DESIGN AND EVALUATION OF A BIM-GIS INTEGRATED INFORMATION MODEL USING RDF GRAPH DATABASE

A-Hadi Hor[1,*], and Gunho Sohn[1]
Department of Earth and Space Science and Engineering, York University.
4700 Keele Street, Toronto, ON M3J 1P3, Canada; elhadi@yorku.ca, gsohn@yorku.ca.

KEY WORDS: BIM, GIS, Information, Modelling, RDF, CityGML, IFC, Graph, Semantic, Databases, Design, Neo4J, SPARQL, BIMServer.

ABSTRACT:

The semantic integration modeling of BIM industry foundations classes and GIS City-geographic markup language are a milestone for many applications that involve both domains of knowledge. In this paper, we propose a system design architecture, and implementation of Extraction, Transformation and Loading (ETL) workflows of BIM and GIS model into RDF graph database model, these workflows were created from functional components and ontological frameworks supporting RDF SPARQL and graph databases Cypher query languages. This paper is about full understanding of whether RDF graph database is suitable for a BIM-GIS integrated information model, and it looks deeper into the assessment of translation workflows and evaluating performance metrics of a BIM-GIS integrated data model managed in an RDF graph database, the process requires designing and developing various pipelines of workflows with semantic tools in order to get the data and its structure into an appropriate format and demonstrate the potential of using RDF graph databases to integrate, manage and analyze information and relationships from both GIS and BIM models, the study also has introduced the concepts of Graph-Model occupancy indexes of nodes, attributes and relationships to measure queries outputs and giving insights on data richness and performance of the resulting BIM-GIS semantically integrated model.

1. INTRODUCTION

Considering the large number of applications and the nature of information in the BIM and GIS domains Over the recent years, the usage of se- mantic web technologies and RDF data models have been notably increased in many fields, and multiple knowledge-based frameworks are started using RDF data to describe information resources semantic associations with complex relationships. RDF data plays a very important role in advanced information retrieval, and graphs are efficient ways to visualize and represent real-world data by providing solutions to many real-time scenarios that can be simulated and implemented using graph databases and to efficiently query graphs with multiple attributes representing a different source of datasets coming from many do- mains of knowledge. Given those graph databases are schema-less with efficient storage for semi- structured data, they can provide fast and deep traversals instead of slow RDBMS SQL based joins allowing Atomicity, Consistency, Isolation, and Durability (ACID) transactions with rollback support. The common graph databases include Neo4j; this open-source graph database implemented in Java employs mathematics of graph and utilizes its enormous potential for fast in- formation extraction and storage of information in the form of nodes and relationships. In this paper, we are presenting an architectural design with complete implementation of the BIM-GIS integrated RDF graph databased workflows describing different phases of the construction of the BIM-GIS semantic integrated RDF model on Neo4j instance. The proposed integration pipeline phases design and implementation are discussed in the following sections:

- Conceptual BIM-GIS Ontological Model Construction Model, the mapping process, and the introduction of the concept of the BIM-GIS sematic bridge.
- Translation workflow design: This section will discuss the data flow from BIM and GIS sources models to the BIM-GIS integration model.
- CityGML to RDF translation workflow: Describe each of the components used in this workflow from parsing module and data repository builder for geometrical and thematical triples from the GIS model.
- IFC to RDF translation workflow: this section is providing an overview of the most critical components of components used in the semantic extraction, translation and transformation of IFC objects and their properties from their native format (Revit model) to conversion module to developing RDF repository using ontology engine protégé.
- SPARQL query design: This section will describe the design of SPARQL queries leading to create the sub-models to be loaded into the Neo4j graph database.
- Cypher filtering: Once the sub-model is loaded using Neo4j import/export tools, different queries to extract and visualize the data are developed using the Neo4j desktop or web console.
- Test data, experimentation, and evaluation: In this section we designed and executed a series of experimentation on the resulting BIM-GIS semantic model, also we have introduced Graph-Model occupancy index for nodes, attributes, and relationships, these indices will give insights on the information richness of the BIM-GIS semantic model, and comparison between models was used to evaluate differences Interoperable data formats, then an import into a graph database with querying and filtering capabilities. The workflows and transformations of IFC and CityGML schemas into object graph databases model are developed and applied to an intelligent urban mobility web application to validates the integration methodology

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-VIII-4-W2-2021-175-2021 | © Author(s) 2021. CC BY 4.0 License.

2. BACKGROUND

The framework and workflows described and presented in this research paper were developed using best designed semantic integration method- ologies and products available to ensure the highest integration between BIM and GIS platforms. This is achieved by applying semantic web RDF data modelling supported and maintained by a graph database system fully optimized to pro- cess BIM-GIS integrated models. This study is the first research that proposes and parallel de- signed workflows in bringing BIM and GIS ontologies into a unified semantic model using the simplicity of RDF models representation and a schema-less graph model maintained in a graph database, most of the previous research works have focused on BIM and GIS data conversion and transformation. For instance, Wu and Hsieh (2007) proposed an approach to transform the geometric information of an IFC model into geometric objects for the GML model. Döllner and Hagedorn (2007) utilized the data integration capabilities of IFC to CityGML transformation to integrate data from CAD, GIS, and BIM applications into a virtual 3D city model. Nagel et al. (2009) provided conceptual requirements for the automatic transformation of IFC geometry to the different levels of detail in CityGML. Laat and Berlo (2011) developed an extension to map IFC classes and their properties to CityGML models and attributes. Hijazi et al. (2011) presented an approach for mapping information from IFC to CityGML to model interior utilities within a GIS context. Although these approaches have provided a range of means to enable data translation between BIM and GIS platforms, their applications are limited to solve interoperability problems at the syntactic level which can- not guarantee that an integrated BIM-GIS model would reflect the meaning of the data from either domain given the inconsistencies between the CityGML levels of details and IFC levels of developments, this has led to investigating approach of the usability, the implementation of a an ontological integration framework for BIM and GIS which provides ability to organize, and structure of objects from these two domain knowledge resulting the highest level of integration possible, which that cannot be achieved without using an adequate data modelling standard, and this project Resource Description Framework(RDF) was adopted and used to bright the IFC and CityGML translation pipelines together. While there are many conventional tools for dealing with data objects attributes and interrelation- ships, RDF is the most simplistic, most powerful standard designed by now, it is mature, robust technology for modelling data and provides a foundation for publishing and linking data. Even that RDF was initially part of the semantic web stack, currently it is used in all conceivable ways where is a need for representing high quality connected data like the ones we get from BIM and GIS systems. An RDF graph database can be used for managing both unstructured and structured data alike; they have a minimum of alternatives for capturing, storing and manipulating data that can be consumed by applications, systems, and other different information and analytics systems, having a BIM-GIS semantic graph database that uses RDF will help to : 1) Construct a data-centric infrastructure.2) Provide an integrated view of BIM-GIS data using ontological liked data principles. 3) Establish a new modeling information standard for Architecture, Engineering and Construction (AEC) compliant with W3C specifications. 4) Create new methodologies to unlock and analyze knowledge patterns of data from BIM and GIS domains. 5) Creates a unified metadata layer for BIM- GIS oriented applications that facilitates identification, integration, and discovery. 5) Which can lead to an integration workflow which can offer an easier knowledge exploration and navigation, a rich semantic BIM-GIS context and provides an ab better discovery ability of semantic content of unified BIM-GIS model.This paper distinguished itself from others by proposing a detailed description, implementation, and experimentation to validate the usability of a BIM-GIS integrated model based on semantic integration using ontology workflows by designing an integrated BIM-GIS graph database model around resource description framework (RDF) using ontology construction and mapping tools.

3. BIM-GIS INTEGRATED ONTOLOGY BASED MODELLING

One of the most challenging problems consists in integrating different models into a unique conceptualized model of a specific knowledge or ap- plication domain, it is a hard task to accomplish given the inconsistencies and heterogeneities between the source models at different levels and would affect the output model representation, information retrieval, and management greatly. However, it also provides an effective solution to the problem on standardizing the information and to building a robust semantic web framework that can be used to support different decision support systems, this research work has looked at using the data representation schemas and properties from BIM and GIS domains then constructing a semantic bridge between them in order to develop one unique knowledge base model consisting a new set of objects and well-defined properties. The workflow of creating the BIM-GIS semantic bridge must take into considerations to encode spatial and non-spatial information from BIM and GIS objects separately before using the semantic schema to bring them together into one ontology model. The information from source models will mainly consist of instance, data properties, types, relationship, and relationships types, the information of similar objects or elements from BIM or GIS will be filtered through the graph matching for ontologies algorithm process into the new BIM-GIS integrated model into a unified model representation with a unique class object, its inherited sub-class, instances, properties and relationships in which the attributes and values of the object are either from BIM or GIS, or from one model enhanced using additional information from the other model and the relationships forming an adequate representation of a new BIM-GIS integrated knowledge can be used afterward to construct a graph model using a bulk import data process into a managed graph database, by using graph database to store, manipulate and query RDF data we are in fact able to have 1) A BIM-GIS semantic graph model will enable users to model data exactly as represented in a real-world scenario, 2) BIM-GIS semantic graph can significantly enhance the operation of data and keep all the information about BIM, GIS objects into a single node and display related in- formation by using relationships connected to it, 3) Attribute and spatial related queries can be developed based on the graph structure, 4) Storing the graph into a graph database management system will allow using a functional graph and knowledge discovery algorithms easier and more efficiently.

In summary, the BIM-GIS semantic graph database model will offer to display information about data interconnectivity or topology that facilitates deep data analysis , grouping by aggregating queries, combine and hierarchize multiple dimension when additional data sources or information layers are added into the semantic integrated model, many academic and industry research-oriented studies have also suggested that ontology-based graph database will constitute a great Artificial intelligence (AI) infrastructure due to well-structured relational information be- tween graph entities in providing rich information and convenient data accessibility that other data models can't and hardly provide.

3.1 Conceptual Model

The system architecture for BIM-GIS integrated semantic model using RDF representation to a graph database platform on Neo4j is represented in the diagram in Figure 2, It consists of four modules 1- **Data sources**: Consisting of BIM and GIS models in IFC and CityGML formats, respectively. 2- **Data translation**: From native formats followed with an RDF graph matching using Semantic Alignment Techniques. 3- **BIM-GIS integrated model**: management console and providing a user interface to store, maintain, and manage Graph datasets and 4- **Application**: This is a web-based application hosted using web-based services hosted on a cloud platform.

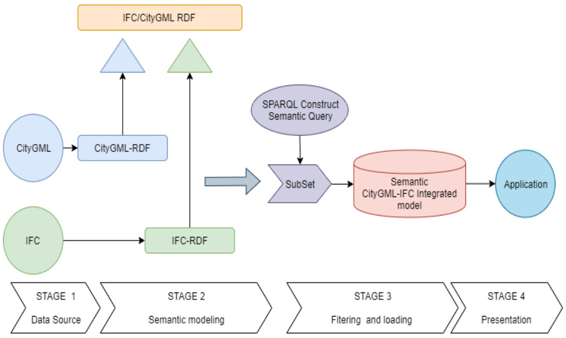


Figure 2: BIM-GIS integration data workflow diagram

The transformation of the CityGML data to the RDF model is completed through a direct mapping to a designated class in the ontology tar- get model; every element is mapped to one class. However, for building, the shape is extracted and converted to the geometry class based on the de- signed implementation of SPARQL query from the subset. BIM and since the structure of IFC is very complicated; therefore, its RDF based model will be the same, the transformation process needs intermediate parsing and translation steps. In this work, we have established the data mapping between the source IFC model of the source data from the BIM model and building ontology for data transformation. Most of the IFC classes are one-to-one mapping. However, we had some special cases. For instance, an IFC instance could be a roof, a floor or a stair, so it was categorized into different classes in the semantic integrated model Building ontology based on its predefined type value in the BIM-IFC semantic model, some other instances like the **IfcCurtainWall** does not have much useful information and serves only as a container class and a reference to multiple curtain panels. Therefore, it was omitted in a new integrated ontology model.

Now, once the objects of the integrated model are imported into a graph model, the attributes of the same or similar objects will be enhanced with each other attributes, topological and geometrical properties like described for the **ifcSlab** from BIM and Slab from GIS.

When it comes to querying or extracting specific datasets or a sub-model, queries have must use a conjuncture of attributes properties uniquely identified by a unique key GUID and join that to associated geometries through the same GUID, using such configuration mutual in- formation could be extracted and exported as one dataset (using same data format) as illustrated in figure 3.

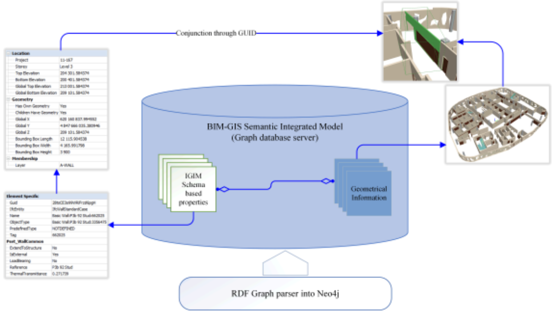

Conjunction through GUID
BIM-GIS Semantic Integrated Model
(Graph database server)
IGIM Schema based properties
Geometrical Information
RDF Graph parser into Neo4j

Figure 3: Object linkage between properties and associated geometries

The mapping of IFC and CityGML into Meta Graph Model representing all IFC classes and CityGML elements, their attributes, and relation- ships are completed through Cypher scripts with LOAD CSV clauses and commands to validate data quality. Once all classes and attributes are mapped into nodes and connected using various relationships like ' *has_property* ' to connect a class node with direct attributes or ' *subtype_of* ' to join the class with its corresponding element in CityGML and with its sub-class resulting into a complete integrated graph model with all classes and features stored a unified property graph in which the relationships for referenced, inversed and derived attributes for IFC classes will be created automatically including geospatial relationships from CityGML elements, the next step will consist in establishing relationships between graph nodes and their connected information and dropping redundant relationships in the unified, integrated model and using property normalization and non-direct attributes of BIM classes and GIS elements of the same object can be assigned as direct nodes attributes, further each node (object) can be assigned a set of labels of its parent class/element, them running pre-defined queries for classification and normalization of objects rep- resenting building elements with accordance to their object type and property sets.

The following section in this paper, will be going through the details of each workflow:

3.2 Workflow Design

3.2.1 CityGML to RDF Translation Workflow

Converting 3D geospatial data from CityGML to Resource description framework (RDF) im- plies representing not only their thematic proper- ties (which can be handled like any other data) but also their associated geometries in RDF, Geotools libraries enables geospatial data con- version into RDF. Geotools is an open-source Java library that provides methods for manipulating geospatial data. It includes many features and support for a wide number of GIS formats like XML, GML, KML, and Esri Shapefiles, with co- ordinate reference systems descriptions and operations on coordinates such as datum transformations and map projections. The conversion process takes CityGML (extracted from the BIM model using FME workspace) and generates RDF triples; the geometries associated with geographic features will be represented in RDF as structured geometries like GeoVocab libraries 3 for GML and XML.

The Translation pipeline (figure 4) extracts from the CityGML file all the features and their description including schema, types, and coordinate reference system defined by the Open Geospatial consortium standard to have both spatial and non- spatial attributes.

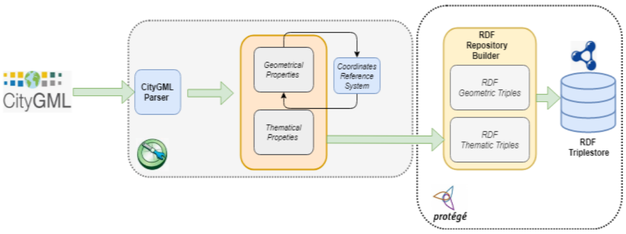


Figure 4: CityGML to RDF translation workflow

The RDF Repository builder module generates a collection of RDF triples represented as subjects, predicates, and objects. The thematic and geometric properties are processed separately. By default, the pipeline generates predicates by reusing properties names, they can be replaced by dataset vocabulary, the matching task with de- fined ontology predicates is handled by the RDF platform used, in this case, Stanford protégé.

3.2.2 IFC to RDF Translation Workflow

The diagram shown in Figure 5 is showing a complete IFC model to RDF translation workflow from loading the BIM model from the native format as an Autodesk Revit project to exporting the model (or the subset of the model) into an RDF triple-store to be stored into a Neo4j graph database used in the system implementation. The parsing process consists of selecting a specific building element (based on the target application) and executes the IFC-to-RDF conversion and, ultimately, a file schema validation. All geometrical properties and thematical attributes will be linked based on a unique identifier or a GUID to keep the relationships and class hierarchy between the triples as close as possible to the pre-parsed IFC elements.

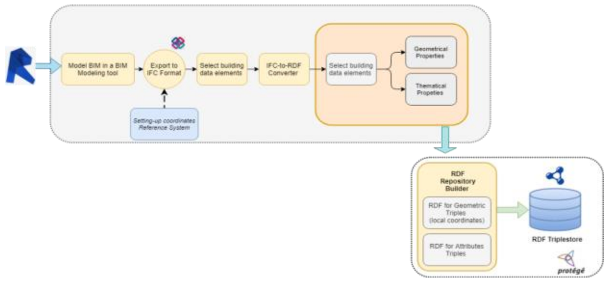

Model BIM in a BIM Modeling tool
Export to IFC Format
Select building data elements
IFC-to-RDF Converter
Select building data elements
Geometrical Properties
Thematical Propeties
Setting-up coordinates Reference System
RDF Repository Builder
RDF for Geometric Triples (local coordinates)
RDF for Attributes Triples
RDF Triplestore
protégé

Figure 5 IFC to RDF translation workflow

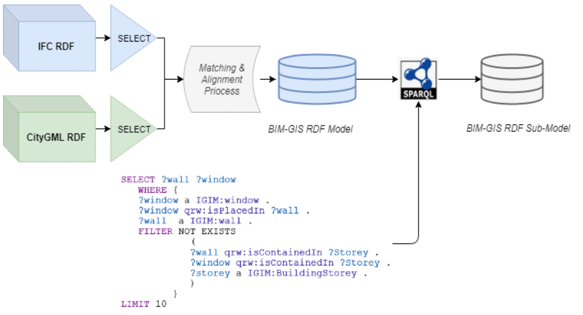
3.2.3 SPARQL Query

Given that BIM and GIS entities are captured as sub-layers of spatial objects in the IFC and CityGML structural trees respectively, this geometry classification is crucial when designing and creating the SPARQL queries and using graphs with extended geometric and spatial relations would require knowing the good data structure describing these domains of knowledge, otherwise it can also be very computationally intensive.

SPARQL language was used to query the BIM-GIS integrated RDF Model resulting from semantic integration developed in chapter and shown in Figure 5, the results were populated into a repository indicated in the Figure below as BIM- GIS Sub-Model, the query in this case will return all the doors including information about the ID, name, exposure the floor, and associated material layers (sample results represented in JSON at the end of the paper).

3.2.4 Neo4j and Cypher Filtering

After Converting, exporting the resulting sub- model into the RDF triple-store, the BIM-GIS elements are loaded into a graph database using import/export (ETL tools) to populate the graph database objects. We developed some scripts to extract the BIM-GIS data (object, relationships) and all necessary Cypher statements for the data import and relationship generation. The scripts include an option to include/exclude certain elements (IFC classes and/or CityGML) from the integrated model to satisfy the output and the target application (figure 6).



```
SELECT ?wall ?window
   WHERE {
   ?window a IGIM:window .
   ?window qrw:isPlacedIn ?wall .
   ?wall  a IGIM:wall .
   FILTER NOT EXISTS
             (
             ?wall qrw:isContainedIn ?Storey .
             ?window qrw:isContainedIn ?Storey .
             ?storey a IGIM:BuildingStorey .
             )
          }
LIMIT 10
```

Figure 6: SPARQL Query of BIM-GIS sub-model

The Integrated Information Model data is extracted and saved in CSV format before imported into Neo4j to create our graph-based model as illustrated in Figure 7.

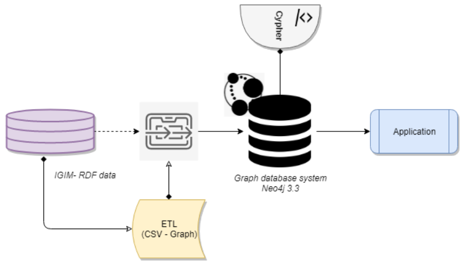


Figure 7: ETL process from the RDF model to Graph- based database

The query below represents the spaces associated with level 3 and level 2 of the Bergeron building-integrated model.

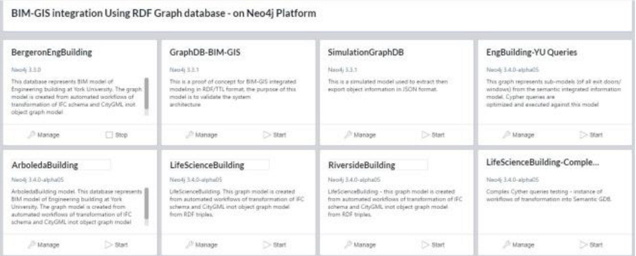
4. SYSTEM COMPONENTS

The high-level system architecture for a complete BIM-GIS integrated semantic model from an RDF representation to a graph database platform on Neo4j is represented in the diagram in Figure 1, mainly the system consists of an ex- traction module of sources datasets from IFC and CityGML models and IFC and CityGML models which are parsed using Apache Jena into RDF data. The Apache Jena is an open-source semantic web framework for Java; it provides a rich and well-documented Application programming interface (APIs) to extract and write to RDF graphs, this framework includes:

- **RDF API**: To read and write RDF in RDF/XML, Turtle, N3 triples, data in any of these formats can be stored in memory and used on runtime or can be stored in databases.
- **ARQ Engine**: This is a query language that supports SPARQL.
- **TDB Engine**: Can be used as a high-performance store on a single machine, a TDB store can be accessed and managed with command lines scripts and via Jena API.
- **Apache Jen Fuseki**: This is the SPARQL server, it is tightly integrated with TDB to provide a robust, transactional persistent storage layer.

After importing the integrated model into Neo4j graph database, a management console shown in Figure 8. (also available as a web service) can be used to run queries for data retrieval, advanced analysis and graph data mining algorithms of BIM-GIS integrated graph model, these queries could be ranging 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's information and indoor navigation paths.

To test and validate the BIM-GIS integrated RDF graph-based model, we created four databases instances all running on the same Neo4j Server version 3.3.0 (enterprise) and Neo4j browser version 3.2.17, and each one of these in- stances described below was used to demonstrate some aspects of the integration methodologies developed in this research.



BIM-GIS integration Using RDF Graph database - on Neo4j Platform

BergeronEngBuilding
Neo4j 3.3.0
This database represents BIM model of Engineering building at York University. The graph model is created from automated workflows of transformation of IFC schema and CityGML inot object graph model

GraphDB-BIM-GIS
Neo4j 3.3.1
This is a proof of concept for BIM-GIS integrated modeling in RDF/TTL format, the purpose of this model is to validate the system architecture

SimulationGraphDB
Neo4j 3.3.1
This is a simulated model used to extract then export object information in JSON format.

EngBuilding-YU Queries
Neo4j 3.4.0-alpha05
This graph represents sub-models (of all exit doors/ windows) from the semantic integrated information model. Cypher queries are
optimized and executed against this model

ArboledaBuilding
Neo4j 3.4.0-alpha05
ArboledaBuilding model. This database represents BIM model of Engineering building at York University. The graph model is created from automated workflows of transformation of IFC schema and CityGML inot object graph model

LifeScienceBuilding
Neo4j 3.4.0-alpha05
LifeScienceBuilding. This graph model is created from automated workflows of transformation of IFC schema and CityGML inot object graph model from RDF triples.

RiversideBuilding
Neo4j 3.4.0-alpha05
LifeScienceBuilding - this graph model is created from automated workflows of transformation of IFC schema and CityGML inot object graph model from RDF triples.

LifeScienceBuilding-Comple...
Neo4j 3.4.0-alpha05
Complex Cyther queries testing - instance of workflows of transformation into Semantic GDB.

Figure 8: Neo4j admin console showing model and sub-models created during the system validation phase

- **EngBuildingIFC**: This instance represents the BIM Model of the engineering building at York University; the graph model was created from automated transformation workflows of the IFC schema object graph from the parsed RDF triples.
- **GraphBIMGIS**: This instance was used as a proof of concept for the integrated RDF graph model, this model was also used for data import then extraction of data using RDF/TTL, it also served to validate the system design architecture.
- **SimulationGraphDB** : This instance was created to extract export objects from BIM- GIS integrated graph model into JSON format, it also been used to validate the conjuncture using GUID between attribute properties and geometries object definitions.
- **EngBuilding-YU-Queries**: This instance was launched to extract sub- models of specific objects (doors, windows, stairs . . . or combined) from the BIM-GIS integrated RDF graph model, queries using Cypher language were designed and tested, we also have conducted an optimization query exercise.
- **Other Neo4j GDB instances**: These are instances created to validate the testing and to evaluate system performance using complex Cypher queries against BIM-GIM semantic models.

Using Neo4j import/export capabilities of graph data to multiple file format such as JSON or CSV formats, this tool provides high data transfer rates of around **1,000,000 record/second**, providing high performance in transferring data is achieved by adding individual layer until the entire information is extracted and moved, and the data store is entirely built. Data obtained can be used to create nodes or relationships data in the import tool by applying either this command in the Neo4j database (Figure 11).

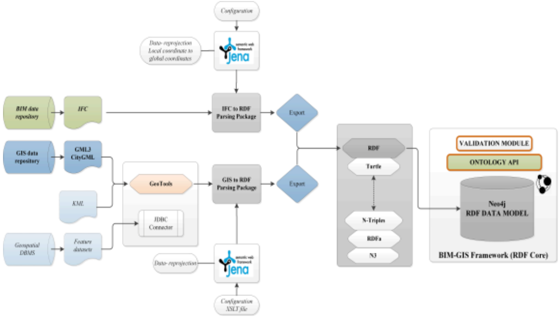


Figure 11: High-level system architecture

It has been used as an input facilitating connection to the BIMServer, using BIMServer JSON API Through BIM Server, the data of the integrated semantic model represented in JSON can be streamed to any of these file formats 3D Collada, ifcXML, KML, etc. or as web REST or SOAP services such as REST or SOAP services as shown in Figure 13.

Using BIMserver, the data that comes in, is stored as objects in the database i.e. we do not have to translate to another internal model making it possible to build import/export utilities at this tier in the system architecture, the database is also accessible for all the other plugins (serializers, services) and open for development using APIs. For the sake of this proof of concept, the model is designed to stream out to an IFC based schema for the purpose of comparison in order to check the number of the classes that have been carried out during the workflow and also to extract the multi-patch geometries used the re-construct and create scene Layer Package (SLPK), the model on ArcGIS platform using ArcGIS Pro 2.2.1 APIs.

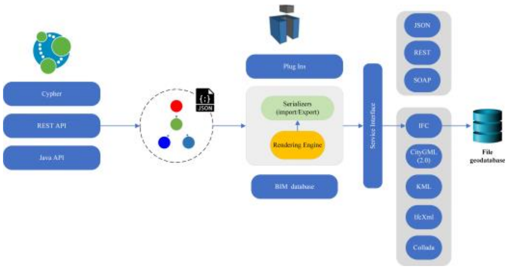


Figure 12. BIMServer Serialization plugins BIM Server components

The georeferencing SDKs from ArcGIS Pro were used to create a coordinate transformation file that the ArcGIS Pro will read

https://doi.org/10.5194/isprs-annals-VIII-4-W2-2021-175-2021 | © Author(s) 2021. CC BY 4.0 License.

and use to apply the appropriate offsets each time the data is loaded into the integration platform, the transformation file called WLD3 (3D World files) in an ASCII file containing sets of to-and-from points that define the 3D coordinate transformation, the collection of georeferencing tools are provided to help generate the file to give the data location.
The resulting SLPK package is then published as a web scene layer on the ArcGIS On-line platform and ArcGIS Server and consumed into a web or a Unity3D simulation application for intelligent urban mobility.

The table 1 summarizes a few of the attributes selected for export from the BIM- GIS integrated model (in Neo4j tabular format); these attributes are initially from CityGML elements and /or from IFC object classes.

5. TESTING DATASETS

The datasets used are BIM and GIS models, which we have pre-estimated; the number of triples and number of nodes of each of the models was tested in this implementation.

For the datasets (Figure 13), we used BIM and GIS models, which we have estimated the number of triples and number of nodes of each of the models tested in this implementation using SPARQL and Cypher scripting.

Our testing data consists of the following models:

- DC Riverside Building from Nemechek Vectorworks Inc.,
- Arboleda building in Santo Domingo, Dominican Republic
- Bergeron Engineering building from York University (Toronto, Canada)
- Life Science Building from York University (Toronto, Canada)

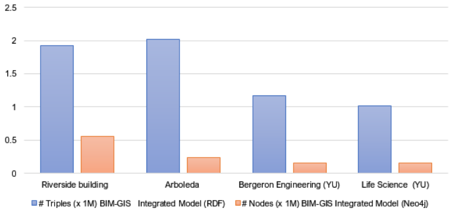


Figure 13: Testing datasets

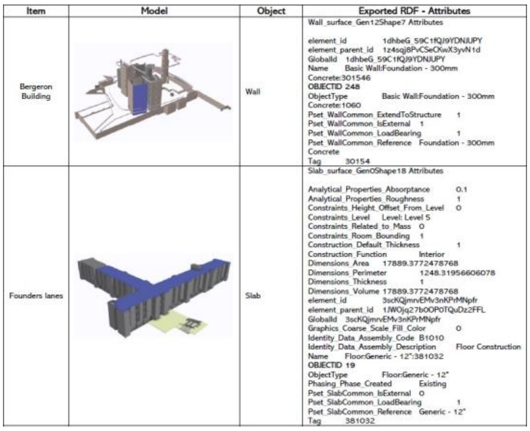
The first two models were used to implement a BIM workflow using *Vectorworks software*, and they are along with the two other models provided by CSBO — facilities development at York University in this research project to demonstrate BIM-GIS semantic integration

6. RDF GRAPH DATABASE PERFORMANCE VALIDATION

For evaluating the system proposed in this re- search project, an environment was simulated in which semantic GeoSPARQL and Neo4j Cypher queries are executed and measured on selected test datasets, these testing will allow us in determining the usability of our BIM-GIS semantic integration approach using ontology modelling methodologies on a graph database platform, the performance testing designed experiments implemented and conducted in this section are using the four models described above.

The investigations documented below are looking to measure and evaluate aspects of the BIM-GIS integration semantic graph model using ontologies.

- Integrated model Data richness.
- System resources and computational efficiency.
- Cypher querying accuracy.

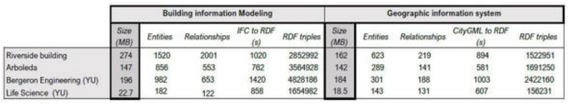


Item	Model	Object	Exported RDF - Attributes
Bergeron Building		Wall	Wall_surface_Gen12Shape7 Attributes element_id 1dhbeG_59C1fQJ9YDNJUPY element_parent_id 1z4sqj8PvCSeCKwX3yvN1d GlobalId 1dhbeG_59C1fQJ9YDNJUPY Name Basic Wall:Foundation - 300mm Concrete:301546 OBJECTID 248 ObjectType Basic Wall:Foundation - 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

Table 1: RDF exported attributes from BIM-GIS Semantic Integrated Model

6.1 Experiment 1: Nodes, Relationships and Properties Occupancy Indexing

In this experiment, we extracted all information of our testing models using both SPARQL, Cypher, and Neo4j administration console. The results are all summarized in Table 2. and shown in Figure 19 graphs. The diagram in Figure 18 shows clearly that number of triples from the BIM models a lot higher compared to the ones from GIS models (before exporting the model into graph database system), same trend when it comes to the numbers of nodes (after export into graph database system) which explains the need for semantic integration between these two domains.



	Building information Modeling					Geographic information system				
	Size (MB)	Entities	Relationships	IFC to RDF (s)	RDF triples	Size (MB)	Entities	Relationships	CityGML 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	1691250
Bergeron Engineering (YU)	196	982	653	1420	4828186	184	301	188	1003	2422160
Life Science (YU)	22.7	182	122	858	1654982	18.5	143	131	607	156231

Table 2: Testing models

The ***sysinfo*** command executed from Neo4j admin console to the models have returned the following graph information, including the number of nodes, relationships, and properties (attributes and values).

To evaluate the graph collection of nodes, relationships (edges) and attributes (values) occupancy in a BIM-GIS integrated instance, we have defined new indexes to express, evaluate and measure the occupancy of each model (BIM or GIS) nodes, relationships and properties within RDF integrated graph instance. The figures 14, 15 and 16 below are showing models information, calculated and visualized defined indexes.

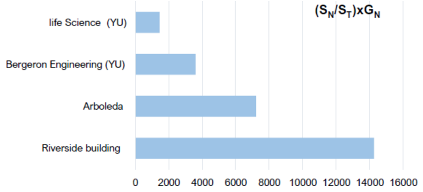


Figure 14: *(S_N/S_T).G_N* - Graph-Model Nodes occupancy

https://doi.org/10.5194/isprs-annals-VIII-4-W2-2021-175-2021

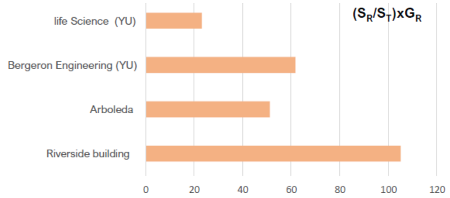


Figure 15: $(S_R/S_T).G_R$ - Graph-Model Relationships occupancy

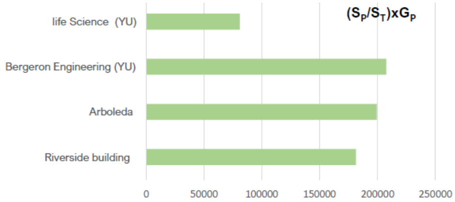


Figure 16: $(S_P/S_T).G_P$ - Graph-Model properties occupancy

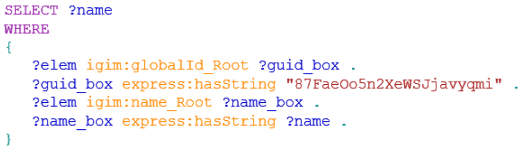
From the comparison of occupancy indexes for each one of the models, several observations can be made, firstly, note that even Bergeron building comes second in file size it is ranked first when it comes properties occupancy index, second in relationships occupancy index, and third in nodes occupancy index, which explains that Bergeron model was the most semantically rich and documented model, given that this building was de- signed to be part of a university, each element (space, door, window . . .) was well documented in the source data in its native format (Autodesk Revit).

6.2 Experiment 2 – Query Performance testing.

6.2.1 RDF-SPARQL Query Performance.

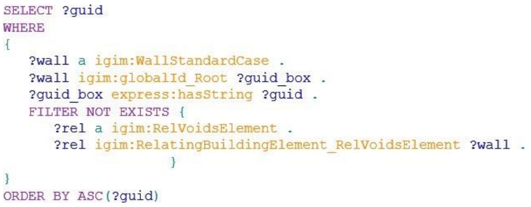
The purpose of this experiment and s to evaluate query execution against RDF model, we have designed and executed many queries ranging from simple to complex on all the testing models, in the test, besides query engine ARQ query engine for Jena (supporting SPARQL RDF Query language) 10 no reasoner was used. Our testing environment is Windows 10 Pro 64-bit machine with Core i7 CPU, 1.60 GHz, and 16 GB of RAM. We show the performance of queries containing and no containing functions related to geometries.

Query 1: *Find the name of an element identified by a specific GUID*



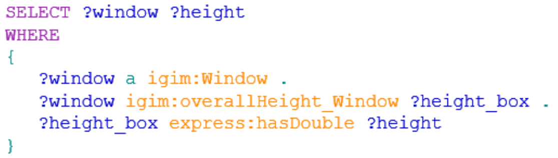
```
SELECT ?name
WHERE
{
   ?elem igim:globalId_Root ?guid_box .
   ?guid_box express:hasString "87FaeOo5n2XeWSJjavyqmi" .
   ?elem igim:name_Root ?name_box .
   ?name_box express:hasString ?name .
}
```

Query 2: *This query will find all the walls without opening*



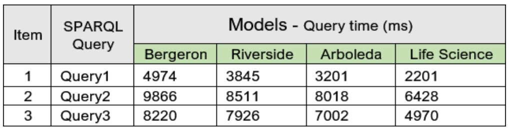
```
SELECT ?guid
WHERE
{
   ?wall a igim:WallStandardCase .
   ?wall igim:globalId_Root ?guid_box .
   ?guid_box express:hasString ?guid .
   FILTER NOT EXISTS {
      ?rel a igim:RelVoidsElement .
      ?rel igim:RelatingBuildingElement_RelVoidsElement ?wall .
                  }
}
ORDER BY ASC(?guid)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

Query 3: *This query will return all the windows heights*



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

The performances of Query2 and Query3 in Table 3. are related to both ARQ engine and geometry engine, and we can see from table 3. that Query2 has a considerably long time for query processing. It is expected because this query needs to calculate ***n-wall * m-opening*** times to return results when there are n-walls and m-openings contained in the model. In further implementations, optimizations are required, and additional 3D geometry libraries might be used.



Item	SPARQL Query	Models - Query time (ms)			
		Bergeron	Riverside	Arboleda	Life Science
1	Query1	4974	3845	3201	2201
2	Query2	9866	8511	8018	6428
3	Query3	8220	7926	7002	4970

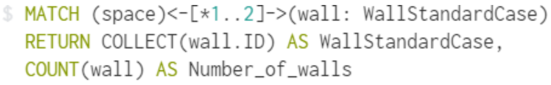
Table 3: SPARQL query evaluations

6.2.2 Neo4j-Cypher Query Performance.

In this section, we looked at the performance at the graph database for each model imported from the RDF files, the testing environment consists of Neo4j 3.4.0-alpha05 and six graph databases. The following plugins were installed and enabled: a) **APOC***:* This library consists of about 300 procedures to help with many different tasks in areas like data integration, graph algorithms and data conversion.
b) **GraphQL-Endpoint** : An extension for Neo4j translates GraphQL queries and mutations into Cypher and executes them on Neo4j, it offers HTTP API, as well as Cypher procedures, even this plugin was not used, it provides background support to queries using embedded sub-queries, c) **GRAPH ALGORITHMS***:* this library pro- vides parallel versions of common graph algorithms for Neo4j 3.x exposed as Cypher procedures.

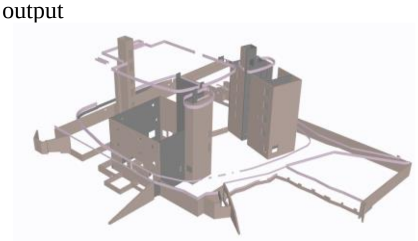
We designed our Cypher queries to investigate different performance aspects of the integrated model, and outputs are expressed in the table as well in a form on a graph.

Query 1: This query will find all walls in the Model.

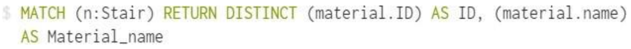


```
$ MATCH (space)<-[*1..2]->(wall: WallStandardCase)
  RETURN COLLECT(wall.ID) AS WallStandardCase,
  COUNT(wall) AS Number_of_walls
```

Resulting output



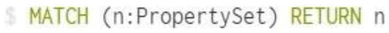
Query 2: This query will return all the stairs in the model.



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

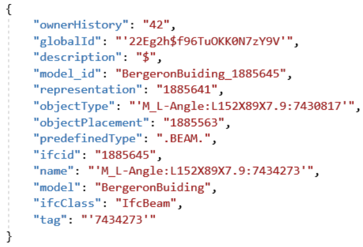
Resulting this graph:

Query 3: Returns all existing properties of all elements in the Model.



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

Resulting into this JSON file of all existing properties of all elements in the model:



```
{
    "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'"
}
```

These queries and many other ones were executed against all the testing models, table 6.6 and the diagram (Figure 28) below summarize their execution times.

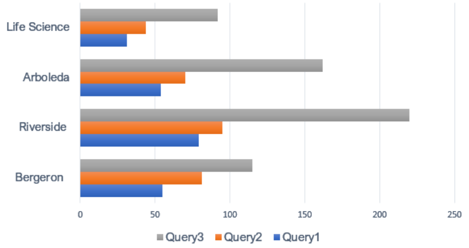


Figure 20. Cypher queries evaluations (in ms)

Although the initial results of response times obtained for the SPARQL queries on the RDF model were not good enough given the structure of the RDF schema, on the contrary the Cypher queries were reasonable compared to the volume of information stored into the Neo4j graph database instance, it shows that execution times for complex queries are quite reasonable given that we are using a single Neo4j instance. It is evident that the choice of the server, the graph database and its configuration are important and since the graph is going to get massive over time, it is recommended using the indexing to make research faster. Through all the experiments we conducted, we realized that without indexing techniques, Neo4j shows excellent data retrieval performance

7. DISCUSSION AND CONCLUSION

In this paper, we have implemented, investigated and analyzed workflows for a BIM- GIS semantic integrated model using RDF graph database, this study can be well-used as benchmark baseline for future works in the areas on integrating data sources to BIM-GIS RDF graph databases, there are many applications that could benefit significantly from these workflows design, query evaluation and the management of IFC and CityGML objects attributes, their complex relationships. This research effort is an emphasis on the potential use of graph structure specifically for large, complex, unstructured diverse and dynamic RDF datasets. The major characteristic of RDF graph structure include flexibility, schema-free representation of data models Overall, Neo4j can be used as a high-performance database platform especially when handling highly interconnected data as in our BIM-GIS semantic integrated model, and can yield very good Cypher queries performance on datasets as big and rich such as IFC and CityGML, based on the test experiments, the initial results of response times obtained for the SPAQRL queries on the RDF model were not good enough given the structure of the RDF schema , on contrary the Cypher queries were good compared to the volume of information stored into the Neo4j graph database instance, it shows that execution times for complex queries are quite reasonable given that we are using a single Neo4j instance. It is evident that the choice of the server, the graph database configuration is important and since the graph is going to get large overtime, it is recommended using the indexing to make the research faster. Through all the experiments we conducted, we realized that without indexing techniques, Neo4j shows good data retrieval performance. In the future, we plan to design and implement benchmarks able to measure the performance of graph database and RDF engines, the impact of implementing different semantic web platforms other than the ones we used in this study into and exploring the limits in adding additional data sources. Additionally, to better understand the pros and cons of existing engines, the analysis of graph structures and core methods is part of our future research work.

REFERENCES

C. J. Anumba, R. R. Issa, J. Pan, I. Mutis Ontology-based information and knowledge management in construction, Constr. Innovation Inf. Process Manage 8 (3) (2008) 218–239 (2008).

K. Balachandar, E. Thirumagal, D. Aishwarya, R. Rajkumar, Ontology mapping techniques and approaches, Int. J. Comput. Appl 65 (24) (2013) 13–20 (2013).

E. Elbeltagi, M. Dawood, Integrated visualized time control system for repetitive construction projects, Autom. Constr 20 (7) (2011) 940–953 (2011).

T. El-Diraby, C. Lima, B. Feis, Domain taxonomy for construction concepts: toward a for- mal ontology for construction knowledge, J. Com- put. Civ. Eng 19 (394) (2005) 394–406 (2005). doi:10.1061/(ASCE)0887-3801.

T. El-Diraby, H. Osman, A domain ontology for construction concepts in urban infrastructure products, Autom. Constr 20 (8) (2011) 1120–1132 (2011).

T. Elghamrawy, F. Boukamp, H. S. Kim, Ontology- based, semi-automatic framework for storing and retrieving on-site construction problem information-An RFID-based case study, ASCE, Reston, VA, 2009 (2009).

N. M. El-Gohary, T. E. El-Diraby, Merging architectural, engineering, and construction ontologies, J. Comput. Civ. Eng (2011) 109–128 (2011). doi:10.1061/(ASCE)CP.1943-5487.0000048.

E. P. Karan, J. Irizarry, Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services, Autom. Constr 53 (2015) 1–12 (2015).

S. Karshenas, M. Niknam, Ontology-based building information modeling, ASCE Int. Workshop on Computing in Civil Engineering (2013) 476–483 Ioannis Brilakis (2013).

S. K. Lee, K. R. Kim, Y. ., J. H. ., BIM and ontology- based approach for building cost estimation, Autom. Constr 41 (2014) 96–105 (2014).

J. Hayes, C. Gutierrez, Bipartite graphs as intermediate model for RDF, Proc., 3rd Int. Semantic Web Conf (2004) 47–61 (2004).