 2018; Hor and Sohn, 2021). A service connection can be established to the BIMServer (BIMserver.org – open source building information server) to stream graph sub-models using pre-designed Restful (REST) and Simple Object Access Protocol (SAOP) JSON APIs to different data formats (IfcXML, Collada, KML) based on target domain application on the presentation tier and type of system integration within the platform, as shown in figure 6.12 and figure 6.13. Furthermore, additional service enhancements and module custom capabilities will be required to add functionalities to the integrated model (Hor and Sohn, 2021).

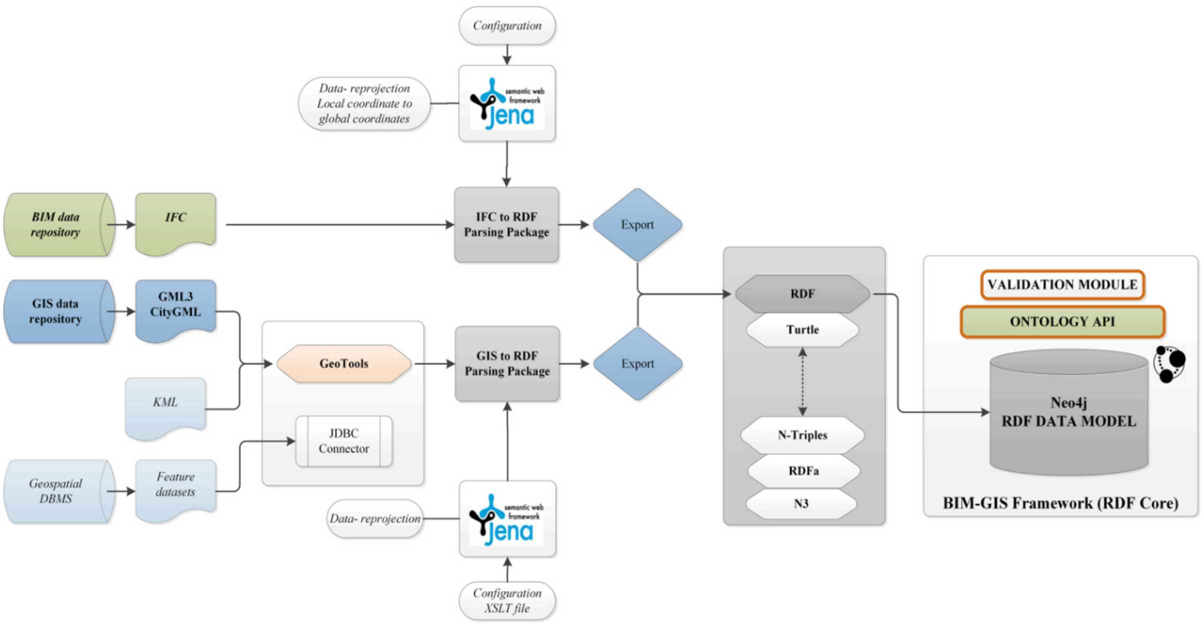


Figure 6.14 High-level system architecture with validation module BIM-GIS RDF core

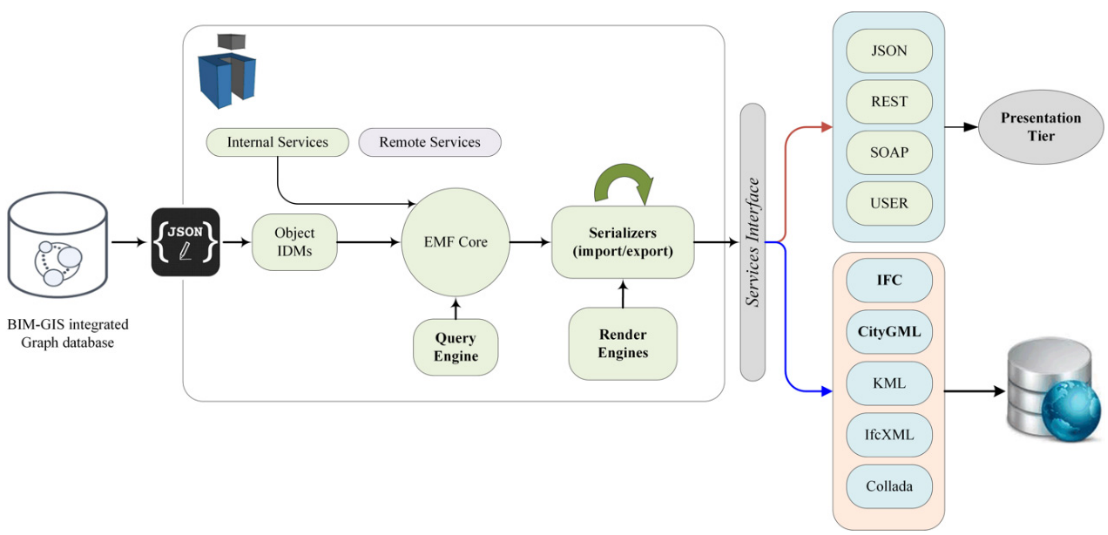


Figure 6.15 High-level BIM Server components

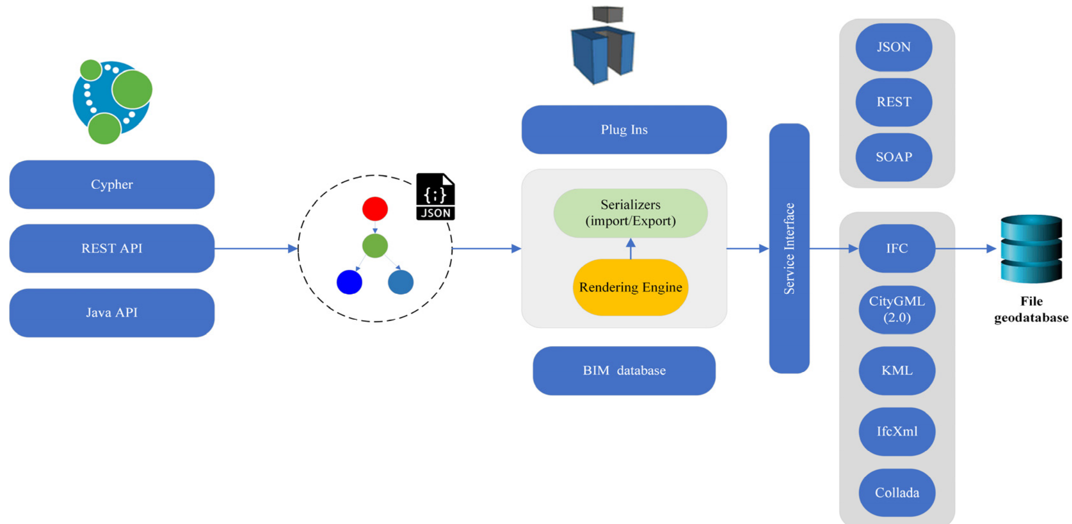


Figure 6.16 BIM Server Serialization plugins BIM Server components

Based on model-driven architecture (Hadi et al., 2016; Hor and Sohn, 2021), the BIMServer stores, manages and presents data as objects with associated attributes and geometry information. It offers advanced functionalities and features to explore Building models. Along the graph database, a BIM-GIS integrated model can take advantage of these two components and provides a backend tier serving out objects in a multi-patch geometries web scene feature layer packages (SLPK) to geospatial platform based on ArcGIS enterprise using ArcGIS Pro georeferencing software development kit (SDK) to generate the WLD3 (3D world file) coordinate transformation file to read and apply proper geographical offsets to the data loaded into the integration platform. The transformation file uses 3D coordinate transformations algorithms (Hor and Sohn, 2021) and a collection of from-to-points and georeferencing tools to define data location. This information will be embedded into a web scene layer package published on Enterprise ArcGIS and ArcGIS Online (cloud GIS) platforms and consumed in a Unity3D game engine to support an intelligent urban mobility application (Hor et al., 2018), as shown in figure 6.15.

Figure 6.17 Web Service and application tier

Figure 6.16 shows selected attributes from BIM-GIS semantic integrated model exported. Originally these attributes are extracted from either IFC classes or/and CityGML elements, respectively representing sources input models (Hor et al., 2018; Hor and Sohn, 2021; Bakis and Pal, 2009a).

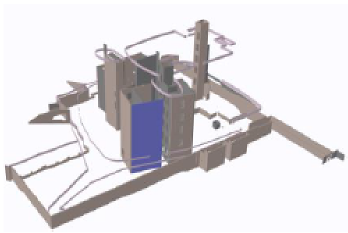
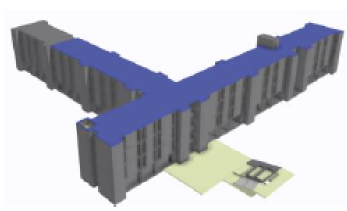
Item	Model	Object	Exported RDF - Attributes
Bergeron Building		Wall	Wall_surface_Gen12Shape7 Attributes element_id 1dhbeG_59C1fQJ9YDNJUPY element_parent_id 1z4sqj8PvCSeCKwX3yvN1d GlobalId 1dhbeG_59C1fQJ9YDNJUPY Name Basic Wall:Foundation - 300mm Concrete:301546 OBJECTID 248 ObjectType Basic Wall:Foundation - 300mm Concrete:1060 Pset_WallCommon_ExtendToStructure 1 Pset_WallCommon_IsExternal 1 Pset_WallCommon_LoadBearing 1 Pset_WallCommon_Reference Foundation - 300mm Concrete Tag 30154
Founders lanes		Slab	Slab_surface_Gen0Shape18 Attributes Analytical_Properties_Absorptance 0.1 Analytical_Properties_Roughness 1 Constraints_Height_Offset_From_Level 0 Constraints_Level Level: Level 5 Constraints_Related_to_Mass 0 Constraints_Room_Bounding 1 Construction_Default_Thickness 1 Construction_Function Interior Dimensions_Area 17889.3772478768 Dimensions_Perimeter 1248.31956606078 Dimensions_Thickness 1 Dimensions_Volume 17889.3772478768 element_id 3scKQjmrvEMv3nKPrMNpfr element_parent_id 1JWOjq27b0OP0TQuDz2FFL GlobalId 3scKQjmrvEMv3nKPrMNpfr Graphics_Coarse_Scale_Fill_Color 0 Identity_Data_Assembly_Code B1010 Identity_Data_Assembly_Description Floor Construction Name Floor:Generic - 12":381032 OBJECTID 19 ObjectType Floor:Generic - 12" Phasing_Phase_Created Existing Pset_SlabCommon_IsExternal 0 Pset_SlabCommon_LoadBearing 1 Pset_SlabCommon_Reference Generic - 12" Tag 381032

Figure 6.18 RDF exported attributes from IGIM

6.2.4 Performance Evaluation

To evaluate the semantic integration methodology and system architecture designed for this research study, we designed and developed many semantic queries and filtering procedures ranging from simple data selection to complex semantic analytics using GeoSPARQL (GeoSPARQL - A Geographic Query Language for RDF Data | OGC) and Neo4j Cypher (Cypher Query Language - Developer Guides (neo4j.com)) query languages. The performance evaluation testing in this section aims to validate the BIM-GIS semantic integration approach and determine the usability and practicality of using ontology modelling and graph database to integrate AEC and Geospatial domains. Several experiments are designed to provide deep insights into the BIM-GIS integration. Besides valuating and measuring the performance metrics of the integration system (Hor et al., 2018; Hor and Sohn, 2021; Bakis et al., 2007), we are mainly looking at these criteria:

- Data richness of the integrated semantic model.
- Computational efficiency and systems resources usage compared to volume and model complexity.
- Data accuracy.

1. **Experiment A: Nodes, relationships, and properties occupancy indexing**

   In this test, we extracted all our testing models' detailed information using SPARQL, Cypher, and Neo4J administration tools, and we have summarised them in Table 6.2.

	Building Information Modeling					Geographic Information System				
	Size (MB)	Entities	Relationships	IFC to RDF (s)	RDF triples	Size (MB)	Entities	Relationships	IFC to RDF (s)	RDF triples
Riverside building	274	1520	2001	1020	2852992	162	623	219	894	1522951
Arboleda	147	856	553	762	3564928	142	289	141	581	16311250
Bergeron Engineering (YU)	196	982	653	1420	4848186	184	301	188	1003	2422160
Life Science (YU)	22.7	182	122	858	1654982	18.5	143	131	607	156231

Table 6.2 Testing models

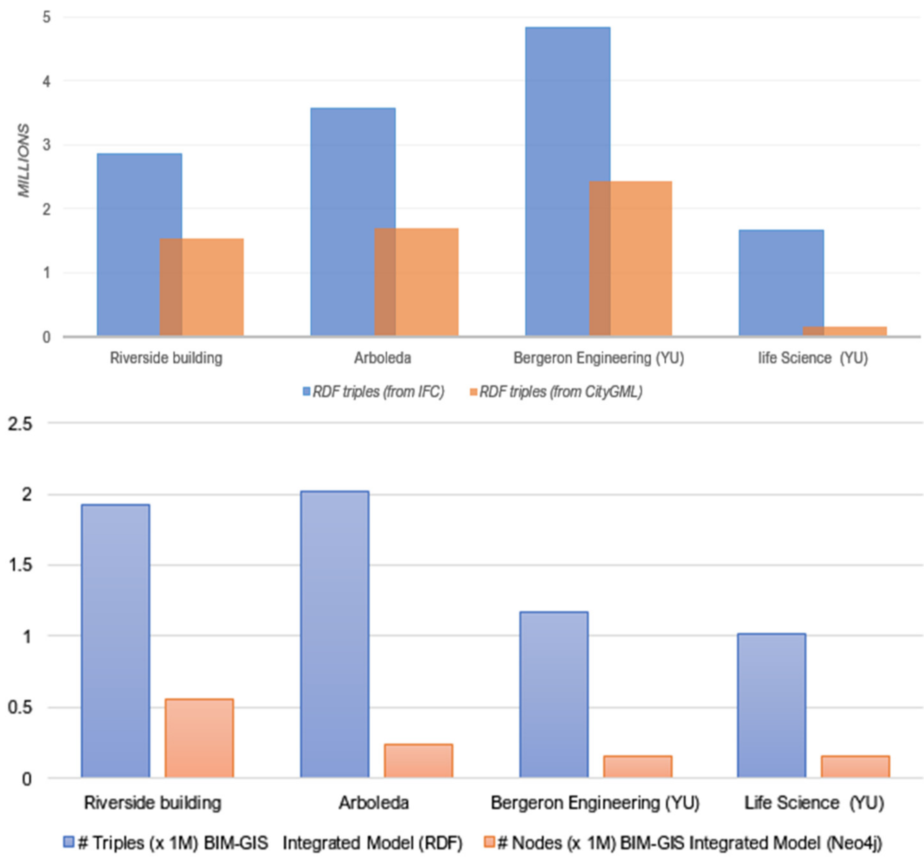


Figure 6.19 RDF attributes exported from BIM-GIS Semantic Model / Test data

Figure 6.17 indicates that the number of BIM/IFC models is much higher than those from GIS/CityGML models before exporting into graph database systems. A similar pattern can be noticed for the numbers of nodes between the two models after an export into the graph database system was executed. This variation is notably decreased when we perform semantic integration between these two domains (fig. 6.17 lower portion) (Hor and Sohn, 2021).

The **SYSINFO** command provides graph database statistics such as object allocation, transactional information, and storage sizes. In figure. 6.18 these details from the

Neo4j admin console return our BIM-GIS integrated graph information, including nodes, relationships, and properties (Hor and Sohn, 2021) (attributes and values).

```
$ :sysinfo
```

Store Sizes	
Count Store	23.09 KiB
Label Store	16.02 KiB
Index Store	408.00 KiB
Schema Store	8.01 KiB
Array Store	8.01 KiB
Logical Log	102.68 MiB
Node Store	3.30 MiB
Property Store	34.38 MiB
Relationship Store	565.91 KiB
String Store	4.05 MiB
Total Store Size	145.43 MiB

ID Allocation	
Node ID	158874
Property ID	878868
Relationship ID	16318
Relationship Type ID	5

Page Cache	
Faults	80
Evictions	0
File Mappings	37
Bytes Read	604497
Flushes	1
Eviction Exceptions	0
File Unmappings	20
Bytes Written	8192

Transactions	
Last Tx Id	454
Current	1
Peak	3
Opened	1515
Committed	724

Figure 6.20 Graph database instance information
(Bergeron Engineering Building, York University)

To evaluate and measure the level of semantic integration between BIM and GIS domains. This research introduced the concepts of semantic occupancy indexes for nodes, relationships (edges), and attributes (values) in the IGIM graph model. Each of these semantic occupancy indexes is calculated from three critical pieces of information:

- Graph allocation for nodes, properties, and relationships in the integrated model.
- Storage allocations for nodes, properties, and relationships in the integrated model.
- Total storage of the integrated model.

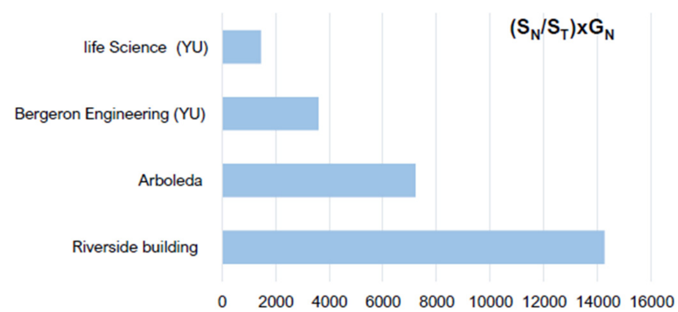
The new semantic occupancy indexes will give practitioners and designers new parameters to look at when integrating BIM and GIS domains. Table 6.3 provides information extracted from the graph database console to define semantic occupancy indexes (Hor and Sohn, 2021; Hor et al., 2018).

	Graph allocations			Store allocations (MiB) – Neo4j 3.3.0 Enterprise					Total Store Size (S_T) (MiB)
	Node (G_N)	Property (G_P)	Relationship (G_R)	Schema	Index	Node (S_N)	Property (S_P)	Relationship (S_R)	
Riveside building	356219	542812	19841	12.26	0.6	6.2	0.82	51.81	154.9
Arboleda	264110	664625	13231	10.14	0.42	4.1	0.58	45.02	150.02
Bergeron Engineering (YU)	158876	878868	16318	8.01	0.39	3.3	0.55	34.38	145.43
Life Science (YU)	102521	401012	10021	6.12	0.22	1.9	0.31	27.05	134.07

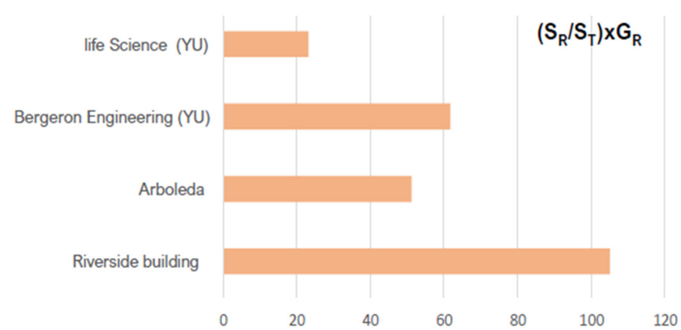
Table 6.3 Testing models

These concepts are the following:

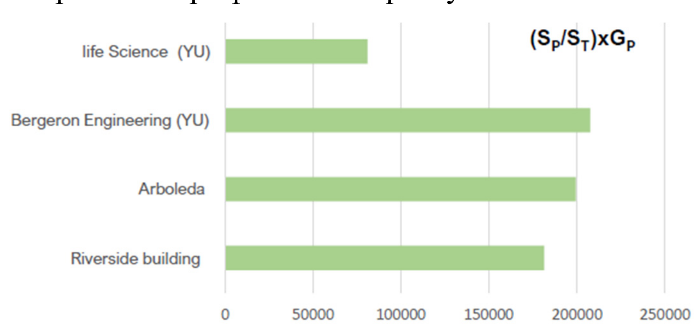
- *$(S_N/S_T).G_N$:* Graph-Model Nodes occupancy:



- *$(S_R/S_T).G_R$:* Graph-Model Relationships occupancy



- *$(S_P/S_T).G_P$;* Graph-Model properties occupancy



By comparing the results of the calculated semantic occupancy indexes for the testing models, we can conclude that even though the Bergeron building model comes second in file size. Moreover, it is ranked first in the properties occupancy index, second in the relationships occupancy index, and third in the nodes occupancy index. This explains that the Bergeron model is the most documented and semantically rich, knowing that this is a relatively new building, where all building elements (slab, door, window, space) data and metadata in the source data are well detailed in the source native format (Autodesk Revit) (Hor et al., 2018; Hor and Sohn, 2021)

2- Experiment B: Query Performance Testing

1- RDF & SPARQL Query Performance Testing

These experiments will perform query performance for data retrieval and information filtering queries from the IGIM RDF graph model. The queries developed in these tests range from simple selects to complex constructs, including nested sub-queries, ordering objects, and grouping properties from the integrated graph model (Hadi et al., 2016; Bansal, 2011). The semantic graph database uses an ARQ query engine supporting the SPARQL RDF Query language (Apache Jena - ARQ - A SPARQL Processor for Jena) (Hor and Sohn, 2021). The following section discusses some samples of the queries used in the testing, with results in different output formats (Hor and Sohn, 2021).

- Query-1: *Retrieving the name of a graph element name identified by GUID.*

```
SELECT ?name
WHERE
{
   ?elem igim:globalId_Root ?guid_box .
   ?guid_box express:hasString "87FaeOo5n2XeWSJjavyqmi" .
   ?elem igim:name_Root ?name_box .
   ?name_box express:hasString ?name .
}
```

- Query-2: *Identifying all walls without openings*.

```
SELECT ?guid
WHERE
{
   ?wall a igim:WallStandardCase .
   ?wall igim:globalId_Root ?guid_box .
   ?guid_box express:hasString ?guid .
   FILTER NOT EXISTS {
      ?rel a igim:RelVoidsElement .
      ?rel igim:RelatingBuildingElement_RelVoidsElement ?wall .
                 }
}
ORDER BY ASC(?guid)
```

- Query-3: *Listing windows with heights information*.

```
SELECT ?window ?height
WHERE
{
   ?window a igim:Window .
   ?window igim:overallHeight_Window ?height_box .
   ?height_box express:hasDouble ?height
}
```

In Table 6.5, all queries and execution times are presented for the four models. It is noticeable that Query-2 has the most extended query execution compared to Query-1 and Query-3 across all the testing models. This can be because this query would need to compute (n_{walls} * $m_{opening}$) times to give results with considerable resources and CPU processing. Query execution plans will be examined for future research, and optimization algorithms will be applied to include additional 3D geometry libraries (Hor and Sohn, 2021).

Item	SPARQL Query	Models – Query time (ms)			
		Bergeron	Riverside	Arboleda	Life Science
1	Query 1	4974	3845	3201	2201
2	Query 2	9866	8511	8018	6428
3	Query 3	8220	7926	7002	4970

Table 6.4 SPARQL query evaluations

2- Cypher Query Performance

The purpose of this test is to evaluate the RDF graph database performance. The environment uses all graph database instances installed and configured on Neo4j 3.4.0-alpha05, shown in figure 6.19 (Hor and Sohn, 2021).

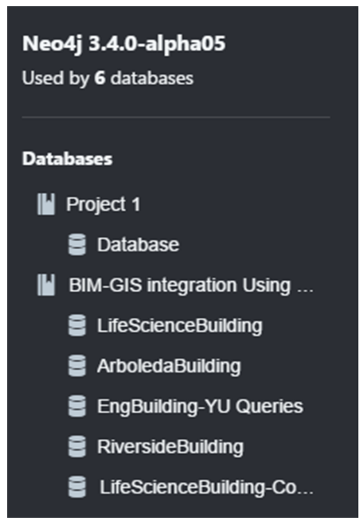


Figure 6.21 Graph database testing environment

Several graph database plugins were enabled to conduct the required tests (Hor and Sohn, 2021):

a. **APOC**: This powerful library includes more than 300 procedures for tasks in many areas ranging from graph and data integration algorithms and data translation.

b. **GraphQL-Endpoint**: This is a Neo4j extension for GraphQL queries and mutations translation into Cypher scripts. It also provides HTTP API and Cypher procedures and offers background processing support using embedded sub-queries.

c. **GRAPH ALGORITHMS**: This Cypher procedures library offers standard parallel graph algorithms processing for the Neo4j graph database.

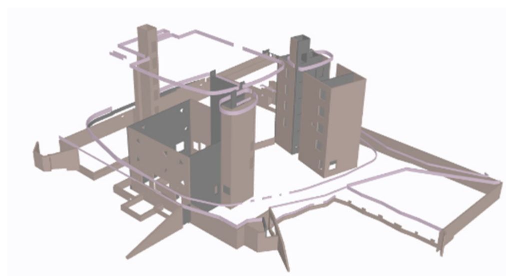
The following Cypher queries are designed to examine various aspects of IGIM model performances. The Queries results are presented in a table and as graph data.

- Query1: *Finding all walls in an integrated graph model.*

```
$ MATCH (space)<-[*1..2]->(wall: WallStandardCase)
  RETURN COLLECT(wall.ID) AS WallStandardCase,
  COUNT(wall) AS Number_of_walls
```

output:

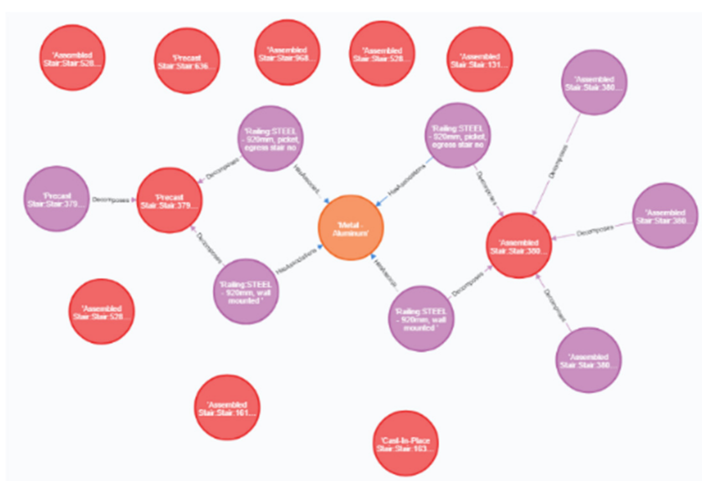
"WallStandardCase"	"Number_of_walls"
[]	5078



- Query2: *Retrieving all the stairs in the integrated graph model*

```
$ MATCH (n:Stair) RETURN DISTINCT (material.ID) AS ID, (material.name)
  AS Material_name
```

Resulting sub-graph:



- Query3: *Extracting all properties information of all elements in the graph model.*

```
$ MATCH (n:PropertySet) RETURN n
```

Output: JSON file


```
{
    "ownerHistory": "42",
    "globalId": "'22Eg2h$f96TuOKK0N7zY9V'",
    "description": "$",
    "model_id": "BergeronBuiding_1885645",
    "representation": "1885641",
    "objectType": "'M_L-Angle:L152X89X7.9:7430817'",
    "objectPlacement": "1885563",
    "predefinedType": ".BEAM.",
    "ifcid": "1885645",
    "name": "'M_L-Angle:L152X89X7.9:7434273'",
    "model": "BergeronBuiding",
    "ifcClass": "IfcBeam",
    "tag": "'7434273'"
}
```


Table 6.5 and figure 6.22 summarise a query execution processing times comparison.

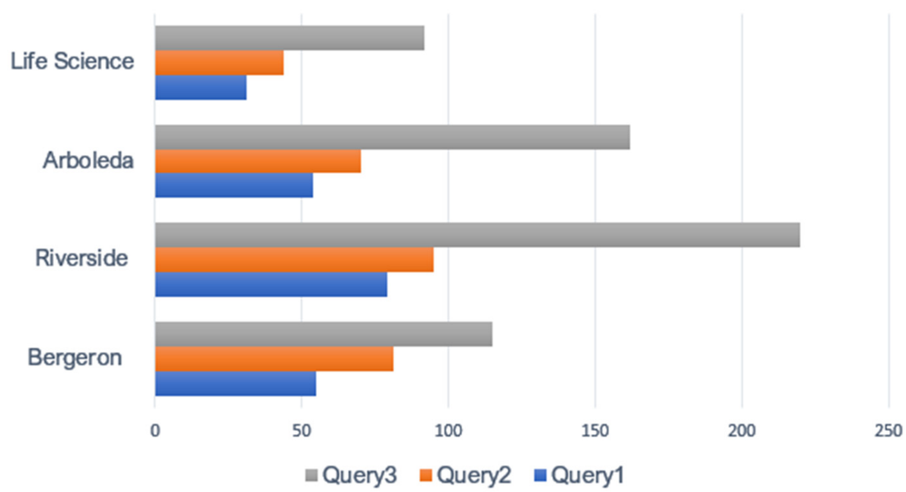


Figure 6.22 Graph database Query testing comparison testing environment

Item	Cypher Query	Models – Query time (ms)			
		Bergeron	Riverside	Arboleda	Life Science
1	Query 1	55	79	54	31
2	Query 2	81	95	70	44
3	Query 3	115	220	162	92

Table 6.5 Cypher Query execution times

A closer look at testing results from SPARQL (RDF query language) and Cypher (graph query language) queries indicates that RDF model results from SPRQL queries were not that good because of RDF file structure and schema (Hor et al., 2018; Hor and Sohn, 2021). On the other hand, Cypher queries were performing better, and their processing execution times were faster, given the volume of graph data to be retrieved from the graph database. It is also important to mention that choosing the configuration of the deployment environment and components (server, database, design queries) is also critical. The semantic BIM-GIS integration can be easily extended by adding more models, such MEP (Mechanical, Electrical and Plumbing) model, and enhanced with analytical workflows in which data retrieval is an important task; therefore, creating an index and performing graph normalization are recommended (Hor and Sohn, 2021; Bakis et al., 2007).

6.3 Summary

In this chapter, we designed, implemented, and evaluated the performance of the IGIM model. The system was developed based on a semantic web technologies framework. It used BIM and GIS from resources description frameworks (RDF) ontological representations from IFC and CityGML domains semantics and graph database principles, respectively.

The System evaluation included creating translation workflow pipelines for BIM/IFC and GIS/CityGML with translation processes and ontologies matching algorithm that looks deeper into integrating the objects based on well-established ontology rules and inputs from semantic dictionaries. An extraction, transformation, and loading of data from RDF into triplestores is developed and enhanced to provide an accurate and rich BIM-GIS integrated model. To achieve such a model, this research has introduced the concepts of semantic occupancy indexes for nodes, relationships, and properties (or values) in the domain of BIM-GIS semantic integration. These occupancy indexes will give a benchmark and measure metrics to evaluate and compare accuracy and richness between different models and integration strategies or methodologies. These new parameters can be utilized as inputs to a decision support system or a master dashboard for various BIM-GIS domain-based applications. By comparing the results using newly introduced occupancy indexes between tested models. We noticed that the Bergeron building model ranked second in file size, comes first with properties occupancy index, second when considering relationships occupancy index, and third in nodes occupancy index. This is because the Bergeron model was the most documented and rich. This building was built to be part of York University and designed so that each element (window, space, door, etc.) has completed and detailed data and metadata from the source model in its native format (Autodesk Revit).

This chapter also designed a performance evaluation and testing using advanced SPARQL and Cypher filters, constructs, and queries. These scripts were developed and compared based on factors like data volume, indexing, and complexity of the source inputs offering insightful information about constructed BIM-GIS Model. We found out that:

- The Semantic ETL process was evaluated based on the number of objects from the sources datasets from both BIM and GIS combined and the final count on these objects in the BIM-GIS Semantic integrated model by using the Occupancy indexes for nodes, relationships and properties, the data loss for the four tests datasets from section 6.2.2 was about between 5.2% and 11.01 % which can be improved by using

AI/ML graph convolutional neural networks to create quality source models, however, this would require training data, and that was out of the scope of this research

- The Cypher queries performed better than RDF SPARQL, considering the data volume processed in the graph database.
- Graph database configuration is essential, including choosing the sizing of the database instance, the volume of data to be stored, and so on.
- Given that the BIM-GIS graph model will get only bigger over time by adding sub-model like MEP (Mechanical, Electrical, or plumbing). We concluded that the Neo4j graph database provides the best data retrieval performance even without indexing– tests have shown that using single property–indexes for nodes and single property index on relationships return better results and improve the query execution times to up 15%.

Chapter 7

Conclusions and Future Directions

7.1 Conclusions

This research project introduced and presented a complete study of a novel semantic integration methodology between Building information modelling and Geographic Information system using the RDF graph database model. The approach combined RDF graph models from BIM and GIS domains and provided an inclusive graph data model combining information from BIM and GIS fields with complete graph analysis and data mining capabilities. The semantic concepts were drawn by representing the building models using both IFC and the CityGML formats. Subsequently, they were translated into a unique, integrated semantic model using semantic web technologies framework and ontology engineering methodologies using translation and import/export workflows into a graph database system using SPARQL and Cypher scripting. The new Semantic Extract-Transform-load (SETL) methodology has provided an approach for data and applications integration methods allowing adding multiple data from different sources with efficient handling of their properties, attributes, and relationships. This model is now named the Integrated Geospatial Information Model (IGIM).

This study introduced a new IGIM model designed and developed using robust semantic matching and alignment technique processes with data extraction, transformation, and loading pipelines into an ontological framework from native IFC and CityGML. Also, this research has :

- Provided a complete BIM-GIS semantic integration pipeline combining BIM (IFC) and GIS (CityGML) Models based on semantic web services and ontology methodologies.
- Determined a new BIM-GIS Data Ontological Oriented Model (DOOM) with more robust data analytics and mining capabilities using graph databases for BIM and GIS models.
- Evaluated multiple semantic matching techniques for BIM-GIS integration between IFC classes and CityGML elements to achieve the best Alignment, Merging and ontological integration (AMI). This research has looked at AMI and Graph Ontology Mapping, which were based on a critical review of many works from various fields and domains of knowledge. This aimed to develop a new version of the mapping algorithm, set thresholds, and specify all criteria of accuracy and richness of the BIM-GIS semantic integrated model.
- Introduced new concepts of nodes, relationships, and properties semantic occupancy indexes to measure compatibility, useability, richness, and accuracy of the BIM-GIS integrated model. The occupancy indexes will new input parameters into a decision support system that would provide an accurate comparison between BIM-GIS semantically integrated models and help in the design architecture and development

of target domain-based applications. There will be a focus on designing BIM-GIS semantic integration dashboards using occupancy factors in the future.

- Developed a flexible service-oriented architecture-based system, providing opportunities to add more online services integration and making it possible to create and consume Web Service Modelling Ontology (WSMO) in an Ontology-Based Application.

7.2 Challenges and Limitations

This research has presented a complete BIM-GIS integration modelling workflow using IFC, CityGML, and RDF graph databases ontology engineering methodologies and frameworks. This project has proven that multiple platforms can be put together. The implementation has demonstrated the usefulness of the proof-of-concept system in information retrieval and pattern discovery management settings. However, there are some limitations and challenges that we have encountered during the conceptualization, architecting, and development of the integration models:

- RDF graphs are enormous. Running SPARQL queries against these files could be slow. To overcome this issue, we created sub-models to work on instead.
- Mapping IFC and CityGML schemas were executed using XSLT transformation, a lightweight conversion that can only be executed at the instance level. It is incredibly time-consuming and prone to errors. Building the XSLT file takes much time (and focus), and the method has limited expressiveness in RDF format. This would affect the export to graph database objects stored on Neo4j as well as the application.

- Query optimization on BIM-GIS integrated RDF graph-based data is a challenging task given the enormous size of data involved in SPARQL and Cypher queries due to the verbosity of the data format. With a lack of schema that challenges the cardinality estimation process during this research work, it was easy to construct a query with less than 20 triple patterns.
- Processing RDF data in BIM or GIS models are computationally intensive (primarily through a web portal). Therefore, a capacity planning system architecture exercise that considers scalability and robustness will be required before considering building the system.
- Some limitations and a few enhancements have been identified during this study, which could affect the reliability of the graph database, such as:
    a) Graph representation of the entire BIM-GIS semantic model is ineffective in very rich datasets since hundreds of nodes and relationships should be presented within the relatively small stream.
    b) Even though this study has provided a methodology to examine the correctness of the BIM-GIS integrated model graph database, it cannot provide a technique for data loss prevention. Therefore, a new technique that can enhance Neo4j with new functions and procedures could be developed to deal with this issue.
- The process is not fully automated, and extending the present research to develop a completely automatic converter should be next in the near future.

- The version control system for the BIM-GIS integrated model to track changes and development history is still unavailable.

7.3 Directions for Future Research

In the near future, following this research work, we will focus on studying cross-platform systems and domain-oriented applications using BIM-GIS semantic information models. A focus will be on designing enhanced matching algorithms using machine and deep learning principles. We will develop new procedures and processes for graph data extraction and sub-model export. We will also merge new models like weather models, the internet of things (IoT), and virtual urban digital twin. Moreover, we like to explore cloud computing platforms and serverless and microservices capabilities and their web services integration-based architecture to provide collaboration, scalability, and automatic, instant updates to models. We believe that what we have done with performance evaluation in this research is the beginning of a long journey of investigating semantic ontologies for BIM-GIS integration. The plan is to study semantic query design and optimization and develop a tuned graph database with well-defined threshold and benchmark metrics for queries and analytical reports execution. We will work to develop new sets of automatic graph tools to help validate geometries from source RDF graphs before integration to ensure valid and rich integrated models.

Our semantic BIM-GIS integration strategy using graph data modelling presented in this work has been proven to be very effective in creating semantic workflows involving BIM and GIS to construct a solid new integration between these two domains. It would open the doors wide for many research projects in the future. However, much effort will be needed to understand these domains of knowledge ontological models. We will focus on their information discovery, data structure, and schema characteristics to make integration flexible and adaptable to the new era. Furthermore, that will begin by creating new research disciplines from industry experts, practitioners, and researchers from BIM and GIS. For our foreseeable future, work will include:

- Investigating the BIM-GIS integrated graph database model KPI metrics for load/import/export and querying operations.
- Developing novel techniques for query processing and optimization in BIM-GIS integrated RDF-based graph databases, looking into shortest path queries, complex graph patterns matching queries, and new indexes (on nodes) for the approximate shortest path between objects in the model.
- Developing new workflows to translate the BIM IFC schema from EXPRESS to RDF/OWL and using ontology taxonomies to build RDF instance graphs. We believe from the testing conducted in this research that it is less prone to errors, i.e., once the mapping between EXPRESS elements and RDF/OWL elements is made. Finally, all tasks will go smoothly and correctly. Also, it has high expressiveness, especially when using N3 notation syntax.
- Developing baseline benchmarks for BIM-GIS integration using graph modelling and comparing integration models using other graph database management systems. Further efforts will focus on implementing these benchmark tasks for quantitative

evaluation of the presented system and standardized query workloads or network/resources processing.

- Developing tools to validate geometries from different data sources.
- Developing a cross-platform application allowing model editing using mobile web services.
- Creating an automated procedure to extract graph subsets and merge sub-models for analysis purposes and what-if scenarios.
- Exploring cloud-based services integration (AWS, Azure), internet of things (IoT), and big data technologies. Given the immense interest in leveraging such technologies to improve the efficiency of geo-design and construction processes, doors will open to explore and deploy IoT. This will also open the doors for using big data analytics engineering techniques in various BIM and GIS domain-specific applications.
- Exploring and developing application extension modules to open-source software tools of platforms such as GeoServer to handle IFC and 3D spatial data to support 3D spatial operations. This could be completed by enhancing CGAL libraries and implementing new internal data structures that represent spatial information from the feature interface in GeoServer. Designing a datastore module to store and handle 3D spatial information in JSON, GeoJSON, and GML could open a new perspective in introducing RDF into FOSS platforms and make it possible to share 3D spatial data via GeoServer.

Finally, it is essential to say that the theories, methodologies, algorithms, and technologies associated with semantic web data modelling with graph databases in the field of architecture, engineering, and construction (AEC) and geographic information systems make it a strong candidate to solve many problems that current integration systems are facing. Currently, web services, geoinformation, BIM modelling life cycles, and embedding new data sources are some of the leading areas where semantics and ontology engineering can be brought to solve the integration and interoperability problems. Furthermore, using web services representing different resources to map concepts in a web service description to ontological concepts would allow users to define semantics for any given domain. Thus, this could enable the discovery of new services and provide more significant benefits when developing mapping for exchanging messages between services that participate in a process such as facility management using a semantic BIM-GIS model.

Bibliography

Hor, A. H., & Sohn, G. 2021. Design and evaluation of a Bim-Gis integrated information model using Rdf graph database. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *8*, pp. 175-182.

Zhu, J.; Wu, P. 2021. Towards Effective BIM/GIS Data Integration for Smart City by Integrating Computer Graphics Technique. Remote Sens. 13, 1889

ESRI. 27 January 2021. Fundamentals of Georeferencing a Raster Dataset. Available online: https://desktop.arcgis.com/en/arcmap/latest/ manage-data/raster-and-images/fundamentals-for-georeferencing-a-raster-dataset.htm.

Mattana, L.; Souza, J.C.; De Faria, M.L.T. 10 February 2021. The Study of Innovation Process in Civil Construction Through BIM Technology. In Proceedings of the International Conference on Production and Operations Management Society, 15 October 2019; Springer: Berlin/Heidelberg, Germany; pp. 771–778. Available online: https://link.springer.com/chapter/10.1007/978-3-030-23816-2_76.

Clemen, C.; Hendrik, G. 3 March 2021. Level of Georeferencing (LoGeoRef) using IFC for BIM. J. Geod. 2019, 10, 15–20. Available online: https://jgcc.geoprevi.ro/docs/2019/10/jgcc_2019_no10_3.pdf.

Andrews, C. 2020. Common patterns for BIM and GIS integration. https://www.esri.com/arcgis-blog/products/arcgis-pro/transportation/common-patterns-for-bim-and-gis-integration/

Hbeich, E., Roxin, A., & Bus, N. 2020. Previous BIM-GIS integration approaches: Analytic review and discussion. p. 47.

Kutzner, T., Chaturvedi, K., & Kolbe, T. H. 2020. CityGML 3.0: New functions open up new applications. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, *88*(1), pp.43-61.

Zhu, J.; Wu, P.; Chen, M.; Kim, M.J.; Wang, X.; Fang, T. 2020. Automatically processing IFC clipping representation for BIM and GIS integration at the process level. Appl. Sci..

Borrmann, A.; Amann, J.; Chipman, T.; Hyvärinen, J.; Liebich, T.; Muhiˇc, S.; Mol, L.; Plume, J.; Scarponcini, P. 19 October 2020. IFC Infra Overall Architecture Project: Documentation and Guidelines. Available online: https://www.buildingsmart.org/wp-content/uploads/2017/07/08_bSI_OverallArchitecure_Guidelines_final.pdf.

Zhu, J.; Tan, Y.; Wang, X.; Wu, P. 2020. BIM/GIS integration for web GIS-based bridge management. Ann. GIS 27, 99–109.

Hor, A. E., Gunho, S., Claudio, P., Jadidi, M., & Afnan, A. 2018. A semantic graph database for BIM-GIS integrated information model for an intelligent urban mobility web application. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, *4*(4).

Uggla, G.; Horemuz, M. 2018. Geographic capabilities and limitations of Industry Foundation Classes. Autom. Constr. 96, 554–566.

Arroyo Ohori, K.; Diakité, A.; Krijnen, T.; Ledoux, H.; Stoter, J. 2018. Processing BIM and GIS models in practice: Experiences and recommendations from a GeoBIM project in the Netherlands. ISPRS Int. J. Geo-Inf. 7, 311.

Liu, X., Wang, X., Wright, G., Cheng, J. C., Li, X., & Liu, R. 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*(2), p. 53.

Ma, Z., & Ren, Y. 2017. Integrated application of BIM and GIS: an overview. *Procedia Engineering*, *196*, pp. 1072-1079.

Pauwels, P., Zhang, S., & Lee, Y. C. 2017. Semantic web technologies in AEC industry: A literature overview. *Automation in Construction*, *73*, pp. 145-165.

Ruy, F. B., Guizzardi, G., Falbo, R. A., Reginato, C. C., & Santos, V. A. 2017. From reference ontologies to ontology patterns and back. *Data & Knowledge Engineering*, *109*, pp. 41-69.

Hor, A. H., Jadidi, A., & Sohn, G. 2016. BIM-GIS integrated geospatial information model using semantic web and RDF graphs. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, *3*(4), pp. 73-79.

Karan, E., Sivakumar, R., Irizarry, J., and Guhathakurta, S. 2014. Digital modelling of construction site terrain using remotely sensed data and geographic information systems analyses." *Journal of Construction Engineering and Management*, 140(3), p. 04013067.

Karan, E. P., and Irizarry, J. 2014. Developing a spatial data framework for facility management supply chains. *Construction Research Congress 2014*, D. Castro-Lacouture, J. Irizarry, and B. Ashuri, eds., American Society of Civil Engineers, Atlanta, GA, pp. 2355-2364.

Balachandar, K., Thirumagal, E., Aishwarya, D., & Rajkumar, R. 2013. Ontology mapping techniques and approaches. *International Journal of Computer Applications*, *65*(24).

Jena, A. 2011. ARQ-A SPARQL Processor for Jena. *Acedido em*, *9*. http://jena.apache.org/documentation/query/.

Curry, E., O'Donnell, J., Corry, E., Hasan, S., Keane, M., and O'Riain, S. 2013. Linking building data in the cloud: Integrating cross-domain building data using linked data. *Advanced Engineering Informatics*, 27(2), pp. 206-219.

GeoTools. 2013. "GeoTools." http://www.geotools.org /.

GIS, Q. 2013. Quantum GIS. http://www.qgis.org/

Irizarry, J., Karan, E. P., and Jalaei, F. 2013. Integrating BIM and GIS to improve the visual monitoring of construction supply chain management. *Automation in Construction*, 31, pp. 241-254.

Jiang, S., Zhang, J., and Zhang, H. 2013. Ontology-based Semantic Retrieval for Risk Management of Construction Project. *Journal of Networks*, 8(5).

Mazairac, W., and Beetz, J. 2013. BIMQL–An open query language for building information models. *Advanced Engineering Informatics*, 27(4), pp. 444-456.

Park, C.-S., Lee, D.-Y., Kwon, O.-S., and Wang, X. 2013. A framework for proactive construction defect management using BIM, augmented reality and ontology-based data collection template. *Automation in Construction*, *33*, pp. 61-71.

Niknam, M., and Karshenas, S. 2013. A Semantic Web Service Approach to Construction Cost Estimating. *ASCE International Workshop on Computing in Civil Engineering* Los Angeles, CA.

Lee, S.-K., Kim, K.-R., and Yu, J.-H. 2013. BIM and ontology-based approach for building cost estimation. *Automation in Construction*, 41, pp. 96-105.

Teicholz, P. 2013. *BIM for Facility Managers*, John Wiley & Sons, New York, N.Y.

Cheung, F. K., Rihan, J., Tah, J., Duce, D., and Kurul, E. 2012. Early stage multi-level cost estimation for schematic BIM models. *Automation in Construction*, 27, pp. 67-77.

USGS. 2013. United States geological survey (USGS): Geospatial semantics and ontology. http://cegis.usgs.gov/ontology.html.

Cromley, E. K., and McLafferty, S. 2012. *GIS and public health*, Guilford Press.

Irizarry, J., and Karan, E. P. 2012. Optimizing location of tower cranes on construction sites through GIS and BIM integration. *Journal of Information Technology in Construction (itcon)*, 17, pp. 351-366.

Liu, R., and Issa, R. 2012. 3D visualisation of sub-surface pipelines in connection with the building utilities: integrating GIS and BIM for facility management. *Proceedings of Computing in Civil Engineering (2012)*, ASCE, pp. 341-348.

Nepal, M. P., Staub-French, S., Pottinger, R., and Webster, A. 2012. Querying a building information model for construction-specific spatial information. *Advanced Engineering Informatics*.

Törmä, S., Oraskari, J., and Hoang, N. V. 2012. Distributed transactional building information management. *First International Workshop on Linked Data in Architecture and Construction*, Ghent University, pp. 9-12.

Azhar, S., Carlton, W. A., Olsen, D., & Ahmad, I. 2011. Building information modeling for sustainable design and LEED® rating analysis. *Automation in construction*, *20*(2), pp 217-224.

Bansal, V. K. 2011. Use of GIS and topology in the identification and resolution of space conflicts. *Journal of Computing in Civil Engineering*, *25*(2), pp. 159-171.

Deutsch, R. 2011. *BIM and Integrated Design: Strategies for Architectural Practice*, John Wiley & sons.

Eastman, C., Teicholz, P., Sacks, R., and Liston, K. 2011. *BIM Handbook: A Guide to Building Information Modelling for Owners, Managers, Designers, Engineers and Contractors*, John Wiley & Sons.

El-Diraby, T., and Osman, H. 2011. A domain ontology for construction concepts in urban infrastructure products." *Automation in Construction*, 20(8), pp. 1120-1132.

Elbeltagi, E., and Dawood, M. 2011. Integrated visualised time control system for repetitive construction projects. *Automation in Construction*, 20(7), pp. 940-953.

Fidan, G., Dikmen, I., Tanyer, A. M., and Birgonul, M. T. 2011. Ontology for relating risk and vulnerability to cost overrun in international projects." *Journal of Computing in Civil Engineering*, 25(4), pp. 302-315.

Hijazi, I., Ehlers, M., Zlatanova, S., Becker, T., and van Berlo, L. 2011. Initial investigations for modelling interior utilities within 3D Geo Context: Transforming IFC-interior utility to CityGML/UtilityNetworkADE. *Advances in 3D Geo-information sciences*, Springer, pp.95-113.

Laat, R., and Berlo, L. 2011. Integration of BIM and GIS: The development of the CityGML GeoBIM Extension. *Advances in 3D Geo-Information Sciences*, T. H. K. Kolbe, G.; Nagel, C., ed. Berlin, Germany, pp.211-225.

Leite, F., Akcamete, A., Akinci, B., Atasoy, G., and Kiziltas, S. 2011. Analysis of modelling effort and impact of different levels of detail in building information models. *Automation in Construction*, 20(5), pp. 601-609.

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van de Walle, R., and Van Campenhout, J. 2011. A semantic rule checking environment for building performance checking." *Automation in Construction*, *20*(5), pp. 506-518.

Singh, V., Gu, N., and Wang, X. 2011. A theoretical framework of a BIM-based multidisciplinary collaboration platform. *Automation in construction*, 20(2), pp. 134-144.

Underwood, J., and Isikdag, U. 2011. Emerging technologies for BIM 2.0." *Construction Innovation: Information, Process, Management*, *11*(3), pp. 252-258.

Venugopal, M. 2011. *Formal specification of industry foundation class concepts using engineering ontologies*. PhD thesis, Georgia Institute of Technology, Atlanta, GA.

Vos, N., van der Meijden, H., and Denessen, E. 2011. Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & Education*, *56*(1), pp. 127-137.

Agdas, D., & Ellis, R. D. 2010. The potential of XML technology as an answer to the data interchange problems of the construction industry. *Construction Management and Economics*, *28*(7), pp. 737-746.

El-Gohary, N. M., and El-Diraby, T. E. 2010. Domain ontology for processes in infrastructure and construction." *Journal of Construction Engineering and Management*, 136(7), pp. 730-744.

Peña-Mora, F., Chen, A. Y., Aziz, Z., Soibelman, L., Liu, L. Y., El-Rayes, K., Arboleda, C. A., Lantz Jr, T. S., Plans, A. P., and Lakhera, S. 2010. Mobile Ad Hoc Network-Enabled Collaboration Framework Supporting Civil Engineering Emergency Response Operations. *Journal of Computing in Civil Engineering*,*24*(3), pp. 302-312.

Sørensen, K. B., Christiansson, P., and Svidt, K. 2010. Ontologies to Support RFID-Based Link between Virtual Models and Construction Components. *Computer-Aided Civil and Infrastructure Engineering*, *25*(4), pp. 285-302.

Van den Broek, M., Brederode, E., Ramírez, A., Kramers, L., van der Kuip, M.,Wildenborg, T., Turkenburg, W., and Faaij, A. 2010. Designing a cost effective CO< sub>

2</sub> storage infrastructure using a GIS based linear optimization energy model. *Environmental Modelling & Software*, *25*(12), pp. 1754-1768.

Van Deursen, D. 2010. IFC2X3 ontology. http://multimedialab.elis.ugent.be/organon/ontologies/IFC2X3.

Bansal, V. K., and Pal, M. 2009. Construction schedule review in GIS with a navigable 3D animation of project activities. *International Journal of Project Management*, 27(5), pp. 532-542.

Bansal, V. K., and Pal, M. 2009. "Extended GIS for construction engineering by adding direct sunlight visualisations on buildings." *Construction Innovation: Information, Process, Management*, 9(4), pp. 406-419.

Beetz, J. 2009. Facilitating distributed collaboration in the AEC/FM sector using Semantic Web Technologies. Eindhoven University of Technology.

Eastman, C., Jeong, Y.-S., Sacks, R., and Kaner, I. 2009. Exchange model and exchange object concepts for implementation of national BIM standards. *Journal of Computing in Civil Engineering*, 24(1), pp. 25-34.

El-Gohary, N. M., and El-Diraby, T. E. 2009. Merging architectural, engineering, and construction ontologies. *Journal of Computing in Civil Engineering*, 25(2), pp. 109-128.

Elghamrawy, T., Boukamp, F., and Kim, H.-S. 2009. Ontology-based, semi-automatic framework for storing and retrieving on-site construction problem information-an

RFID-based case study. *Construction Research Congress* ASCE, Seattle, Washington, pp. 457-466.

Kalfoglou, Y. 2009. *Cases on semantic interoperability for information systems integration: practices and applications*, Information Science Reference, Hershey, Pennsylvania.

Karimi, H. A., and Akinci, B. 2009. *CAD and GIS integration*, Auerbach Publications, Boca Raton, Florida.

Nagel, C., Stadler, A., and Kolbe, T. H. 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*Vancouver, Canada, pp. 46-53.

Schlueter, A., and Thesseling, F. 2009. Building information model-based energy/exergy performance assessment in early design stages. *Automation in Construction*, *18*(2), pp. 153-163.

Tserng, H. P., Yin, S. Y., Dzeng, R., Wou, B., Tsai, M., and Chen, W. 2009. A study of ontology-based risk management framework of construction projects through project life cycle. *Automation in Construction*, *18*(7), pp. 994-1008.

Hill, L.L. Georeferencing: 2009. The Geographic Associations of Information; MIT Press: Hill Cambridge, MA, USA; London, UK

Akinci, B., Karimi, H. A., Pradhan, A., Wu, C. C., & Fichtl, G. 2008. CAD and GIS interoperability through semantic web services. *Electronic Journal of Information Technology in Construction.*, *13*, pp. 39-55.

Anumba, C. J., Pan, J., Issa, R. R. A., & Mutis, I. 2008. Collaborative project information management in a semantic web environment. *Engineering, Construction and Architectural Management*. 15(1), pp. 78-94.

Anumba, C. J., Issa, R. R., Pan, J., & Mutis, I. 2008. Ontology-based information and knowledge management in construction. *Construction Innovation. Information, Process, Management*, 8(3), pp. 218-239.

Choi, J. W., Kim, S. A., Lertlakkhanakul, J., and Yeom, J. H. 2008. Developing Ubiquitous Space Information Model for Indoor GIS Service in Ubicomp Environment. *Proc., 4th Int. Conference on Networking Computing and Advanced Information Management*, IEEE, pp. 381-388.

Eskrootchi, R., Karan, E. P., and Sebt, M. H. 2008. "Optimal location of concrete batch plant by GIS." *Proceedings of CSCE 2008 Annual Conference*, pp. 241-250.

Goedert, J. D., and Meadati, P. 2008. Integrating construction process documentation into building information modelling. *Journal of Construction Engineering and Management*, 134(7), pp. 509-516.

Isikdag, U., Underwood, J., and Aouad, G. 2008. An investigation into the applicability of building information models in geospatial environment in support of site selection and fire response management processes. *Advanced Engineering Informatics*, 22(4), pp. 504-519.

Karan, E., and Ardeshir, A. 2008. Safety Assessment of Construction Site Layout Using Geographic Information System." *AEI 2008*, pp. 1-14.

OGC. 2008. Open geospatial consortium. OpenGIS® city geography markup language (CityGML) encoding standard. https://portal.opengeospatial.org /files/?artifact_id=28802.

Sebt, M. H., Karan, E. P., and Delavar, M. R. 2008. Potential application of GIS to layout of construction temporary facilities. *International Journal of Civil Engineering*, *6*(4), pp. 235-245.

Sebt, M. H., Karan, E. P., and Eskrootchi, R. 2008. Layout of construction temporary facilities using GIS. *Proceedings of CSCE 2008 Annual Conference*, pp. 251-260.

Wang, C. C., and Xue, D. 2008. Using domain ontology in a semantic blogging system for construction professionals. *Tsinghua Science & Technology*, *13*, pp. 279-285.

Bakis, N., Aouad, G., & Kagioglou, M. 2007. Towards distributed product data sharing environments—progress so far and future challenges. *Automation in Construction*, *16*(5), pp. 586-595.

Bansal, V. K., and Pal, M. 2007. "Potential of geographic information systems in building cost estimation and visualisation." *Automation in Construction*, 16(3), pp. 311-322.

Cutting-Decelle, A. F., Das, B., Young, R., Case, K., Rahimifard, S., Anumba, C. J., and Bouchlaghem, N. 2007. A review of approaches to supply chain communications: from manufacturing to construction. *Journal of Information Technology in Construction (ITcon)*, 12, pp. 73-102.

Döllner, J., and Hagedorn, B. 2007. Integrating urban GIS, CAD, and BIM data by service-based virtual 3D city models. *Urban and Regional Data Management-UDMS Annual 2007*Stuttgart, Germany, pp. 157-160.

Ergen, E., Akinci, B., and Sacks, R. 2007. Tracking and locating components in a precast storage yard utilising radio frequency identification technology and GPS. *Automation in Construction*, 16(3), pp. 354-367.

Fu, C., Kaya, S., and Aouad, M. K. G. 2007. The development of an IFC-based lifecycle costing prototype tool for building construction and maintenance: integrating lifecycle costing to nD modelling." *Construction Innovation: Information, Process, Management*, 7(1), pp.85-98.

Fuertes, A., Forcada, N., Casals, M., Gangolells, M., and Roca, X. 2007. Development of an ontology for the document management systems for construction. *Complex Systems Concurrent Engineering*, Springer, pp. 529-536.

Kim, H., and Grobler, F. 2007. Building ontology to support reasoning in early design." *Computing in Civil Engineering*, L. S. a. B. Akinci, ed. Pittsburgh, PA, pp.151-158.

Lapierre, A., and Cote, P. 2007. Using Open Web Services for urban data management: A testbed resulting from an OGC initiative for offering standard CAD/GIS/BIM services. *Proceedings of Urban and Regional Data Management. Annual Symposium of the Urban Data Management Society*, pp. 381-393.

Mitchell, J., and Hans Schevers, C. 2007. Adopting BIM for facilities management: Solutions for managing the Sydney Opera House. *Cooperative Research Center for Construction Innovation,* Brisbane, Australia.

Pandit, A., and Zhu, Y. 2007. An ontology-based approach to support decision-making for the design of ETO (Engineer-To-Order) products. *Automation in Construction*, *16*(6), pp. 759-770.

OGC. 2007. OGC web services architecture for CAD GIS and BIM., https://portal.opengeospatial.org/files/?artifact_id=28802.

Studer, R., Grimm, S., and Abecker, A. 2007. *Semantic web services: concepts, technologies, and applications*, Springer.

Alder, M. A. 2006. Comparing time and accuracy of building information modelling to on-screen takeoff for a quantity takeoff of a conceptual estimate. Master of Science, Brigham Young University, UK.

Berners-Lee, T. 2006. Linked data." http://www.w3.org/DesignIssues/LinkedData.html.

Borrmann, A., Van Treeck, C., and Rank, E. 2006. Towards a 3D spatial query language for building information models. *Proceedings of the International Conference of Computing and Decision Making in Civil and Building Engineering (ICCCBE-XI)*Montreal, Canada, pp. 1374-1385.

Cardoso, J., & Sheth, A. 2006. The semantic web and its applications. In *Semantic Web Services, Processes and Applications.* Springer, Boston, MA. Pp. 3-33.

Peachavanish, R., Karimi, H. A., Akinci, B., and Boukamp, F. 2006. An ontological engineering approach for integrating CAD and GIS in support of infrastructure management. *Advanced Engineering Informatics*, *20*(1), pp. 71-88.

El-Diraby, T., Lima, C., and Feis, B. (2005). "Domain taxonomy for construction concepts: toward a formal ontology for construction knowledge." *Journal of computing in civil engineering*, 19(4), pp. 394-406.

Hu, W., Jian, N., Qu, Y., and Wang, Y. 2005. GMO: A graph matching for ontologies. *Proceedings of K-CAP Workshop on Integrating Ontologies* Banff, Canada, p. 41.

Lima, C., El-Diraby, T., and Stephens, J. 2005. Ontology-based optimization of knowledge management in e-construction. *Journal of IT in Construction*, 10, pp. 305-327.

Ma, Z., Shen, Q., and Zhang, J. 2005. Application of 4D for dynamic site layout and management of construction projects. *Automation in Construction*, 14(3), pp. 369-381.

Walker, J., Pan, E., Johnston, D., Adler-Milstein, J., Bates, D. W., and Middleton, B. 2005. The value of health care information exchange and interoperability. *Health Affairs-Millwood VA Then Bethesda MA-*, 24, pp. W5.

Hu, W., Jian, N., Qu, Y., & Wang, Y., GMO. 2005. A graph matching for ontologies. Integrating Ontologies Workshop, K-CAP

GCR, N. 2004. *Cost analysis of inadequate interoperability in the US capital facilities industry*, National Institute of Standards and Technology.

Hayes, J., and Gutierrez, C. 2004. Bipartite graphs as intermediate model for RDF. *Proceedings of the 3rd International Semantic Web Conference, LNCS*, Springer, p. 47

Adachi, Y. 2003, July. Overview of partial model query language. In *ISPE CE* pp. 549-555.

Davies, J., Fensel, D., and Harmelen, F. V. 2003. *Towards the Semantic Web: Ontology-driven Knowledge Management*, John Wiley & Sons, Chichester, UK.

Horrocks, I., Patel-Schneider, P. F., and Van Harmelen, F. 2003. From SHIQ and RDF to OWL: The making of a web ontology language. *Web semantics: science, services and agents on the World Wide Web*, 1(1), pp. 7-26.

Li, H., Kong, C. W., Pang, Y. C., Shi, W. Z., and Yu, L. 2003. Internet-Based geographical information systems system for e-commerce application in construction material procurement. *Journal of Construction Engineering and Management*, 129(6), pp. 689-697.

Lima, C., Diraby, T., Fies, B., Zarli, A., and Ferneley, E. 2003. The e-cognos project: current status and future directions of an ontology-enabled IT solution infrastructure supporting knowledge management in construction. *Construction Research Congress*, pp. 1-8.

Underwood, J., and Watson, A. 2003. An XML metadata approach to seamless project information exchange between heterogeneous platforms. *Engineering,Construction and Architectural Management*, *10*(2), pp.128-145.

Valcik, N. A., and Huesca-Dorantes, P. 2003. Building a GIS database for space and facilities management. *New Directions for Institutional Research*, *2003*(120), pp. 53-61.

Brunnermeier, S. B., and Martin, S. A. 2002. Interoperability costs in the US automotive supply chain." *Supply Chain Management: An International Journal*, 7(2), pp. 71-82.

Cheng, M. Y., and Yang, S. C. 2001. "GIS-based cost estimates integrating with material layout planning." *Journal of Construction Engineering and Management*, 127, pp. 291-299.

Crowther, P., and Hartnett, J. 2001. Handling Spatial Objects in a GIS Database- Relational v Object Oriented Approaches." *Proc., CD Proceedings of the 6th International Conference on GeoComputation, University of Queensland, Brisbane, Australia*, Citeseer, pp. 24-26

Goodchild, M., Egenhofer, M., Fegeas, R., and Kottman, C. 1999. *Interoperating geographic information systems*, Kluwer Academic Publishers, Norwell, Massachusetts.

Bishr, Y. 1998. Overcoming the semantic and other barriers to GIS interoperability. *International Journal of Geographical Information Science*, 12(4), pp. 299-314.

Schraw, G. 1998. Promoting general metacognitive awareness. *Instructional Science*, *26*(1-2), 113-125.

Appendix A

1- JSON OUTPUT FOR THE BERGERON ENGINEERING OBJECT FROM BIM-GIS SEMANTIC INTEGRATED MODEL.


```
{
    "keys": [
        "n"
    ],
    "length": 1,
    "_fields": [
        {
            "identity": {
                "low": 15497,
                "high": 0
            },
            "labels": [
                "IGIM_BuildingElement",
                "IGIM_Door",
                "IGIM_Element",
                "IGIM_Object",
                "IGIM_ObjectDefinition",
                "IGIM_Product",
                "IGIM_Root"
            ],
            "properties": {
                "ownerHistory": "42",
                "overallWidth": "1950.0",
                "globalId": "'3s5$zJ9qPCSQYinqmwssMq'",
                "description": "$",
                "model_id": "EngBuiding_IGIM_2x3_59119",
                "representation": "59112",
                "objectType": "'915/915 x 2350mm'",
                "objectPlacement": "8914584",
                "predefinedType": ".DOOR.",
                "IGIM_id": "59119",
                "overallHeight": "2410.0",
                "userDefinedOperationType": "$",
                "name": "'YU - Door Type B1 (x2), Frame Type 2 (Reverse Offset):915/915 x 2350mm:549971'",
                "operationType": ".DOUBLE_DOOR_SINGLE_SWING.",
                "model": "EngBuiding_IGIM_2x3",
                "IGIM_Class": "IGIM_Door",
                "tag": "'549971'"
            }
        }
    ],
    "_fieldLookup": {
        "n": 0
    }
}
```

2- JSON OUTPUT FOR THE BERGERON ENGINEERING GEOREFEREING INFORMATION FROM BIM-GIS SEMANTIC INTEGRATED MODEL.


```
        "labels": [
            "IGIM_BuildingElement",
            "IGIM_Door",
            "IGIM_Element",
            "IGIM_Object",
            "IGIM_ObjectDefinition",
            "IGIM_Product",
            "IGIM_Root"
        ],
        "properties": {
            "ownerHistory": "42",
            "overallWidth": "1035.0",
            "refLatitude": "43.772406",
            "refLongitude": "-79.506676",
            "globalId": "'1wyIVaneLDD9K8WJFR1si2'",
            "description": "$",
            "model_id": "EngBuiding_IGIM_2x3_59785",
            "representation": "59778",
            "objectType": "'915x2350'",
            "objectPlacement": "8914669",
            "predefinedType": ".DOOR.",
            "IGIM_id": "59785",
            "overallHeight": "2410.0",
            "userDefinedOperationType": "$",
            "name": "'YU - Door Type A, Frame Type 1:915x2350:584149'",
            "operationType": ".SINGLE_SWING_RIGHT.",
            "model": "EngBuiding_IGIM_2x3",
            "IGIM_Class": "IGIM_Door",
            "tag": "'584149'"
        }
    }
]
```

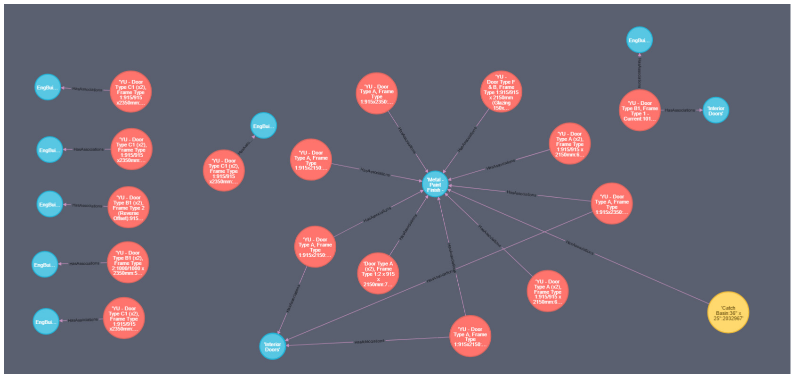
Appendix B

SAMPLE PYTHON CODE TO GENERATE SCENE LAYER FROM MULTIPATCH GEOMETRIES

```
# Set input and output data locations

geodatabase = r'\\data_Hadi\SPK_Automation\EngBuildingYU.gdb'
output_path = r'E:\temp\EngBuildingYU'
upload_to_portal = False

# Set arcpy environment

arcpy.env.workspace = geodatabase

#Handle errors

try:
    # Get list of multipatch feature classes
    feature_list = arcpy.ListFeatureClasses(feature_type="Multipatch")
    print("Found " + str(len(feature_list)) + " multipatch feature classes for processing.")

    # Loop through each multipatch feature and create a scene layer package.
    counter = 0
    for current_feature in feature_list:
        print("Processing feature class: " + current_feature)
        feature_path = geodatabase + os.path.sep + current_feature

        #Create a temporary feature layer from the current multipatch feature class
        arcpy.MakeFeatureLayer_management(feature_path, "temp_in_memory_feature" + str(counter))

        #Save the temporary feature layer into a layer file of extension .lyrx
        output_lyrx_name = output_path + os.path.sep + current_feature + ".lyrx"
        arcpy.SaveToLayerFile_management("temp_in_memory_feature" + str(counter), output_lyrx_name)

        print("    Layer file created at: " + output_lyrx_name)

        #Create a scene layer package for the current feature class
        output_spk_name = output_path + os.path.sep + current_feature + ".spk"
        arcpy.CreateSceneLayerPackage_management(output_lyrx_name, output_spk_name)

        print("    Successfully created scene layer package at: " + output_spk_name)

        counter = counter + 1
        #Upload scene package to portal if required
        #Note - this section would work only if ArcGIS Pro is started and signed into the desired
        #portal
        if (upload_to_portal):
            arcpy.SharePackage_management(output_spk_name)

except Exception as spk_ex:
    print("Error processing. Details: " + str(spk_ex))
    print(arcpy.GetMessages())
```

Appendix C

- Doors classifications from Neo4J Graph database



- Extracting Bergeron building stories using graph query.