Jean Monnet University

Ecole des Mines de Saint-Etienne

Master of Science -
Cyber-Physical and Social Systems

Master Thesis - M1

LINKED-DATA AUGMENTED IN-BROWSER INTERACTION WITH THE DIGITAL TWIN OF A SMART BUILDING

Student:Arun Raveendran Nair Sheela
Student ID:22008654
Mail ID:ext.21m2022@etu.emse.fr,
raveendran.nair.arun@etu.univ-st-etienne.fr

Supervisor:Maxime LEFRANCOIS
Mail ID:maxime.lefrancois@emse.fr

Saint-Etienne, June 2022

Abstract

Building Information Modeling (BIM) is adopted globally to describe and exchange building data across disciplines during building design and construction. Furthermore, the Internet of Things (IoT) devices are widely used in the construction industry to monitor building environment data for various benefits. In this scenario, numerous studies are going to integrate the BIM model of a building and data collected from the IoT devices installed and thus develop the Digital Twin of the building. However, several challenges still need to be solved for efficient integration and data monitoring through BIM with web technologies. So the objective is to create a more reliable method to integrate BIM with other technologies and construct the digital twin of the intelligent building, thus visualizing in the web browser with performance.

In this work, we addressed some challenges for developing the digital twin of smart building and proposed a research framework for integrating BIM and other technologies like IoT using semantic web technologies and visualizing the BIM in the browser. The proposed framework enables the construction of a digital twin of the building, which is a hot topic in the AECOO (Architecture, Engineering, Construction, Owner, and Operation) industry. The main idea behind the framework is composed of three different phases. The first phase, called BIM as Modular Knowledge Graphs, concentrates on developing a tool to create modular knowledge graphs from the BIM model by extracting the building data. The second phase, New Knowledge Augmentation, integrates more data sources and technologies into the generated modular knowledge graphs. Furthermore, in the third phase, Digital Twin Visualization, we visualized the external data sources integrated with BIM and the 3D model of the building in a browser.

We used the BIM model of the MINES Saint-Etienne Espace Fauriel building developed in Autodesk Revit, and the intelligent building architecture exists in the same building to validate the different phases of the framework.

Acknowledgement

The past five months have been an extraordinary journey. In this moment I would like to acknowledge and give my warmest thanks to my supervisor Maxime LEFRANCOIS who made this work possible. His guidance and advice carried me through all the stages of this project and also his revisions of my scientific and English writings.

Also, I would like to thank the following people who have helped me undertake this research:

To Alaa DAOUD, for his invaluable support throughout the project on many aspects, technical, and administrative.

To Victor CHARPENAY, whose effort to correct one my technical issue in the project.

To Fabien BADEIG, for giving me valuable feedback to improve various aspects of this project.

To Antoine Zimmermann, give useful comments about this work which helps for further improvements.

Also, In this moment I give special thanks to my academic coordinator Pierre Maret for his support throughout the academic year.

To conclude, I cannot forget to thank my family and friends for all the unconditional support in this very intense academic year.

Contents

Acronyms

API Application Programming Interface.

BaaS Building as a Service.

BAS Building Automation System.

BIM Building Information Modeling.

BOT Building Topology Ontology.

BPO Building Product Ontology.

CDC Construction Dataset Context Ontology.

CDO Concrete Damage Ontology.

DOT Damage Topology Ontology.

IFC International Foundation Class.

IoT Internet of Things.

MQTT Message Querying Transfer Telemetry.

OCC Ontology for openCASCADE.

OMG Ontology for Managing Geometry.

OOP Ontology for Object Oriented Programming.

OPS Ontology for Parametric System.

OWL Web Ontology Language.

RDF Resource Description Framework.

REST Representational State Transfer.

SAREF Smart Applications REFerence Ontology.

SEAS Smart Energy Aware Systems Ontology.

SOSA Sensor, Observation, Sample, and Actuator.

SPARQL SPARQL Protocol and RDF Query Language.

SSN Semantic Sensor Netwrok Ontology.

TDY Teddy Ontology.

Introduction

The goal of digitization in the smart building is to obtain a fully semantic and interoperable building called Building as a Service (BaaS) [1] or, more specifically, the Digital Twin of the Building. The central concept of the BaaS model is to develop a collaborative data system by combining the 3D geometry model of the building and the data received from the IoT sensors to improve user interaction and efficient building operation, thus helping to maximize the performance of the technological system, which defines efficient control strategies, energy savings, and cost optimization.

Primarily, The development of the digital twin of the building is composed of two steps. The first step is to build the 3D geometry model of the building, and the second is to deploy IoT devices in the building for capturing real-time instances. The construction industry introduced BIM which is an intelligent 3D virtual model to exchange building information [2]. As a 3D digital representation of buildings, a BIM model contains geometric and semantic information of the building elements [3].Now BIM is a common way to share, exchange and manage information of the whole life cycle of the building. However, BIM provide only static data about the built environment and cannot be directly integrated with any other data sources. The primary problem of using BIM for developing a digital twin of the smart building is the difficulty of integration with other technologies.

On the other hand, developing intelligent buildings includes deploying IoT devices and data analytics for effective decision-making, control, and monitoring. Technologies that help to enable the above tasks are Networking, Edge and Cloud computing, Artificial intelligence, Machine Learning, etc. Integrating the above technologies with the BIM is a challenging task. Furthermore, issues like security, trust, and connectivity exist in addition to the interoperability problem.

In this work, we propose a framework to integrate BIM with other technologies and finally visualize it in the browser. The proposed framework comprises semantic web technologies, the Web of Things, 3D graphics, and modular and distributed principles. Furthermore, this framework effectively integrates the building data with other technologies like IoT with the help of linked data principles.

We take the MINES Saint-Etienne Espace Fauriel building as a reference as we already have BIM model of the building for validating the framework. In the same building, several sensors are installed to monitor the comfort parameters like CO2 level, relative humidity, luminance, etc., and control devices to energize windows, heaters, and other field equipment, which are also defined in the BIM model. The main objective is to build a platform by following the workflow of the proposed framework to visualize the BIM model of the MINES Saint-Etienne Espace Fauriel building in the browser after integrating with IoT and other technologies. Using the platform, the user can monitor the real-time and historical values from installed sensors and give commands to control windows, doors, and heaters. Also, the user can monitor the occupant's details with the platform's assistance.

Objectives

The main objective of this work is to integrate the BIM model of a smart building with other technologies like the Internet of Things, Artificial Intelligence, etc., thus developing the digital twin.

Also, we need to visualize the digital twin in the web browser with good performance.

Thesis outlines

This thesis is organized into seven chapters that are outlined below:

No	Chapter Name	Description
1	Context	Illustrates the background knowledge required to understand the concepts explained in the thesis. Explained about digital twin, semantic web technologies, smart building, modular and distributed architecture, and BIM.
2	Motivation and Research Questions	Describes the Main research question and sub research questions and also, the prerequisites and use cases of this work
3	Methodology	Explains how the research and development of this work was organised
4	Related Work	Describes about existing ontologies, platforms, tools and other research works related to this work
5	Proposed Framework	Illustrates the solution proposed by this work to develop the digital twin of a smart building using a BIM model
6	Implementation and Results	Describes the implementation of the proposed framework and results
7	Discussion	Compares the proposed solution with other related works and explains the advantages and limitation of this work

Table 1. Thesis Outline

Chapter 1

Context

1.1 Definitions

This section explains the definition of some important terms used in the whole work:

1.1.1 Digital Model

A digital model is a computerised, data model of a building, product or some other object that describes the form of an existing or proposed object [4].Another way, A digital model can be defined as a digital copy of a physical object, and there is no information exchange between a digital model and the physical objects. Therefore, once the digital model is created, the change in the physical object will not automatically reflect in the digital model and vice versa [5].

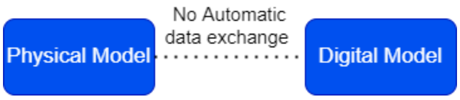

Figure 1.1. Digital Model

1.1.2 Digital Shadow

Based on the definition of a Digital Model, if there further exists an automated one-way data flow between the state of an existing physical object and a digital object is called digital shadow [6]. In digital shadow, there should be an information exchange from a physical object to a digital model. A change in the physical object will reflect in the digital model but not vice versa [5].

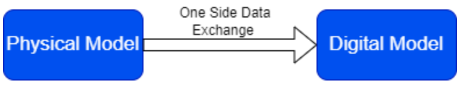

Figure 1.2. Digital Shadow

1.1.3 Digital Twin

Digital representation of a physical facility that receives sensor information from the facility and sends actuation information to the facility [7].In Digital Twin, data flow exists between the physical object and the digital model. Therefore, the change in the physical object automatically leads to a change in the digital model and vice versa [5].

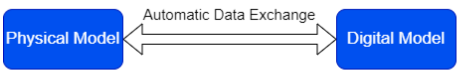

Figure 1.3. Digital Twin

Workflow to Enable Digital Twin -

The first step to build a digital twin is to construct the digital model of the physical object. The next step is to make the digital model to monitor the physical object. Here any change in the physical object should reflect in the digital model but not vice versa, called the digital shadow. Finally, enable the control logic through digital shadow, defined as the digital twin of the physical object.

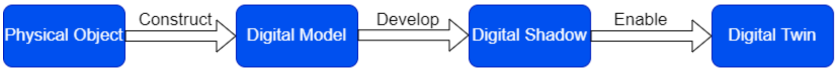

Figure 1.4. work flow to enable Digital Twin

Digital Twin Applications -

The applications of digital twins are widely used across many industries to improve business processes and performance. For example, the building industry, Manufacturing, Supply chain, agriculture etc., are some application areas of the digital twin.

Digital Twin Consortium[1]is an authority that drives the awareness, adoption, interoperability, and development of digital twin technology. Through a collaborative partnership with industry, academia, and government expertise, the Consortium is dedicated to the overall development of digital twins.

1.1.4 Semantic Web

The term “Semantic Web” refers to W3C’s vision of the Web of linked data. Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data[2].The Semantic Web is a concept designed to enable machines to understand the meaning of information on the Web.

1.1.5 Resource Description Framework

RDF is a standard model for data interchange on the Web. RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a “triple”). Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications definition[3]. Three characteristics define an RDF data:

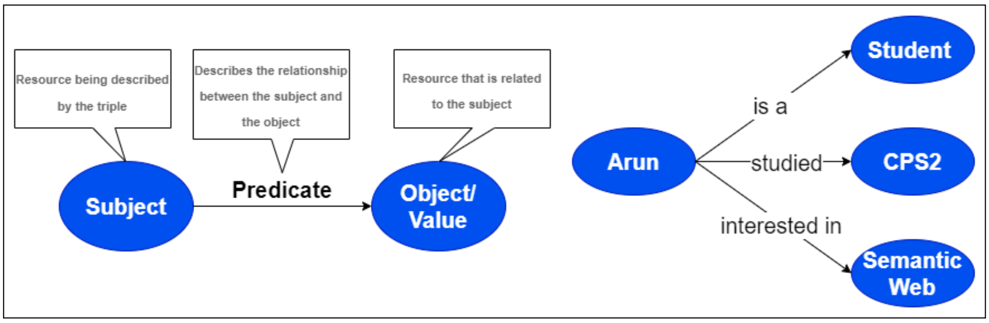

Figure 1.5. Figure to Illustrate the structure of a triple and a simple RDF model

[1]https://www.digitaltwinconsortium.org/

[2]https://www.w3.org/standards/semanticweb/

[3]https://www.w3.org/RDF/

1.1.6 Linked Data Principles

The term Linked Data refers to a set of best practices for publishing structured data on the Web. These principles have been coined by Tim Berners-Lee in the design issue note Linked Data[4]. The principles are:

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information.
4. Include links to other URIs. so that they can discover more things.

1.1.7 Modular Design

Modular design, or modularity in design, is a design principle that subdivides a system into smaller parts called modules which can be independently created, modified, replaced, or exchanged with other modules or between different systems.

1.1.8 Distributed Architecture

A distributed system is a collection of computer programs that utilize computational resources across multiple, separate computation nodes to achieve a common, shared goal. Also known as distributed computing or distributed databases, it relies on separate nodes to communicate and synchronize over a common network.

1.2 Smart Building

Smart buildings are facilities that leverage complex automated systems to maximize operational efficiency and the well-being of occupants. Smart Building integrates IoT devices, Cloud Computing, Artificial Intelligence, and Machine learning for better monitoring user comfort parameters and decision-making. The figure 1.6 represent several component a smart building.

1.2.1 Features

1. Optimized maintenance management system
2. Efficient energy management system
3. Automated control of building operations by considering the user comfort level
4. Improved occupant comfort and well being
5. Increased asset value

1.2.2 Technologies to implement Smart Building

Mainly there are two layers of technologies to implement Smart Building: the Internet of Things layer and the Data Analytics layer. The below table explains the various sub-layers of IoT and Data Analytics layers that make a building smart.

[4] https://www.w3.org/wiki/LinkedData

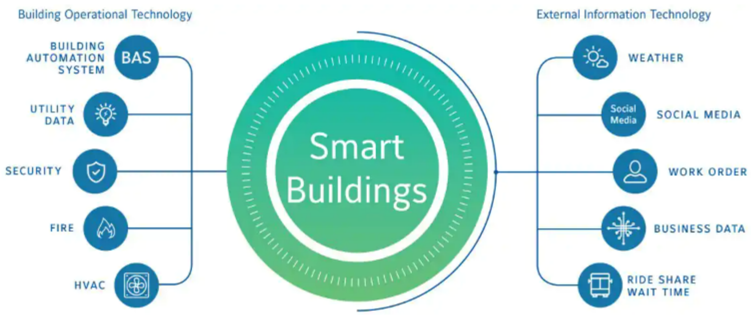

Figure 1.6. Smart Building overview - source [8]

Internet of Things	Data Analytics
Sensors and Actuators	Data Collection and Processing
Embedded controllers	Decision Making
Networking and Communication Protocols	Predictive Analytics Algorithms
Software and APIs	Data Visualization
Data Storage	Statistics and Data Analysis

Table 1.1. Sublayers of IoT and Data Analytics

1.3 Building Information Modeling

BIM is a process of creating and managing information for a built asset throughout its life cycle from planning and design to construction and operations. BIM is a data model that incorporates the 3D geometry and other relevant data of the building and its components. Many software are available to develop the BIM model of a building. Each software produces its own proprietary data format to represent the BIM. Therefore, an open-format named International Foundation Class (IFC)[5] was introduced to solve this software interoperability problem.

Figure 1.7. BIM Tools

1.3.1 Industry Foundation Class

The IFC is an open standard for the exchange of building data models used in building design and construction across different software. IFC defines an EXPRESS based entity-relationship model consisting of several hundred entities organized into an object-based inheritance hierarchy. Examples of entities include building elements such as IfcWall, geometry such as IfcExtrudedAreaSolid, and basic constructs such as IfcCartesianPoint IFC[6].

[5] https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes

[6] https://en.wikipedia.org/wiki/Industry_Foundation_Classes

- IfcObjectDefinition captures tangible object occurrences and types
- IfcRelationship captures relationships among objects
- IfcPropertyDefinition captures dynamically extensible properties about objects.

IFC file formats -

- IFC-SPF is a text format defined by ISO 10303-21 ("STEP-File"), where each line typically consists of a single object record, and having file extension ".ifc". This is the most widely used IFC format, having the advantage of compact size yet readable text.
- IFC-XML is an XML format defined by ISO 10303-28 ("STEP-XML"), having file extension ".ifcXML". This format is suitable for interoperability with XML tools and exchanging partial building models. Due to the large size of typical building models, this format is less common in practice.
- IFC-ZIP is a ZIP compressed format consisting of an embedded IFC-SPF file or IFC-XML file and having file extension ".ifcZIP".

IFC Structure - IFC files create a building model based on a pre-defined structure that builds the model in a logical way. When it is saved, the IFC file format orders the IFC units hierarchically according to their type, as follows.

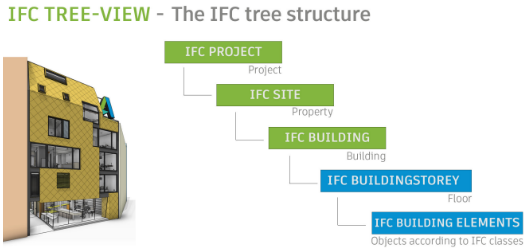

Figure 1.8. Structure of a IFC file - source [9]

IFC Classes and Types -

Physical objects in an IFC file must be categorised to a class of object, which has implications on what properties are expected to be found on the object. For example, a wall has an IFC class of IfcWall.

Three IFC classes and some of its properties are mentioned in the table 1.2:

1.4 Smart Building Digital Twin

A smart building digital twin is a contextual model of an entire smart building environment. It brings together information technology and operation technology systems, that is, IoT sensors and third party data, and contextualizes this with information about processes and people, resulting in a dynamic digital replica that can be used to solve a wide variety of issues. The digital twin of a smart building is not just a digital replica of physical assets, but rather is a complex model of how people and processes interact with their environment. A smart building digital twin essentially acts as a bridge between the digital and physical worlds. It does so by using connected sensors and IoT devices to collect real-time data about physical items. This data is then combined with context and processed and is used to understand, analyze, manipulate and optimize processes within a smart building [10].

IfcWindow	IfcSpace	IfcWall
ThermalTransmittance	Room Name/No	Height
Height	NetFloorArea	Length
Width	Height	Width
Area	GrossPerimeter	GrossFootprintArea
Floor Name	GrossFloorArea	NetVolume
Sill Height	GrossVolume	NetSideArea
Area	Floor No	Floor No
Volume	Area	Unconnected Height
Head Height	Perimeter	Area
Type	Unbounded Height	Length
Family Reference	Volume	Volume
-	Specified Power Load per area	Roughness
-	Total Heat Gain per person	ThermalTransmittance
-	Specified Lighting Load per area	Family Reference
-	Sensible Heat Gain per person	Type
-	Plenum Lighting Contribution	Structural Usage
-	Latent Heat Gain per person	Enable Analytical Model

Table 1.2. Table to compare the different property set of three IFC classes

Key enabling factors to enable smart building digital twin are explained below:

1. Data. Data from across the entire smart building is needed, such as data about people, processes, connected devices, operational building systems, IT and external information like weather or transit feeds.

2. Real-time Information which give the actual state of the Building

3. Reasoning logic to make effective decision making for better operation of the building and control system to execute the decision.

Method to enable smart building digital twin - One method of enabling the smart building digital twin is to integrate the BIM model and the smart building .

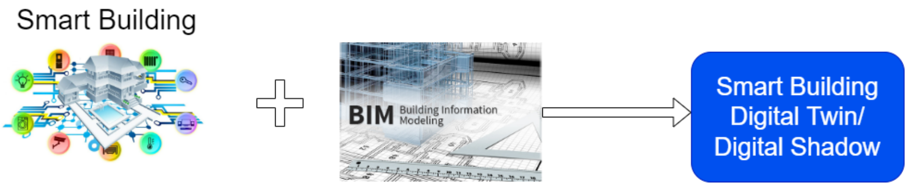

Figure 1.9. Smart Building Digital Twin

Motivation and Research Questions

2.1 Motivation

Building Information Modeling, Internet of Things, and Data Analytics and their convergence are critical factors for enabling the digital twin of the smart building. However, there are several challenges to efficiently integrating these technologies.

- **Challenge 1 -Interoperability of BIM model with other information technologies** [11].

  Currently, using a specific technology will not satisfy the current requirement of the smart building. So we need to integrate more and more technologies with the BIM for different intelligent applications. But currently, there is no standardised approach for integrating BIM with other advanced information technologies in various disciplines. Solving this interoperability issue of BIM with other technologies requires further research exploration and studies.

- **Challenge 2 - Data Quality** [5]

  Data required for a digital twin should be noise-free with a constant and uninterrupted data stream. If the data is poor and inconsistent, it will affect the performance of the digital twin. Also, the heterogeneity of data from different sources is another problem related to this challenge.

- **Challenge 3 - Visualization of real-time and historical data from IoT devices using large BIM models in the web**

  Large BIM models are challenging to visualise in a browser with good performance. Also, monitoring the real-time and historical value of comfort parameters with BIM's assistance in a browser is challenging.

Our motivation is to solve the above said challenges and propose a framework to develop the digital twin of a smart building from a BIM model.

2.2 Research Questions

We define one main research question and three sub-research questions, which helps to solve the current challenges in developing the digital twin of a Smart Building from the BIM model.

2.2.1 Main Research Question

- **MRQ1 - How to integrate the BIM model with other technologies like IoT, Artificial intelligence, etc., for the development of the digital twin of the smart building and visualisation in the browser?**

  This research question defines the problem statement of the whole work. There are many sub-problems related to this. So we split the main research question into three sub-research questions to solve the different challenges explained in the motivation section.

2.2.2 Sub Research Questions

- **SRQ1 - How to generate modular knowledge graphs from a BIM Model of Smart Building?**

Conversion of BIM to Knowledge graphs by following the modular design pattern is the objective of this research question. This approach helps to integrate other knowledge easily and also a faster process. Sustainability is also one of the advantages of this methodology. This research question was defined to solve the first challenge.

- **SRQ2 - How to integrate different distributed knowledge with the modular Knowledge graphs generated from the BIM model of the Smart Building ?**

  This research question addresses the methodology to integrate multiple data sources and technologies distributed in a building network with the modular knowledge graphs generated from the BIM model. This research question focuses on solving the first and second challenges.

- **SRQ3 - How to develop the digital twin and 3D visualise in the browser using the modular knowledge graphs ?** This sub-research question addresses the third challenge to develop a platform for visualising the digital twin in a browser with good performance. Also, with the platform, anyone can monitor IoT data and give commands to control the building elements.

2.3 Prerequisites

Prerequisites required to complete the research work are explained below.

2.3.1 BIM model

The first requirement is to select a BIM model of a building. In this research, We selected BIM models of the Espace Fauriel building of Ecole des Mines to validate the framework.

The Espace Fauriel (EF) building at Mines Saint- Étienne, built around 1920 by Manufrance and re- empowered in 1994, is a building of 6720 m2 on eight floors used for research and teaching at Mines Saint Etienne. It includes lecture halls and auditoriums, courtyards, offices, meeting rooms, and a simulation factory of the future called the ITM factory.

The BIM model of the Espace Fauriel building is developed using the Autodesk Revit software.

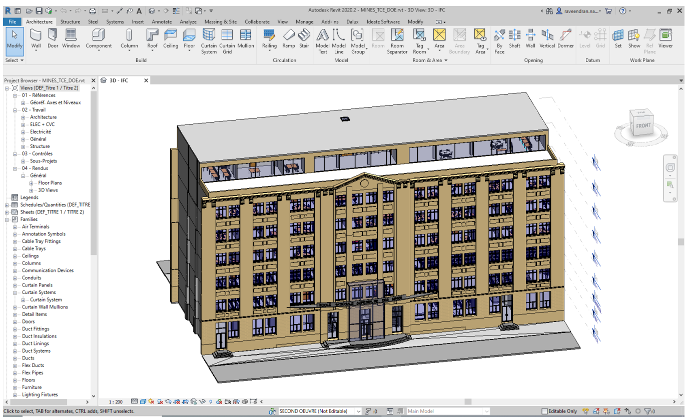

Figure 2.1. BIM model of the Espace Fauriel building created using Autodesk Revit

2.3.2 Smart Building Architecture

Several sensors are installed in the Espace Fauriel building to monitor comfort parameters and the window and door status. Humidity, Temperature, Luminosity, and CO2 level are comfort parameters monitored in the building. Also, monitor the Active and Reactive power with the help of an energy meter.

In each room, sensors are installed to monitor the above parameters connected to an ESP32 module. Finally, through wifi, it broadcasts the data to an MQTT broker. Here Mosquito is used as the MQTT broker. With the help of an MQTT broker, the data is published over a web socket for real-time monitoring and logged to OPENHab, a platform for Smart Home for historical monitoring. Similarly, the energy meter is connected to the MQTT broker with the help of a middleware to access the active and reactive power.

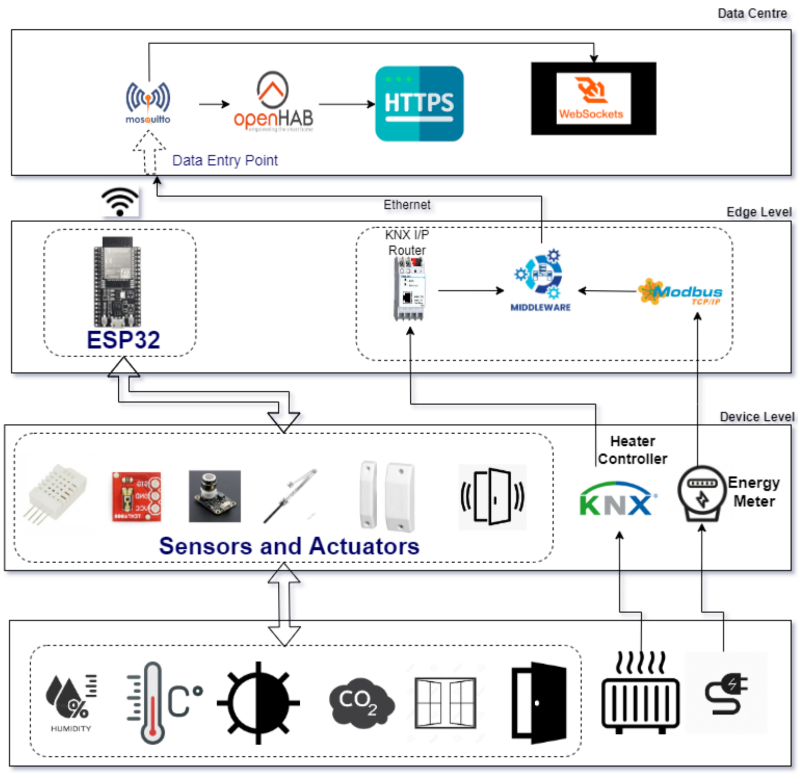

Figure 2.2. Smart Building Architecture in Espace Fauriel Building

2.4 Use Cases

In the research we consider three use cases for the implementation of the proposed framework with reference to the Ecole Des Mines Espace Fauriel Building.

1. **Visualization of comfort parameters using BIM -** Integrate the comfort parameters like temperature, CO2 level, relative humidity etc. and visualize the real-time and historical data using BIM.

2. **Monitor real-time window and door status using the BIM**

3. **Monitor occupants status -** Visualize important information of the occupants in the building with the assistance of BIM.

Chapter 3

Methodology

3.1 Research Methodology

We try to identify the challenges from the initial objective and define the main research question. Regarding challenges, we divide the main research question into three sub-research questions.

Then, We defined three different phases, named them to answer the three sub-research questions, and selected use cases for the implementation.

The phases named as:

1. BIM to Modular knowledge graph
2. New Knowledge Augmentation
3. Visualization of Digital Twin in browser

This work's first phase mainly focuses on the first question, converting the BIM model to modular knowledge graphs. The first step is to identify the existing ontologies to define the BIM models, search for the existing tools available to convert the IFC into RDF graphs, and identify the limitations. The second one is to find the recent papers published regarding the intersection of BIM and Semantic Web technologies, mainly from 2015 to 2022. We used mainly Semantic scholar, Google scholar and the IEEE website to find the paper.

Main Objectives of Phase 1 -

1. Find the existing Ontologies to represent BIM data
2. Find the existing tools to convert BIM to RDF
3. Find the ongoing research regarding the nexus of BIM and Semantic Web Technologies
4. Develop a tool to convert the BIM model to distributed knowledge graphs by referring the data get from the above steps .

Also, We used the below query to search in the Semantic Scholar to find the research papers related to the semantic web and BIM.

```
"Semantic Web" AND "BIM" OR "Building Infromation Modeling" AND "IFC" OR "Industry Foundation Class" AND "RDF" OR "
    Resource Description Framework" .
```

Listing 3.1. Query used in Semantic Scholar

In the second phase, we tried to find the available research frameworks to integrate Knowledge graphs and Digital Twin enabling factors like IoT, Artificial Intelligence etc.

Main Objectives of Phase 2 -

1. Integrate new knowledge to the distributed graphs generated from the BIM to enable Digital Twin
2. Identify some applications that prove the importance of the distributed graphs

Also, I used the below query to search in the Semantic Scholar to find the research papers related to the semantic web and BIM .

```
"Semantic␣Web" AND "BIM" OR "Building␣Infromation␣Modeling" AND "IFC" OR "Industry␣Foundation␣Class" AND "RDF" OR
"Resource␣Description␣Framework" AND "Digital␣Twin" AND "IoT" OR Internet of Things"␣AND␣"Smart Building"␣OR
"Smart Home".
```

Listing 3.2. Query used in Semantic Scholar

Finally, phase 3 mainly covers the visualisation in a browser by combining the BIM model and the modular Knowledge graphs generated from the previous phases. Here we tried the find the existing method available to integrate the BIM model with other data sources and technologies for the development of the digital twin of the smart building.

3.2 Development Methodology

We must develop specific tools and visualization platforms to evaluate the framework for answering the research questions.

First, we must develop a tool to convert the IFC model into distributed Knowledge graphs. For this, We first need to find a library to extract the building information from the IFC file and create knowledge graphs. Then, select a web framework to create the rest APIs for the distributed graphs. The second step is to create a tool to augment new knowledge of the distributed graphs generated. Here we need to select a front-end framework for developing the tool. The third step is to find a library to integrate the IFC model in a browser using a front-end framework. Here we need to create many widgets for visualizing various data.

We used SCRUM methodology for the development of the tool and used GitLab as a tool for applying it. We create the task as issues and try to finish it within the predefined deadlines. Each week there is a meeting with the supervisor to monitor the progress of the current work and discuss future work.

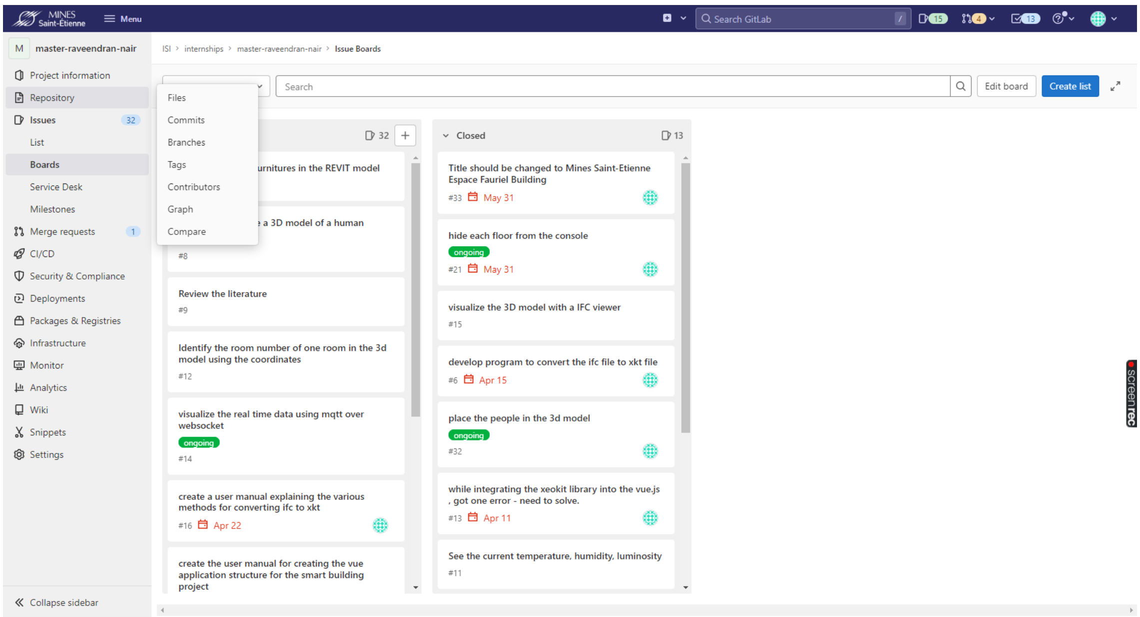

Figure 3.1. Screenshot of gitlab directory used in this work

Chapter 4

Related Works

There are several BIM softwares in the market and each have there own custom data format as output. To solve this software interoperability issue an open format called Industry foundation class was introduced for sharing the BIM data globally which is written in Express schema , a data modeling language [12] .Most of the BIM softwares has there own custom plugin to convert their output to IFC format hence solve the software interoperability problem exist in BIM to a certain limit [13].

4.1 Nexus of BIM and Smart Building

In 2021, Ang Yang and team review various applications and challenges for adopting BIM for the developement of smart building digital twin.

4.1.1 Applications

The diagram below explains various applications and benefits of the BIM integrated Smart Building.As per [11], BIM is adopted in planning, design and construction phase of the smart building and now BIM helps in the operation and the maintenance phase too.Also, there is a possibility to integrate BIM with other technologies which make BIM is an effective approach to make the digital twin of the smart buidling.

Advantages of BIM integrated Smart Building

1. Increasing economic benefits [11]
2. Improved user experience in monitoring parameters
3. Improving the life quality of users [11]

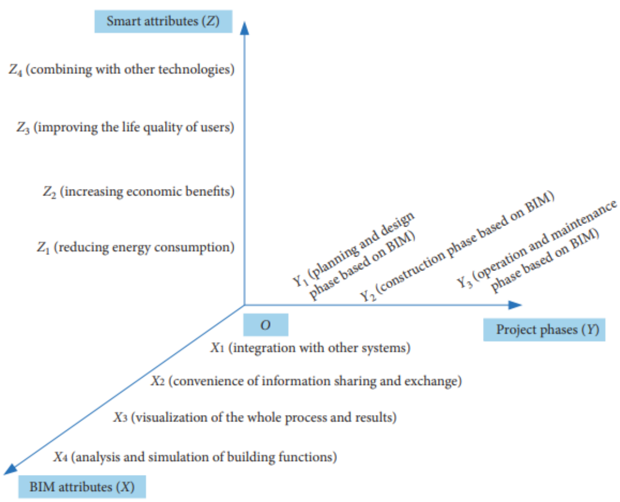

Figure 4.1. nexus of BIM and Smart Building [11]

4.1.2 Research Gaps of Integrating BIM and Smart Building

[11] identifies some significant research gaps in BIM integrated smart building. They are clearly explained in the diagram below and also the diagram explains about the future research trends.

No	Research gaps	Future research trends
1	Limited BIM application in operation and maintenance	Integration of BIM with other information technologies
2	Unclear economic benefits	Calculate the economic benefits and comprehensive benefit analysis systematically
3	Difficult to integrate BIM effectively with other technologies	Establish an open platform for the integration of BIM with multiple technologies

Table 4.1. Research Gaps and future trends in the nexus of BIM and Smart Building [11]

4.2 Phase 1 - BIM to Modular Knowledge Graphs

4.2.1 Existing Ontologies to define BIM

IfcOWL

In 2016 Pieter Pauwels and Walter Terkaj introduced a meth od to convert the IFC model to RDF graphs. In this work the direct mapping of IFC express schema into Web Ontology Language (OWL) ontologies and as a result produces the IfcOwl ontology.ifcOWL provides a OWL representation of the IFC schema. IFC is the open standard for representing building and construction data (see BuildingSMART). The ifcOWL ontology has the same status as the EXPRESS and XSD schemas of IFC. Complex structure and size of the generated graphs are main drawbacks of this work [14].

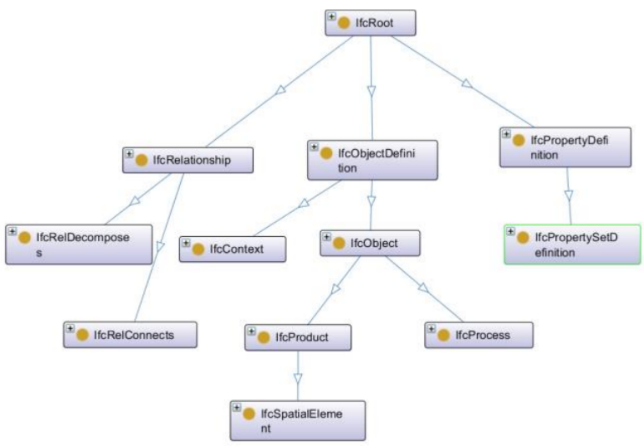

Figure 4.2. A part of the hierarchy of classes in ifcOWL ontology [11]

Building Topology Ontology (BOT)

BOT is an ontology focus on capturing hierarchial concepts in building such as sites, floor, zones and rooms [15].BOT mainly represent the spatial information and formal definitions of the BIM.[1]

[1] https://w3c-lbd-cg.github.io/bot/

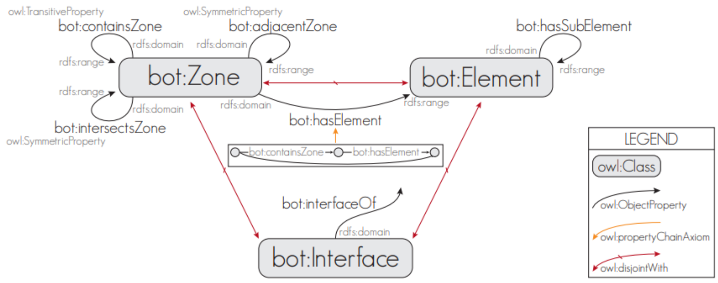

Figure 4.3. Illustration of main three classes of BOT ontology [15]

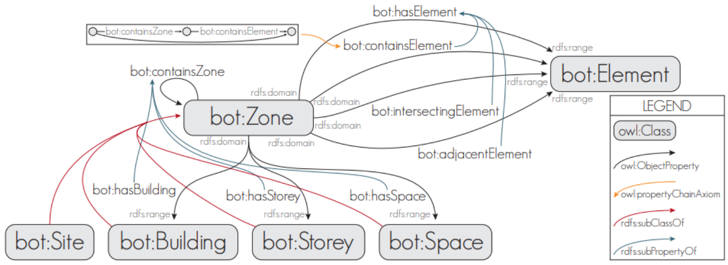

Figure 4.4. Illustration of the four sub-classes of bot:Zone and the three sub-properties of bot:hasElement [15]

Brick

Brick was introduced to represent the metadata of the building which help to define the entities and their relationships necessary for the effective representation of the building and their subsystem [16].Bricks also helps to represent the HVAC , electrical system , lighting system and also the sensor network of the building.[2] Illustration of Brick schema is shown in the figure 5.6 and 5.7.

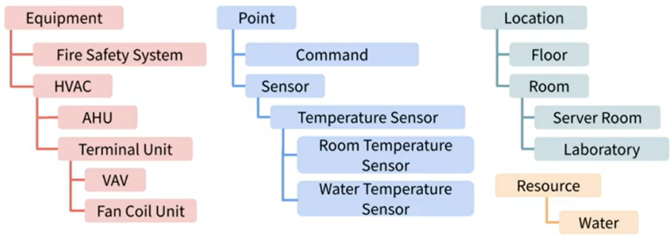

Figure 4.5. Brick class hierarchy [15]

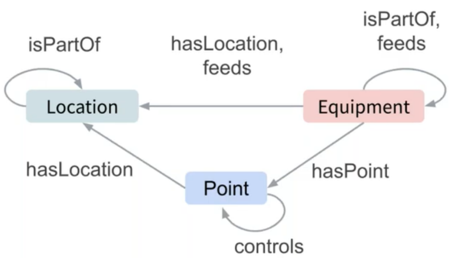

Figure 4.6. Brick relationships [15]

[2] https://brickschema.org/ontology

RealEstateCore Ontology

RealEstateCore[3] is a modular ontology defined to represent spaces, building elements, assets and logical devices in a building [17]. The figure 4.7 illustrated the structure of RealEstateCore ontology.

- A core:Space is contiguous part of the physical world that has a 3D spatial extent and that contains or can contain sub-spaces.
- A core:BuildingComponent defines all element like window, walls, doors etc. in the building.
- A core:Asset is an object which is placed inside of a building, but is not an integral part of that building's structure.
- A core:LogicalDevice: A physical or logical object defined as an electronic equipment or software that communicates and interacts with a digital twin platform.

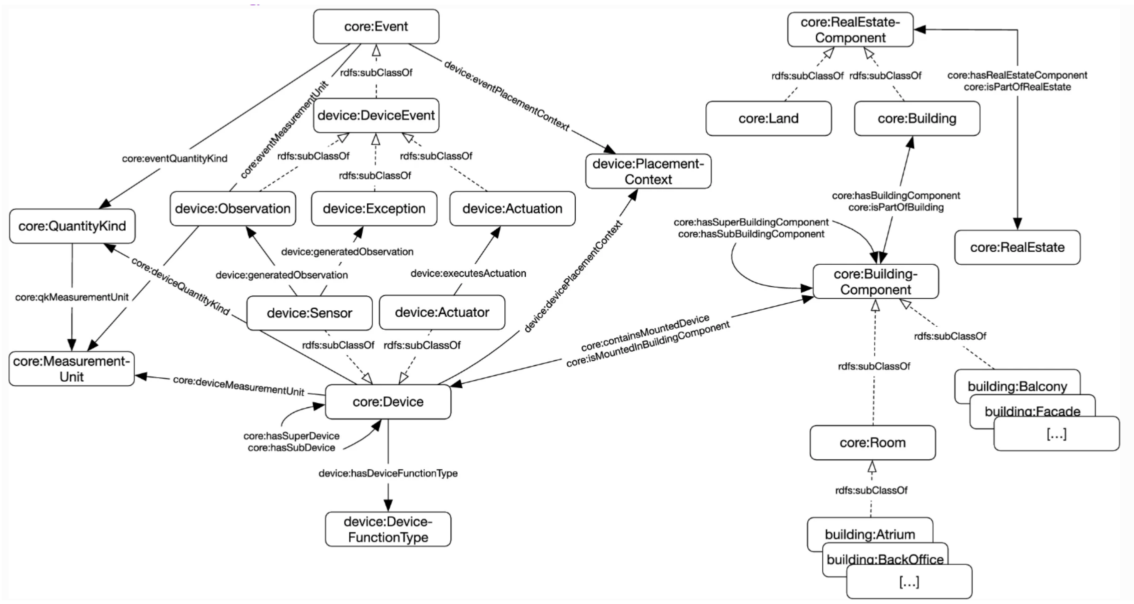

Figure 4.7. RealEstateCore Ontology [17]

4.2.2 Current research work in the nexus of BIM and Semantic Web Technologies

Several studies were going on to convert BIM to RDF graphs in which the building data is both human-readable and usable by machines [14]. As a result, several ontologies, tools, and platforms emerged for this purpose.The BIMserver [4] which is an IFC model server which helps to store and manage building information [18]. The main advantage of this platform is to query(Java query code) and get information via BIM regarding IFCGuid or IFC Class. This method does not use semantic web concepts, but the architecture of this approach was a good reference for the development of other platforms.

A platform named DRUMBEAT[5] was introduced for publishing BIM-related data on the web(Web of Building Data) [19]. The platform provides building data through Representational State Transfer (REST) Application Programming Interface (API) which was spread over multiple DRUMBEAT servers. IFCOWL is the primary semantic model of the DRUMBEAT platform. Using this platform,

[3] https://www.realestatecore.io/

[4] https://github.com/opensourceBIM/BIMserver

[5] http://www.drumbeat.fi/

we can upload various BIM models and get building data of each regarding IFCGuid using the REST interface.

Another initiative called Project SCOPE[6] proposes a micro-service-based architecture to interact with full RDF-based BIM stored in a triple store [20]. Ontologies developed and used in this research project focus on linking geometric and non-geometric contents, geometry data, and building product descriptions.

Ontologies used in project SCOPE

1. Ontology for Managing Geometry (OMG)[7]
2. Building Product Ontology (BPO) [8]
3. Ontology for Object Oriented Programming (OOP) [9]
4. Ontology for openCASCADE (OCC) [10]
5. Ontology for Parametric System (OPS) [11]
6. Teddy Ontology (TDY) [12]

In 2021, Jeroen Werbrouck, Pieter Pauwels, Jakob Beetz and Erik Mannens proposed a web based framework for the management of heterogeneous and federated building data. They developed a custom namespace called LBDserver in addition to BOT, Construction Dataset Context Ontology (CDC),Damage Topology Ontology (DOT) for the development of this framework [21].The BIM model is converted into IFC, then to RDF using the ontologies described above, and to a geometric model for visualization(glTF). Sharing of damage cases is one of the use cases implemented in this approach with the help of DOT and Concrete Damage Ontology (CDO).

4.2.3 Existing Tools to convert BIM to RDF

There are currently two tools available for the convertion of BIM to RDF:

IFC - to - RDF Converter

This tool[13] convert IFC to a big complex RDF by implementing IfcOWL standard [14]. The tool is developed in Java and have a command-line interface and a graphical user interface for the conversion process. The listing 4.1 represents the name-spaces used in the tool.

```
@prefix ifcowl:  <http://standards.buildingsmart.org/IFC/DEV/IFC2x3/TC1/OWL#> .
@prefix inst:  <http://linkedbuildingdata.net/ifc/resources/20220720_142331/> .
@prefix list:  <https://w3id.org/list#> .
@prefix express:  <https://w3id.org/express#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix owl:  <http://www.w3.org/2002/07/owl#> .
```

Listing 4.1. Namespaces used in IFCtoRDF Converter

[6] https://www.projekt-scope.de/ontologies/

[7] https://www.projekt-scope.de/ontologies/omg/

[8] https://www.projekt-scope.de/ontologies/bpo/

[9] https://www.projekt-scope.de/ontologies/oop/

[10] https://www.projekt-scope.de/ontologies/occ/

[11] https://www.projekt-scope.de/ontologies/ops/index.html

[12] https://www.projekt-scope.de/ontologies/tdy/

[13] https://github.com/jyrkioraskari/IFCtoRDF-Desktop

IFC to Linked Building data

This tool[14] extract relevant information from IFC building models and convert it into one big RDF graph. The tool mainly consider BIM properties and relationships of individual IFC elements for the conversion process [22].

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix bot: <https://w3id.org/bot#> .
@prefix ifc: <https://standards.buildingsmart.org/IFC/DEV/IFC2x3/TC1/OWL#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix beo: <https://pi.pauwel.be/voc/buildingelement#> .
@prefix lbd: <https://linkebuildingdata.org/LBD#> .
@prefix props: <http://lbd.arch.rwth-aachen.de/props#> .
@prefix geo: <http://www.opengis.net/ont/geosparql#> .
@prefix unit: <http://qudt.org/vocab/unit/> .
@prefix IFC4-PSD: <https://www.linkedbuildingdata.net/IFC4-PSD#> .
@prefix smls: <https://w3id.org/def/smls-owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix inst: <https://www.ugent.be/myAwesomeFirstBIMProject#> .
@prefix mep: <http://pi.pauwel.be/voc/distributionelement#> .
@prefix furn: <http://pi.pauwel.be/voc/furniture#> .
```

Listing 4.2. Namespaces used in IFCtoLBD Converter

4.3 Phase 2 - New Knowledge Augmentation

4.3.1 Web of Things Architecture

W3C's Web of Things[15] architecture was introduced to solve the interoperability problem in different IoT devices.This architecture is built based on the existing web standards for describing the IoT devices [23].

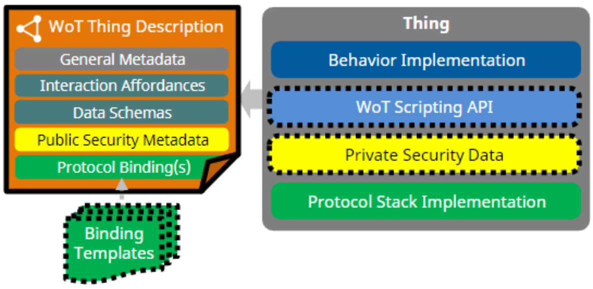

Figure 4.8. Web of Things Architecture [23]

WoT Building blocks

1. **Thing Description** - Thing description is the main building block of W3C's Web of Things architecture introduced to solve the interoperability problem in different IoT devices.As per thing description, every IoT device has four main components:textual metadata of the device itself,how to interact with the device,data format used and the relation with other devices.

```
@prefix td: <https://www.w3.org/2019/wot/td#> .
```

Listing 4.3. Namespaces used in Thing Description

[14] https://github.com/jyrkioraskari/IFCtoLBD

[15] https://www.w3.org/TR/wot-architecture/

2. **Binding Templates** - This block mainly focus on the application layer of an IoT network.IoT devices has a variety of protocols to access data from them.Binding Templates helps to represent the particular protocols of a IoT device in the thing description.

3. **Scripting API** - This is a optional building block of WoT architecture which enable the implementation of application logic of a IoT device/thing.

4. **Security and Privacy guidelines** - Security and privacy are a cross-cutting issues that need to be considered in all WoT building blocks and WoT implementations.

4.3.2 Exisiting Research Works for integrating BIM and IoT data

Shu Tang and team review several methods of integrating building contextual data and time-series data generated in the IoT layer [24]. One approach is to represent the building contextual data and sensor information into RDF format and, similarly, serialize the time-series data from IoT device to RDF. Then link both RDFs using a unique identification. The architecture of this method is shown in figure 4.9. Regarding [24] Shu Tang, Cheng Zhang, Jianli Hao, and Fangyu Guo developed

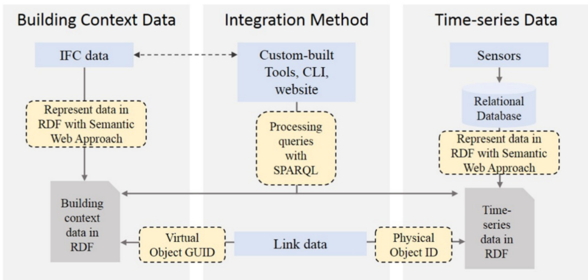

Figure 4.9. Integration of time-series data and building contextual data [24]

a framework to facilitate information exchange between BIM-based building contextual data,time-series data generated from the IoT devices stored in a relational database, and Building Automation System (BAS) metadata using Semantic Web Technology [25]. Converting the time series to RDF representation may cause massive data conversion and high computational requirements. So this method uses online URLs to access IoT data directly from relational databases.

In 2020, Markus Kuller and their team proposed another framework to implement a novel digital twin for smart homes and building based on WoT thing description standardized by W3C [26]. This framework contains three components a device data to WoT converter, a triplestore to store the generated knowledge graphs and a web service to publish via REST API for the digital twin implementation. Architecture of the proposed framework is shown in the figure 4.10: Also In 2022, Soroush

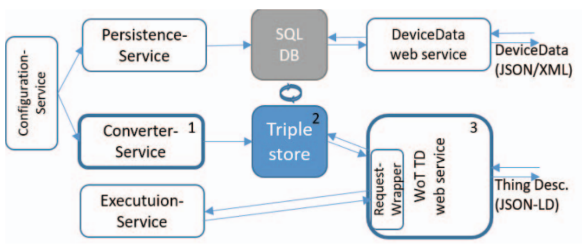

Figure 4.10. Workflow of digital twin using WoT thing description proposed by [26]

Sobhkhiz and Tamer El-Diraby proposed an complete architecture of smart building digital twin with prime focus on semantic web technologies and linked data as the solution to data integration [27].The figure show the full architecture propose by this research work:

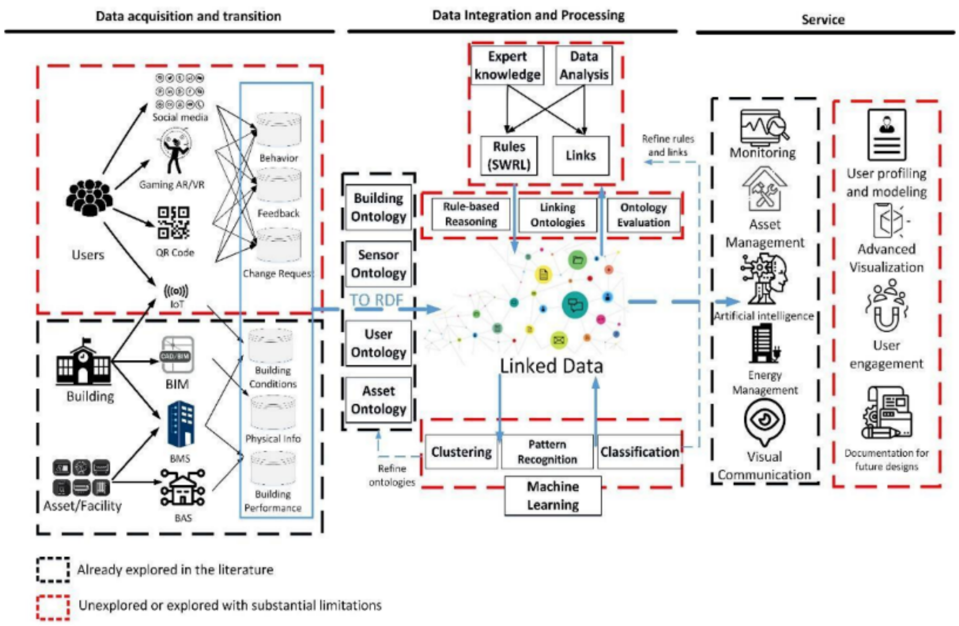

Figure 4.11. Complete Digital Twin architecture by [27]

4.3.3 Common Ontologies used in Smart Building

Smart Applications REFerence Ontology (SAREF)

SAREF ontology are developed to define semantic interoperability among the appliances like heater, washing machine etc in the smart building or home.Overview of the SAREF ontology by showing main classes and their propeties are shown in the below diagram.

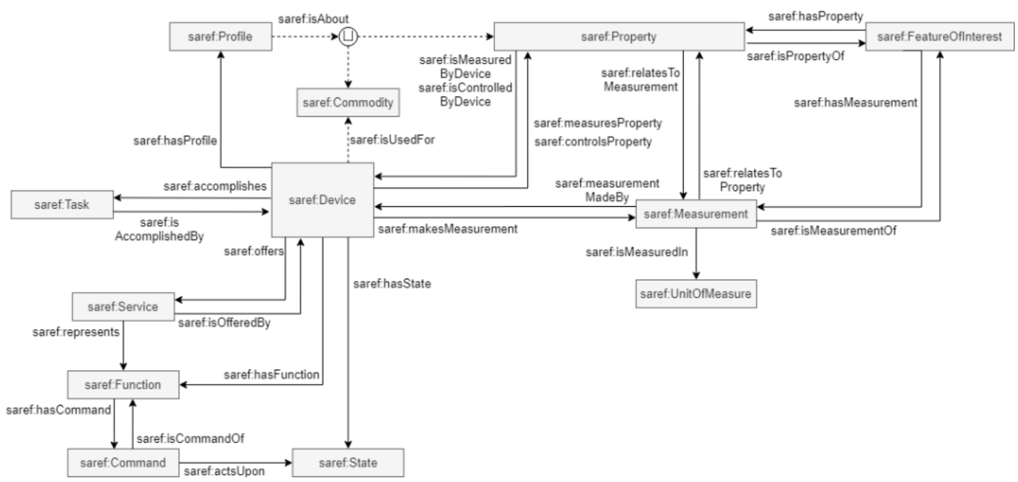

Figure 4.12. overview of the main classes of SAREF and their relationships [11]

Namespace and suggested prefix of Saref Ontology:

```
@prefix saref: <https://saref.etsi.org/core/> .
```

Listing 4.4. Namespaces used for SAREF ontology

The ontologies like Semantic Sensor Netwrok Ontology (SSN)/Sensor, Observation, Sample, and Actuator (SOSA),Smart Energy Aware Systems Ontology (SEAS), SBonto(Ontology of smart building), CTRLont Ontology etc. are some existing ontologies using in the smart building.

4.4 Visualization of BIM

Many researchers are trying to define an efficient methodology to integrate BIM and Smart Building, Thus enabling the Digital Twin. Worawan Natephra and Ali Motamedi propose a framework to directly incorporate the IoT sensor data in the BIM model and visualize it using a game engine called Unreal engine [28]. The architecture proposed by this research work is explained in figure 4.10.

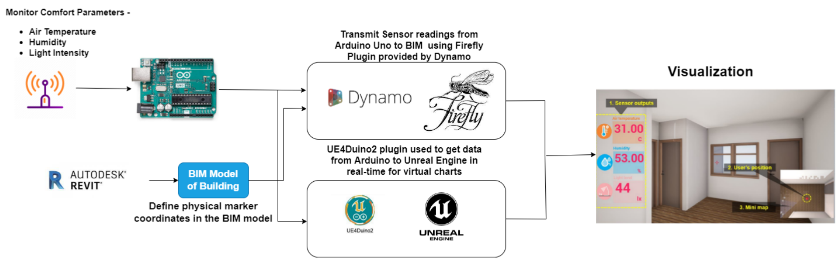

Figure 4.13. Architecture of method proposed by Worawan Natephra and Ali Motamedi

Similarly, Yingying Zhang and Jakob Beetz propose another method that is a real-time, open-source, low-cost solution for monitoring the building environment based on BIM and CPS systems. The solution uses an economical and lightweight Raspberry Pi and sensors to obtain the physical environment data, then uses the Message Querying Transfer Telemetry (MQTT) communication protocol to transfer the data to a time-series database for storage. A web-based BIM viewer is used as a visual dashboard to interact with the 3D building model to subscribe to the sensor data through the MQTT protocol via WebSockets [29].

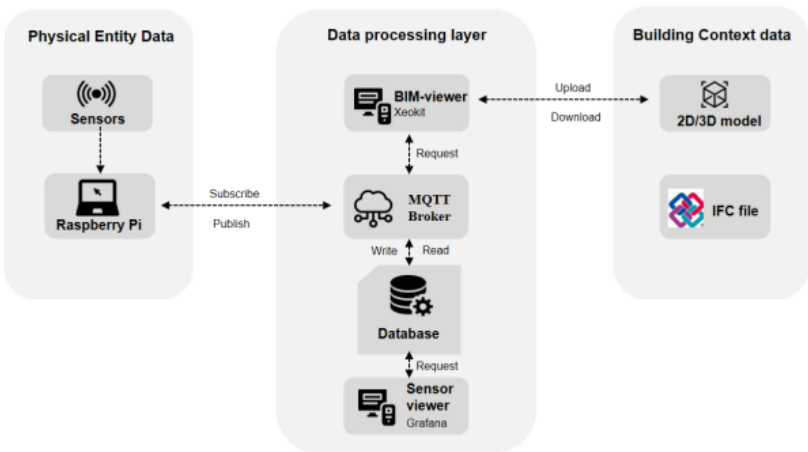

Figure 4.14. Architecture of method proposed by Yingying Zhang and Jakob Beetz [29]

Matti Pouke and team compare two different workflows to develop web-based digital twin visualization of smart home [30]. Also, both methods visualized building sensor data with the assistance of the BIM model in the browser.

Proposed Framework

This chapter describes the research framework proposed by this work to develop the digital twin of a smart building from the BIM model.

5.1 Overview of the Framework

The proposed architecture builds the Digital Twin of a Smart Building by integrating the BIM models and other technologies like IoT using Semantic Web Technologies, which will run on a browser.

The realization of the proposed framework consists of three main phases.

1. **BIM to Modular Knowledge Graphs -** Convert BIM models into modular knowledge graphs, and all are published via REST API.
2. **New Knowledge Augmentation -** Integrate new knowledge like IOT to the modular knowledge graphs generated from the BIM model using Linked data principles.
3. **Visualisation of Digital Twin -** Visualise BIM in the browser using the modular knowledge graphs and thus produce the digital twin of the smart building.

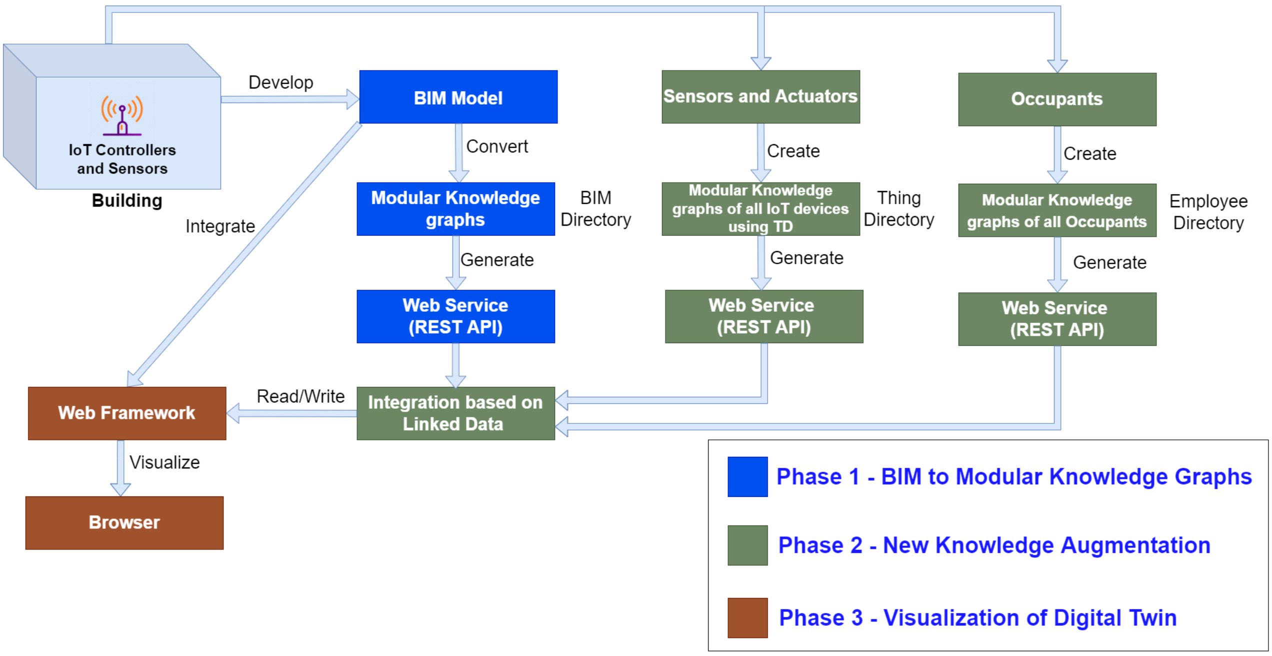

Figure 5.1. Overview of the Proposed Framework

5.2 Phase 1 - BIM to Modular Knowledge Graphs

The main objective of this phase is to convert the created BIM modules to modular knowledge graphs and publish them using the REST API. As the primary task, we need to develop a tool for achieving the above objective. Primarily, a BIM model of the smart building should be created using BIM authoring software and exported the model to IFC format.

IFC to Modular RDF

Now need to develop a tool to extract the above parameters of every element from the IFC and convert them to RDF graphs. Each IFC element has one knowledge graph containing its property definition, geometry data, and object relationships. All these knowledge graphs are published via REST API.

There are three steps to developing the tool which converts BIM to modular knowledge graphs.

1. **Parameter extraction from the IFC file** - Extract IFC parameters of every Ifc element from the BIM model using any IFC library available in the market.
   (a) Property definition - parameters assigned to a IFC element which belong to particular IFC class. Each IFC class has different property set based on their feature.
   (b) Geometry data
   (c) Object relationships - relation of one Ifc element with other elements
2. **RDF graphs creation** - Convert the IFC parameters to RDF graphs of each IFC element by selecting the appropriate vocabulary to represent properties, geometry data and object relationships. We used BOT ontology to describe the object relationships and develop a custom ontology called propdef for representing property definition and geometry data.

```
#property defintion ontology to represent IFC element properties
@prefix propdef: <http://127.0.0.1:5000/test> .
@prefix ifcowl: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL> .
#Building topology ontology to represent object relationship
@prefix bot: <https://w3id.org/bot#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

Listing 5.1. Namespaces used for RDF creation

Why we need a new ontology? There is already an ontology called props ontology [1] for describing IFC property sets. However, props ontology does not contain all the vocabulary we need to model the RDF graph of IFC elements. For example, the properties like sill height(IFCWindow), head height(IFCWIndow), NetVolumn(IFCWall) etc., are missing in the props ontology. So we need a new ontology to define missing properties and the existing vocabulary for maintaining uniformity.

Propdef Ontology It is a light weight custom ontology developed to define the IFC property sets of different IFC classes like IFCWindow, IFCWall, IFCDoor etc. The property sets defined for each IFC class are different(refer to table 2.2 for the property sets of IFCWindow, IFCWall and IFCSpace). A part of propdef ontology for IFCWindow is shown in the figure 5.2.

BIM directory The place where all modular knowledge graphs of the BIM model are generated and maintained as turtle files by the BIM to RDF tool. The root location of the directory should be given to the tool at the time of conversion. The tool extracts IFC classes used in the BIM model and creates folders in the directory. Finally, the IFC elements belonging to every IFC class should be stored in their corresponding folders as turtle files. Refer to figure 6.3 to know the structure of the BIM directory.

[1] http://lbd.arch.rwth-aachen.de/LBD/index-en.html

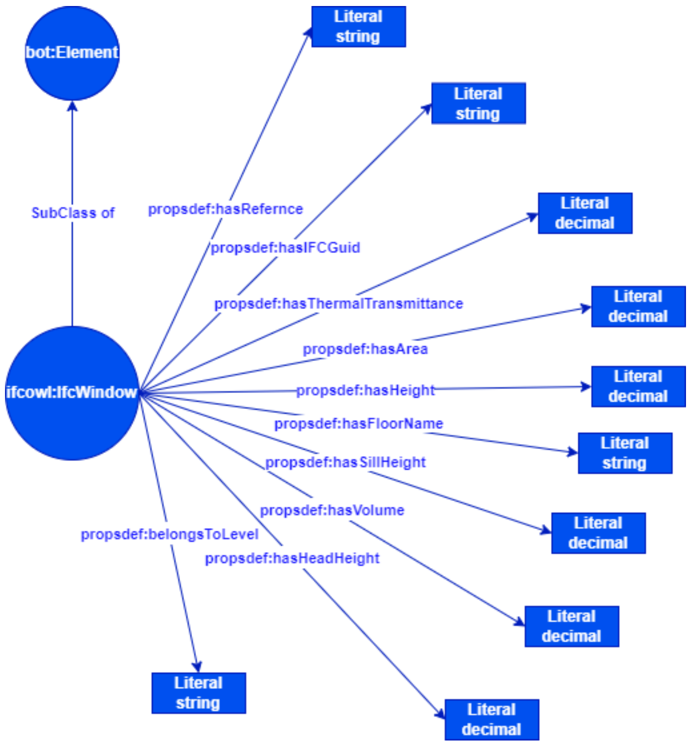

Figure 5.2. A part of Props Ontology - IfcWindow

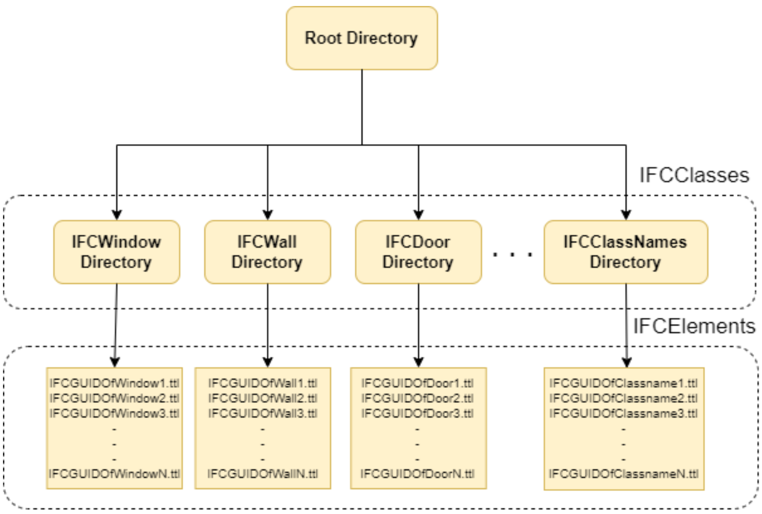

Figure 5.3. Structure of BIM directory

3. **Publish the knowledge graphs via REST API** - Publish the generated graphs by creating REST APIs using any web framework regarding the IFC class and IFCguid of the element from the BIM directory.

   Syntax of hyperlink to access all knowledge graphs are shown in the listing 5.2:

```
http://serverIP:port/:IFCClassName/:IFCGuid
```

Listing 5.2. Syntax of the hperlink to the access the knowledge graphs of IFC elements

where IFCClassName should IFCWindow, IFCWall etc. and IFCGuid is the unique id of every elements in the building.

Algorithm to develop the BIM to Modular Knowledge Graph tool

The below algorithm explains the algorithm of tool to convert BIM to Modular Knowledge graphs:

Algorithm 1 Tool to convert BIM to Modular knowledge graphs

Require: *fileLocationIFC, BIMDirectoryLocation, ifcObjects = {}*
  ifcFile = read(fileLocationIFC) ▷ read ifc file from the given location
  ifcClasses = extractIFCClasses(ifcFile)
  while $i \leq length(ifcClasses)$ **do**
    ifcElements = {}
    ifcElements = ifcFile.type(ifcClasses[i]) ▷ get list of the objects in the ifc class
    while $j \leq length(ifcObjects)$ **do**
      ifcElementPropertySet = ifcElement.getPropertySets
      ifcGuid = ifcElement.getIFCguid
      ifcClass = ifcElement.getIFCClassName
      convertedKnowldegeGraph = rdfGraphConversion(ifcElementPropertySet,ifcGuid,ifcClass, BIMDirectoryLocation)
      j++
    end while
    i++
    end while

5.3 Phase 2 - New Knowledge Augmentation

After the first phase, we have knowledge graphs of all IFC elements in the building, which includes windows, doors, heaters, sensor boxes etc. We can easily integrate additional knowledge into the knowledge graphs. For example, consider that we want to monitor the status and control of a window in the building. We can inject the real-time data of the window into the knowledge graphs generated from the BIM model in the first phase. Similarly, we can integrate many technologies and external data sources with BIM and thus develop the digital twin of the smart building.

This phase aims to integrate multiple data sources and technology into the modular graphs generated from the BIM model using linked data principles.

New Knowledge Integration Tool - A tool should be made by combining the BIM model of the smart building and the BIM directory to integrate new knowledge. We selected two use cases for validating this phase: comfort parameter and Occupant monitoring (refer to section 2.4 for detailed use cases).In this phase, a tool needs to be created to integrate modular knowledge graphs of BIM and the above-said use cases.

1. **Comfort parameter monitoring -** Consider the sensors are already installed in the building. In this use case, we need to monitor the parameters like humidity, CO2 level, temperature, and luminosity, including window, door and heater status. Also, we need to integrate these data with the BIM directory for 3D visualization. We used W3C's Web of Things(WoT) concepts to integrate these data with the BIM model.

   As per WoT architecture, Every device has a metadata file that defines the way to access the data from that IoT device and is called Thing Description. With the help of thing description, we create knowledge graphs that contain the path to access the real-time and historical data of the comfort parameters and other things like a window, heater etc.

   RDF data model using thing description In the figure 6.4 illustrates the rdf model to extract real-time data of relative humidity using the sensor installed in the building using thing description. Similarly we can augment the way to access historical data of relative humidity to

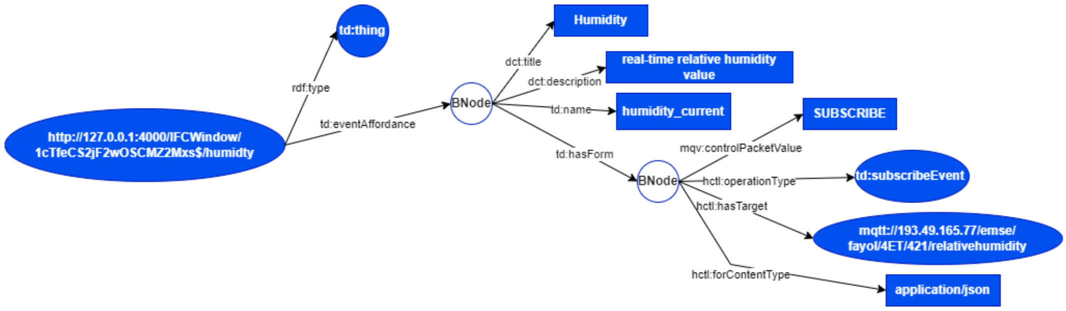

Figure 5.4. RDF data model using the thing description of humidity sensor - real-time data

the same RDF model.

Name Space required for the RDF model

```
@prefix td: <https://www.w3.org/2019/wot/td#> .
@prefix hctl: <https://www.w3.org/2019/wot/hypermedia#> .
@prefix jsonschema: <https://www.w3.org/2019/wot/json-schema#> .
@prefix mqv: <http://www.example.org/mqtt-binding#> .
@prefix htv: <http://www.w3.org/2011/http#> .
@prefix dct: <http://purl.org/dc/terms/> .
```

Listing 5.3. Namespaces used in the RDF model of Thing Description of a sensor

Thing Directory - The directory in which all thing description RDF models of all sensors in the smart building are generated and maintainedas turtle files. A Tool should be developed to generate and sustain knowledge graphs in the thing directory. A user can create a new thing description of a particular sensor in the building by giving the event affordance and property affordance with the help of the tool.

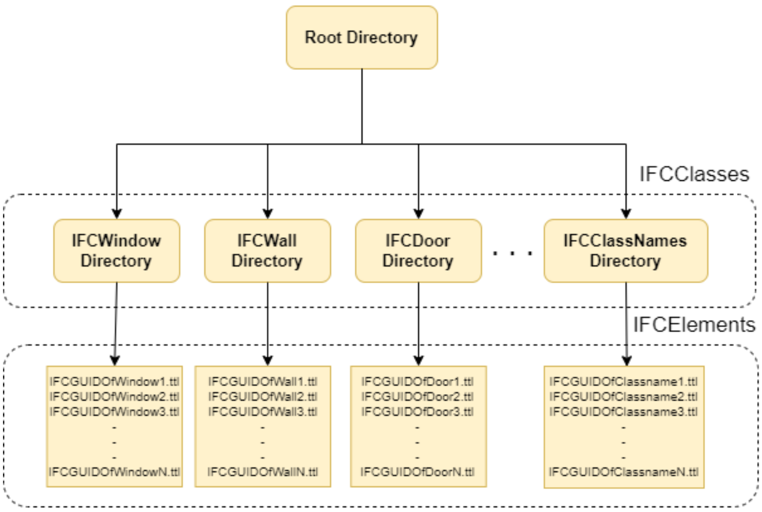

Figure 5.5. Structure of Thing Directory

REST API - All the thing description of sensors are published via REST API with the help of the IOT integration tool and thing directory.

All thing description RDF models are available in the below URL format .

```
http://serverIP:port/:IFCClass/:IFCGuid/:sensorName
```

Listing 5.4. Hyperlink used to publish thing description of every sensors

Integration of IOT directory with BIM directory All the knowledge graphs created in the IoT directory are integrated with the corresponding IFC elements in the BIM directory using the linked data principles. For example, REST API link of thing description of windows, sensors, heaters and other IoT devices in the building are added with the corresponding knowledge graphs of IFC elements in the BIM directory and vice versa.

Figure 5.6. Integration of thing description created in Thing directory with BIM Directory

```
@prefix dogont: <http://elite.polito.it/ontologies/dogont.owl> .
@prefix hctl: <https://www.w3.org/2019/wot/hypermedia#> .
@prefix saref: <https://saref.etsi.org/core/v3.1.1/> .
```

Listing 5.5. Namespaces used for the Integration

2. **Occupants Status -**

   The another use case is to get the occupants details with respect to the IFC model(refer section 2.4 for detailed explanation of this use case).

   RDF model - In the figure illustrates the RDF model of employee knowledge graph using FOAF.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

Listing 5.6. Namespaces used in employee RDF model

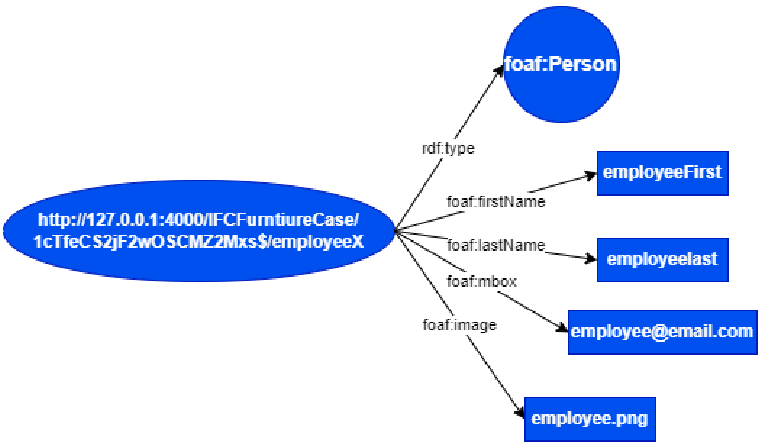

Figure 5.7. RDF model of one employee

Employee Directory - Similar to Thing directory, all knowledge graphs of employees are generated and maintained as turtle files in the employee directory.

REST API - All the knowledge graph of employees are published via REST API and accessed using the url formnat shown below:

```
http://serverIP:port/:IFCClass/:IFCGuid/:employeeName
```

Listing 5.7. Hyperlink to publish generated employee knowledge graphs

Integration of employee directory with BIM directory - Similar to thing directory, knowledge graphs in the employee directory are integrated to corresponding knowledge graphs of IFC elements in the BIM directory using the lined data principles.

5.4 Phase 3 - BIM Visualization

Visualization of BIM in the browser after integrating it with other technologies is the objective of this phase.

This phase has mainly two sub-steps:

1. **Integration of BIM model with a Web Framework -** The first phase needs to integrate the BIM model with a web framework. Here, the main challenge is integrating huge BIM models and visualizing them in a web browser. To solve this problem, first, we need to compress the BIM to another format without reducing the quality. For this purpose, we used a tool provided by Xeokit, which will convert the IFC model into another format called XKT, a compressed version of IFC.

   While compressing the IFC file to XKT, some information, like the IFCGuid of the elements and their corresponding IFC class, should be lost. So to solve this problem, we need to use another convert tool that extracts the IFCGuid and IFC class called IFC to Json metadata converter.

   The procedure to visualize BIM model are illustrated in the below diagram.

   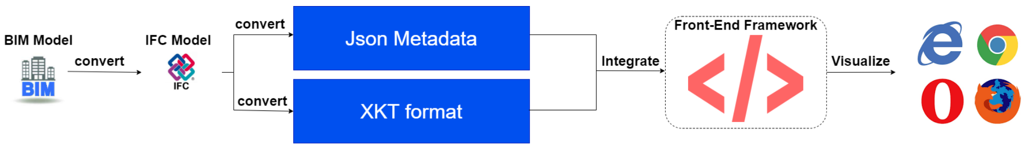

   Figure 5.8. Work Flow of BIM visualization in browser

2. **Find technologies and data sources integrated in the BIM knowledge graphs**

   From the first phase, all elements like a window, door, etc., have a knowledge graph and are published over REST API and BIM model are visualized in the web browser using the first step mentioned above.

   For implementing the use cases mentioned in section 2.4, we need to create a click event on every element in the BIM model. During the click event of an element, get the IFC class name and IFCguid, and read the BIM knowledge graph of that particular element. Using the SPARQL queries, get the URIs of external data sources integrated with that knowledge graph. Using the URIs, read and visualize the information in the knowledge graph through the platform.

   Common Steps to visualise the use cases mentioned in the section 2.4

   (a) Identify the knowledge integrated in each IFC classes manually.

   For example:

   i. window and door status monitoring - IFCWindow and IFCDoor class respectively

   ii. Comfort parameter monitoring - IFCSensor or IFCFlowTerminal

   iii. Occupants Monitoring - IFCFurnitureCase

   (b) Create click events on each IFC classes to read IFC class name and IFCguid of individual IFC elements.

   (c) Read the knowledge graphs using the REST API link with reference to IFC class and IFCguid when clicked on a IFC element in the BIM model using the visualization platform.

(d) Read the target URL of the external data source or technologies integrated with the knowledge graphs of each IFC elements using the SPARQL queries.

For example:

i. For Comfort parameter, window and door status monitoring used the thing description each. Multiple thing description may integrated with one IFC element.

ii. For occupants monitoring used a external employee directory which contain the basic information of all employees.

(e) Using the target URL(REST API) of the external data source get from the above step, read the knowledge graphs and extract useful information using the SPARQL Protocol and RDF Query Language (SPARQL) queries.

(f) Visualise information by creating the appropriate widgets.

Chapter 6

Implementation and Results

This chapter describes the implementation of the framework defined earlier with detailed explanation of the individual phases and also the validate the result.

For the implmentation we used two BIM models which mentioned in the figure 5.1 and 5.2 and convert the model into a open format called IFC.

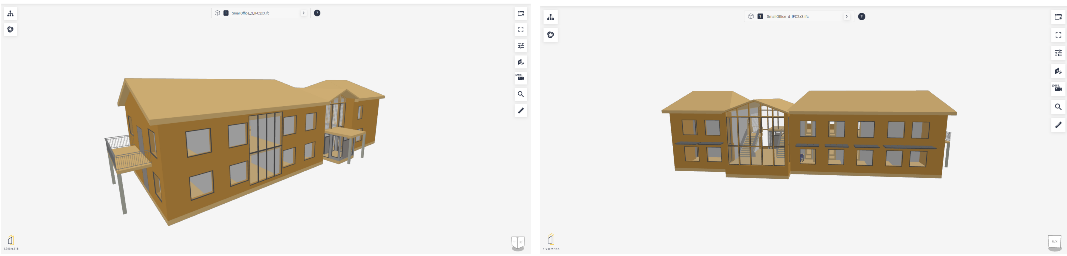

Figure 6.1. IFC model of Small Office building

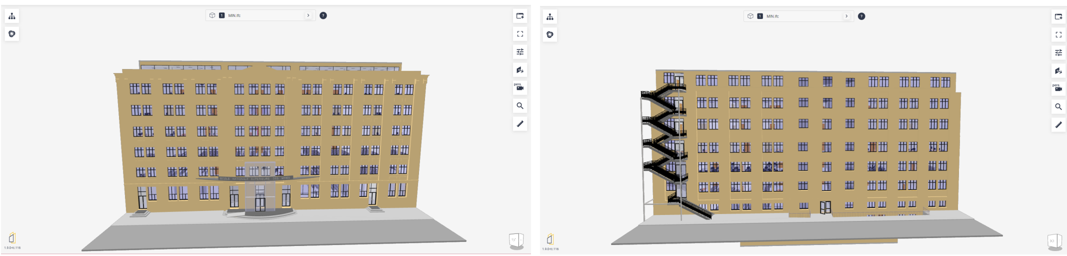

Figure 6.2. IFC model of Mines Espace Fauriel Builing

Note - Used IFC model of small office building and Mines Espace fauriel building to validate the Phase 1 and other 2 phases are validated using only the IFC model of Espace fauriel building.

6.1 Phase 1 - BIM to Modular knowledge graphs

6.1.1 Implementation

In the phase 1 , We need to develop a tool to convert the IFC model to modular knowledge graphs and also publish these graphs using the REST APIs .

bhy bvb yg vggbg For this phase mainly three steps are there .

(a) **Extraction of building information -** First step is to extract the building information from the IFC file . Here we used the Python programming language and one of its library called IfcOpenShell[1] to extract building information, object relationships and geometrical data.

[1] http://ifcopenshell.org/

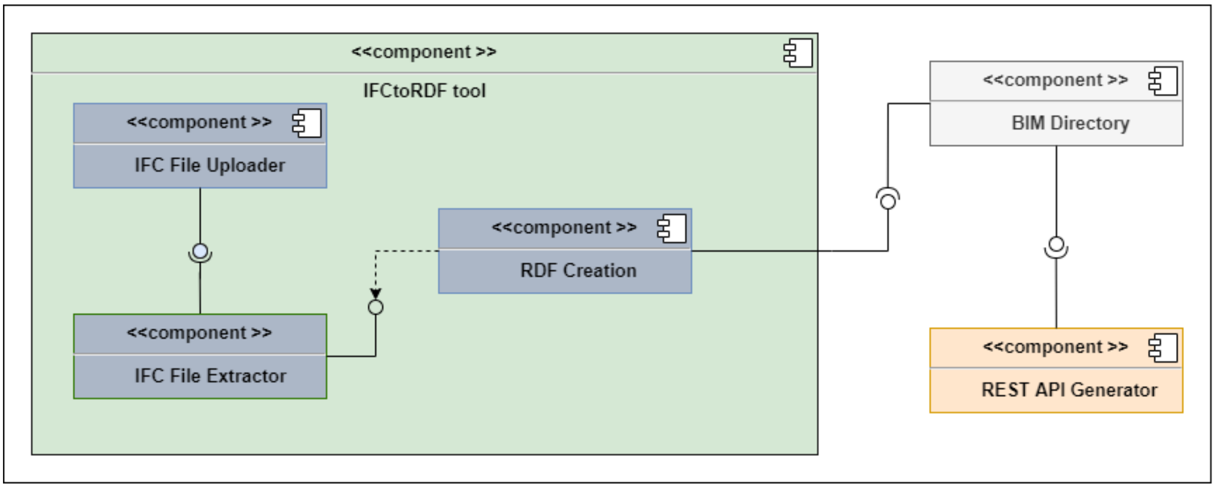

Figure 6.3. Component diagram to convert BIM to Modular knowledge graphs

(b) **Knowledge graph creation**[2] **-** For creating the Knowledge graphs, we used the rdflib[3] library and followed the RDF model defined in the previous chapter.All Knowledge graphs are created as a turtle files in the selected destination folder. The tool first create folders which named based on the IFC classes and then turtle files every elements of the selected IFC model are created in the folder of their corresponding class.

(c) **Publish knowledge graphs via REST API**[4] **-** Publish all knowledge graphs via REST API with reference to IFC class and IFCguid. Here I used the express.js[5] framework to develop the routes of all knowledge graphs.

Mainly, we developed three routes for publishing modular BIM knowledge graphs generated from the tool:

i. "fayol/IFCClass" - route to get the knowledge graph which contain the information of all IFC classes and its element count.

ii. "fayol/:IFCClassName" - route to get the knowledge graph of the selected IFC class and it contains URLs of the elements belongs to the same class.

iii. "fayol/:IFCClassName/:IFCGuid" - route to get the knowledge graph of the selected IFC element from the given IFC class name and IFCguid.

6.1.2 Results

Espace Fauriel Building - I attached the IFC model of the Espace fauriel building and BIM directory folder location and run the tool. Then the tool generate turtle files in the BIM directory and all the created knowledge graphs are published via REST API. Screenshots of the Result created are mentioned below .

[2]source code - **https://gitlab.emse.fr/isi/internships/master-raveendran-nair/-/tree/main/PythonFlask%20-%20Microservice**

[3]https://rdflib.readthedocs.io/en/stable/

[4]source code - **https://gitlab.emse.fr/isi/internships/master-raveendran-nair/-/tree/main/Microservice%20-2%20-%20Server**

[5]https://expressjs.com/

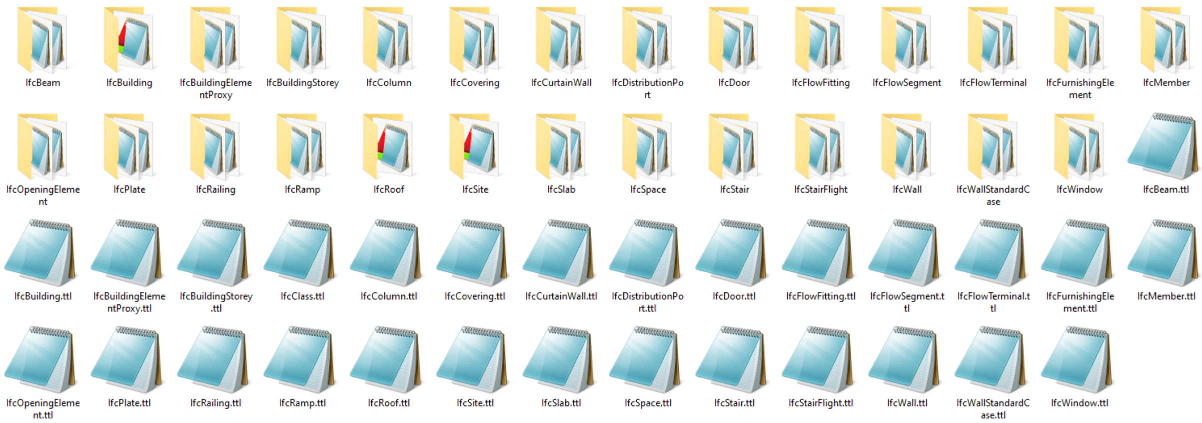

Figure 6.4. Screenshots of knowledge graphs created as turtle files for Espace Fauriel Building in BIM directory

No	IFC class name	IFC element count	Total knowledge graphs created
1	IfcWall	1727	1727
2	IfcCurtainWall	17	17
3	IfcRailing	61	61
4	IfcStairFlight	49	49
5	IfcFlowFitting	223	223
6	IfcFurnishingElement	1306	1306
7	IfcColumn	198	198
8	IfcSlab	88	88
9	IfcBuilding	1	1
10	IfcCovering	23	23
11	IfcRoof	1	1
12	IfcPlate	53	53
13	IfcStair	31	31
14	IfcDoor	324	324
15	IfcMember	384	384
16	IfcFlowSegment	279	279
17	IfcBuildingElementProxy	11	11
18	IfcWallStandardCase	1621	1621
19	IfcWindow	311	311
20	IfcOpeningElement	2940	2940
21	IfcSite	1	1
22	IfcBeam	100	100
23	IfcRamp	2	2
24	IfcSpace	213	213
25	IfcDistributionPort	1623	1623
26	IfcFlowTerminal	1281	1281
	Total	12,868	12,868

Here, the all modular knowledge graphs in the BIM directory are published via rest APIs.

Small Office Building - I repeated the same steps by attaching the IFC model of small office building and the location of new BIM directory.

The screenshots of the result was attached below:

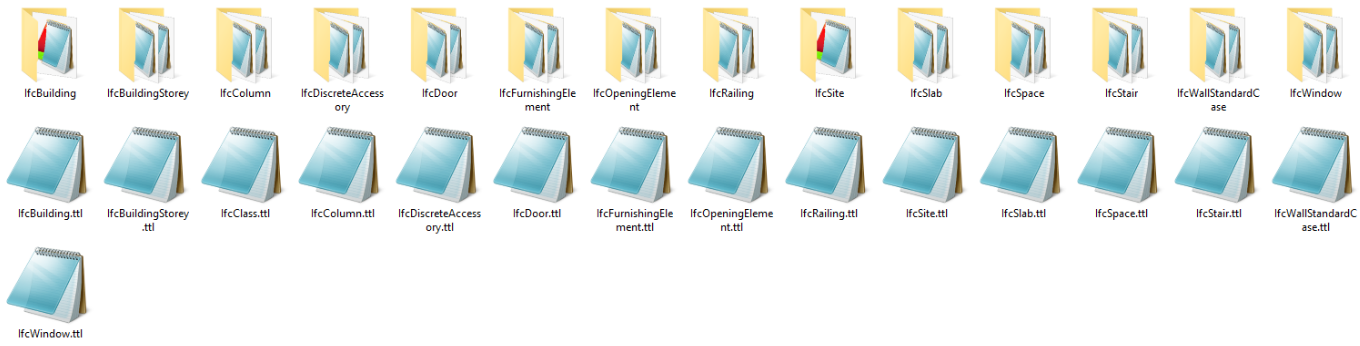

Figure 6.5. BIM directory

No	IFC class name	IFC element count	Total knowledge graphs created
1	IfcSite	1727	1727
2	IfcOpeningElement	17	17
3	IfcBuilding	61	61
4	IfcStairFlight	49	49
5	IfcFlowFitting	223	223
6	IfcFurnishingElement	1306	1306
7	IfcColumn	198	198
8	IfcSlab	88	88
9	IfcBuilding	1	1
10	IfcCovering	23	23
11	IfcRoof	1	1
12	IfcPlate	53	53
13	IfcStair	31	31
14	IfcDoor	324	324
	Total	4131	4131

All the create knowledge graphs are published via REST API. Here there are three variants of knowledge graphs.

(a) **Knowledge graphs to represent IFC classes and total elements count of each IFC class**

Example - Knowledge graph of above type generated from IFC model of Small Office building.

This knowledge graph is accessed via a hyperlink which is mentioned below:

```
    http://serverIP:port/ifcClass
```

Listing 6.1. REST API link to publish the below knowledge graph

```
@prefix ifcowl: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL>
@prefix props: <http://127.0.0.1:5000/test> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://127.0.0.1:5000/IfcClass> rdf:Property [ a ifcowl:IfcSite ;
            props:hasTotalElements 1 ],
        [ a ifcowl:IfcOpeningElement ;
            props:hasTotalElements 113 ],
        [ a ifcowl:IfcBuilding ;
            props:hasTotalElements 1 ],
        [ a ifcowl:IfcColumn ;
            props:hasTotalElements 6 ],
        [ a ifcowl:IfcWindow ;
            props:hasTotalElements 80 ],
        [ a ifcowl:IfcDiscreteAccessory ;
```

```
            props:hasTotalElements 6 ],
        [ a ifcowl:IfcRailing ;
            props:hasTotalElements 10 ],
        [ a ifcowl:IfcFurnishingElement ;
            props:hasTotalElements 10 ],
        [ a ifcowl:IfcSlab ;
            props:hasTotalElements 19 ],
        [ a ifcowl:IfcSpace ;
            props:hasTotalElements 29 ],
        [ a ifcowl:IfcBuildingStorey ;
            props:hasTotalElements 3 ],
        [ a ifcowl:IfcDoor ;
            props:hasTotalElements 33 ],
        [ a ifcowl:IfcStair ;
            props:hasTotalElements 3 ],
        [ a ifcowl:IfcWallStandardCase ;
            props:hasTotalElements 75 ] .
```

Listing 6.2. Query used in Semantic Scholar

(b) **Knowledge graphs to represent a IFC class and URI of the elements contained in that IFC class**

Example - Knowledge graph of above type generated from IFC model of Small Office building for IFCBuildingStorey class.

The knowledge graph is accessed via a hyperlink which is mentioned below:

```
     http://serverIP:port/IfcBuildingStorey
```

Listing 6.3. REST API link

```
@prefix ifcowl: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL> .
@prefix bot: <https://w3id.org/bot#> .
@prefix props: <http://127.0.0.1:5000/test> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://127.0.0.1:3000/IfcBuildingStorey> a bot:Element ;
    props:hasTotalElements 9 ;
    bot:containsElements <http://127.0.0.1:3000/IfcBuildingStorey/3_b98WEDT7feUaJ_WJe22_>,
        <http://127.0.0.1:3000/IfcBuildingStorey/3_b98WEDT7feUaJ_WJe2lo>,
        <http://127.0.0.1:3000/IfcBuildingStorey/3_b98WEDT7feUaJ_WJe3S2>,
        <http://127.0.0.1:3000/IfcBuildingStorey/3_b98WEDT7feUaJ_WJeW$O>,
        <http://127.0.0.1:3000/IfcBuildingStorey/3_b98WEDT7feUaJ_WJeW$i>,
        <http://127.0.0.1:3000/IfcBuildingStorey/3_b98WEDT7feUaJ_WJeW_M>,
        <http://127.0.0.1:3000/IfcBuildingStorey/3_b98WEDT7feUaJ_WJeWmQ>,
        <http://127.0.0.1:3000/IfcBuildingStorey/3_b98WEDT7feUaJ_WJeWog>,
        <http://127.0.0.1:3000/IfcBuildingStorey/3_b98WEDT7feUaJ_WJecOD> .
```

Listing 6.4. Query used in Semantic Scholar

(c) **Knowledge graphs to represent IFC elements, building data and the URI of the related elements**

Example - Knowledge graph of a IFC element belongs to IFC window class generated from IFC model of Small Office building .

The knowledge graph is accessed via a hyperlink which is mentioned below:

```
     http://serverIP:port/IfcWindow/0a_hlvWo11kemlZznb3myS
```

Listing 6.5. REST API link label

```
@prefix bot: <https://w3id.org/bot#> .
@prefix ifcowl: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL> .
@prefix props: <http://127.0.0.1:5000/test> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://127.0.0.1:3000/IfcWindow/0a_hlvWo11kemlZznb3myS> a ifcowl:IfcWindow,
        bot:Element ;
    props:hasAbattant true ;
    props:hasArea 1.04145205060607 ;
    props:hasAsoufflet true ;
    props:hasCategory "Windows"^^xsd:string ;
    props:hasEditedby "Aurelia.D"^^xsd:string ;
```

```
    props:hasEncadrementint rieur false ;
    props:hasFamily "1␣vantail␣-␣Fixe␣-␣Int rieur:␣0.50m␣x␣1.2m"^^xsd:string ;
    props:hasFamilyandType "1␣vantail␣-␣Fixe␣-␣Int rieur:␣0.50m␣x␣1.2m"^^xsd:string ;
    props:hasGeometryType [ props:hasFaces "[]" ;
            props:hasVertices "[]" ] ;
    props:hasHeadHeight 2.42699999999999 ;
    props:hasHeight 0.4 ;
    props:hasHostId "Basic␣Wall:␣Cloisons␣ep␣70"^^xsd:string ;
    props:hasIFCguid "0a_hlvWo11kemlZznb3myS"^^xsd:string ;
    props:hasIsExternal false ;
    props:hasLevel "Level:␣R+5"^^xsd:string ;
    props:hasMark "388"^^xsd:string ;
    props:hasPhaseCreated "Phase␣1"^^xsd:string ;
    props:hasReference "0.50m␣x␣1.2m"^^xsd:string ;
    props:hasSillHeight 2.02699999999999 ;
    props:hasThermalTransmittance 5.5617 ;
    props:hasType "1␣vantail␣-␣Fixe␣-␣Int rieur:␣0.50m␣x␣1.2m"^^xsd:string ;
    props:hasTypeId "1␣vantail␣-␣Fixe␣-␣Int rieur:␣0.50m␣x␣1.2m"^^xsd:string ;
    props:hasVolume 0.0196592860030806 ;
    props:hasWidth 1.1 ;
    props:hasWorkset "SECOND␣OEUVRE"^^xsd:string .
```

Listing 6.6. Query used in Semantic Scholar

6.2 Phase 2 - New Knowledge Augmentation

The main objective of this phase is to integrate the more knowledge to the modular graphs generated from the IFC file using linked data principles from many distributed data sources and technologies.

6.2.1 Implementation

We mainly consider two use cases defined earlier for implementing this phase, i.e., the integration of IoT and employee data. For the integration of IoT data, we used W3C's Web of Things architecture and created new data sources for storing employee information.

The main objective of this approach is to develop a tool in which a user can automatically create modular knowledge graphs that contain thing descriptions of all sensors installed in the building and publish the knowledge graphs via REST API. Also, link the thing description knowledge graph with the corresponding BIM knowledge graph and vice versa using URI. Similarly, using the tool user creates and links the employee knowledge graphs with BIM knowledge graphs.

We consider comfort parameters like temperature, humidity, and Window and door status as IoT data. Currently, several sensors are installed in the building to monitor these data, publish over MQTT protocol for real-time monitoring, and use a platform called OPENHab[6] for historical data. So create a description for every sensor that contains the way to access the real-time and historical data and publish the knowledge graphs via REST API. Then link each sensor's thing description knowledge graphs with corresponding knowledge graphs of the IFC element. In the IFC model, sensors have come under the IFCFlowTerminal class, Window under the IFCWindow class and Door under the IFCDoor class.

Tool to create knowledge graphs of IoT data and Employee information and integrate with BIM knowledge graphs The tool developed is an upgraded version of the one developed in the first phase, which creates new knowledge graphs for IoT and employees. Also, the tool integrates created knowledge graphs with the BIM knowledge graphs. Moreover, the tool has a user interface that helps to view the REST API of the knowledge graphs of IFC elements, see the knowledge graphs of individual IFC elements and create the thing description and

[6] https://www.openhab.org/

employee knowledge graphs by giving required data by the user. The user interface is developed using the IFC model of the building, which helps the virtual user interaction with the building elements. We used the vue.js framework for the user interface development and a library called Xeokit[7] to integrate the IFC model with vue.js. In the figure 6.6 illustrated the component diagram of the new knowledge augmentation tool.

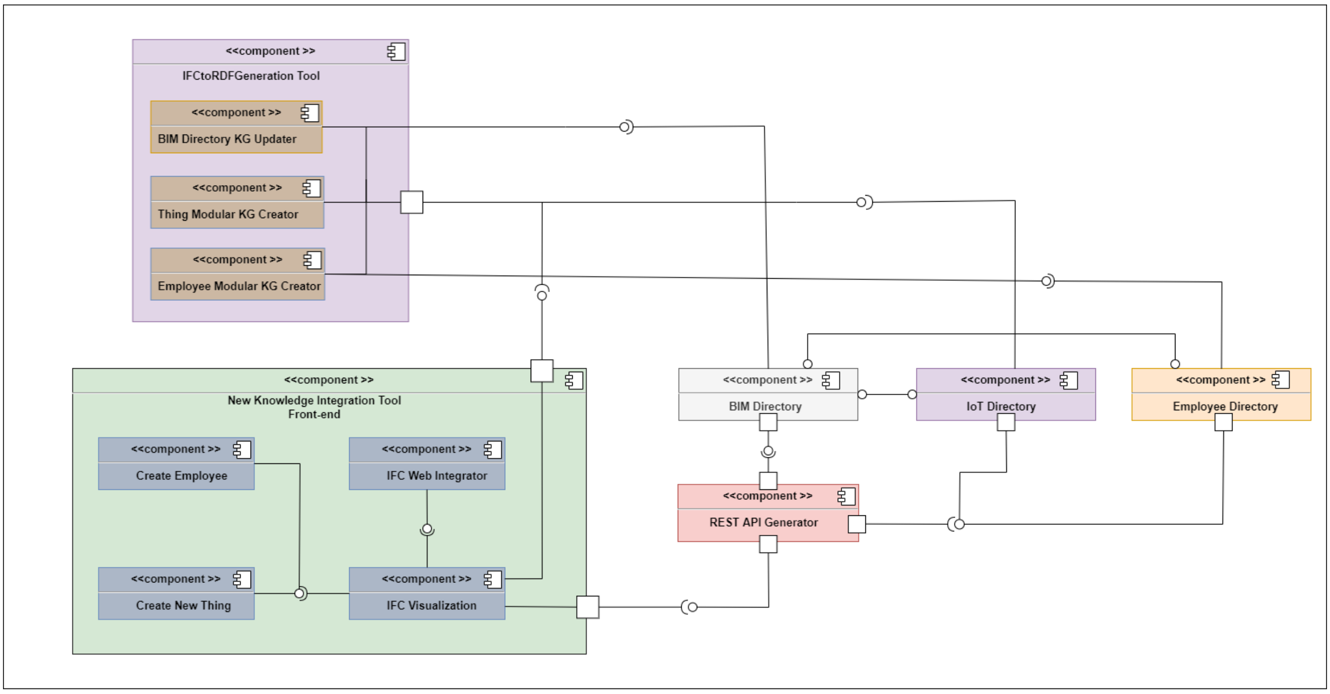

Figure 6.6. Component Diagram of New Knowledge Augmentation tool

The screenshot of the IFC model of Espace Fauriel Building visualized in the browser are shown in the figure 6.7.

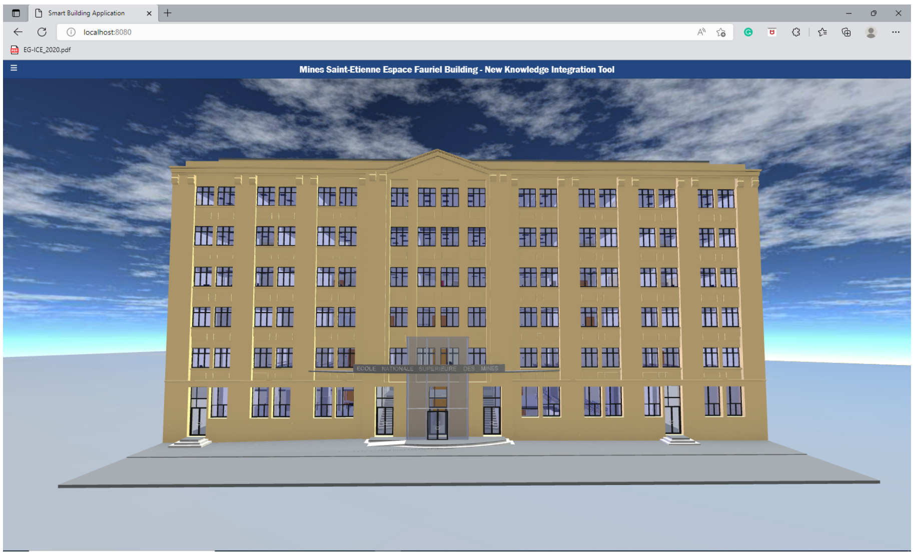

Figure 6.7. BIM visualization in New Knowledge Augmentation Tool

[7] https://xeokit.io/

6.2.2 Results

Using the New Knowledge Augmentation tool we can see the REST API Link and knowledge graphs of individual IFC element by selection.In figure 6.8 show the selection of a window elements in the IFC model(green color).

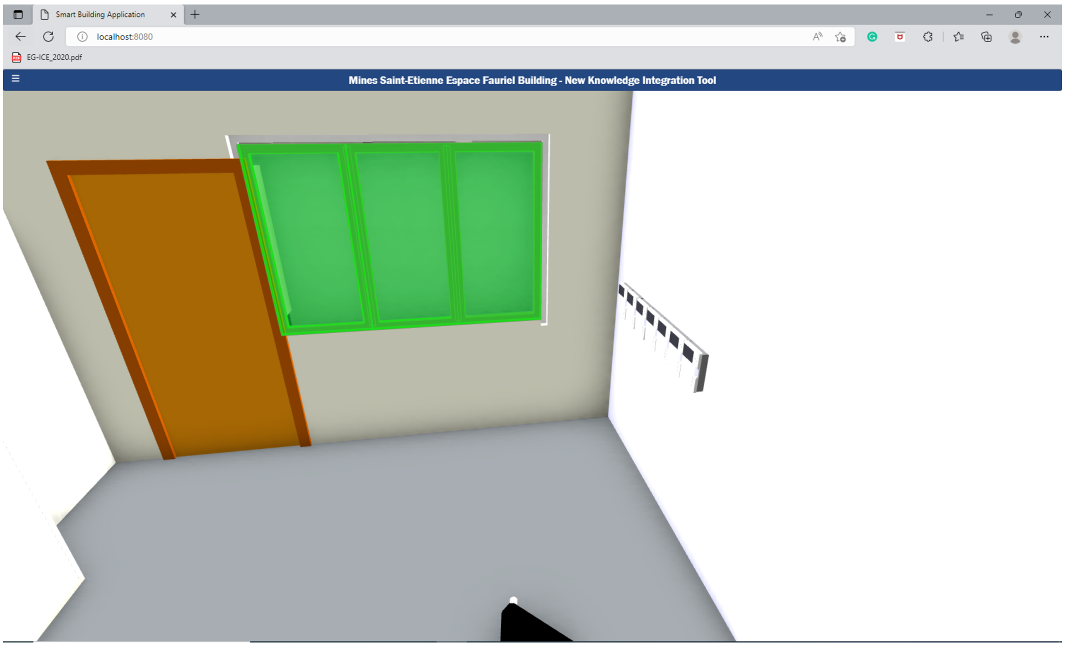

Figure 6.8. Figure to select IFC elements

When we clicked on the selected IFC element, a popup display will come to show the REST API and the knowledge graphs of the selected element. The figure 6.9 shows the screenshot the popup display shown when a window object was selected and clicked.

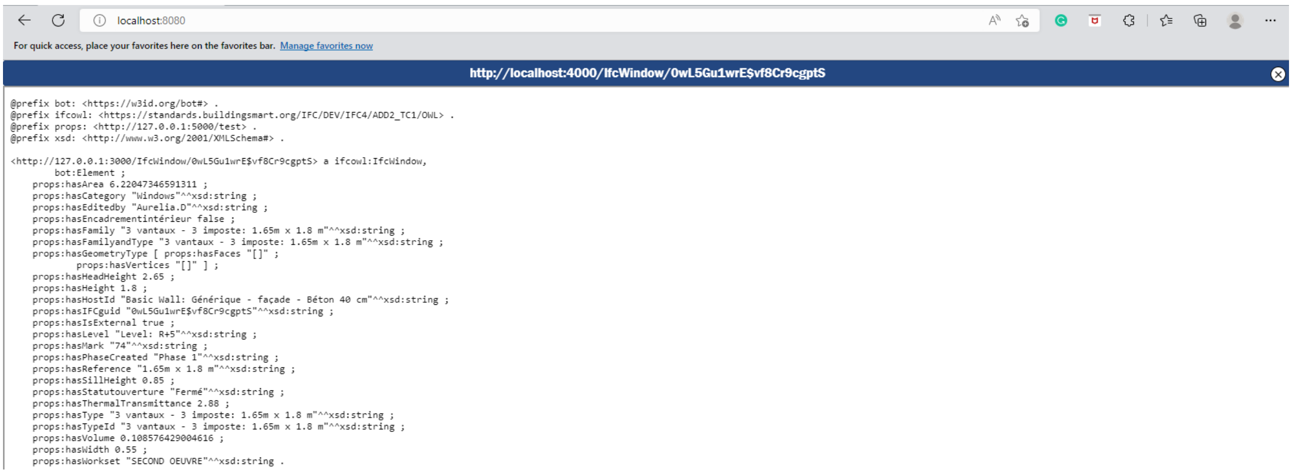

Figure 6.9. Figure to see the knowledge graph of selected IFC elements

Add New Knowledge

We give a context menu which will popup when we do mouse right-click from a selected IFC element. In the context menus there are options to create and link thing description and employee details to the selected IFC element.The screenshot the context menu popup are shown in the figure 6.10.

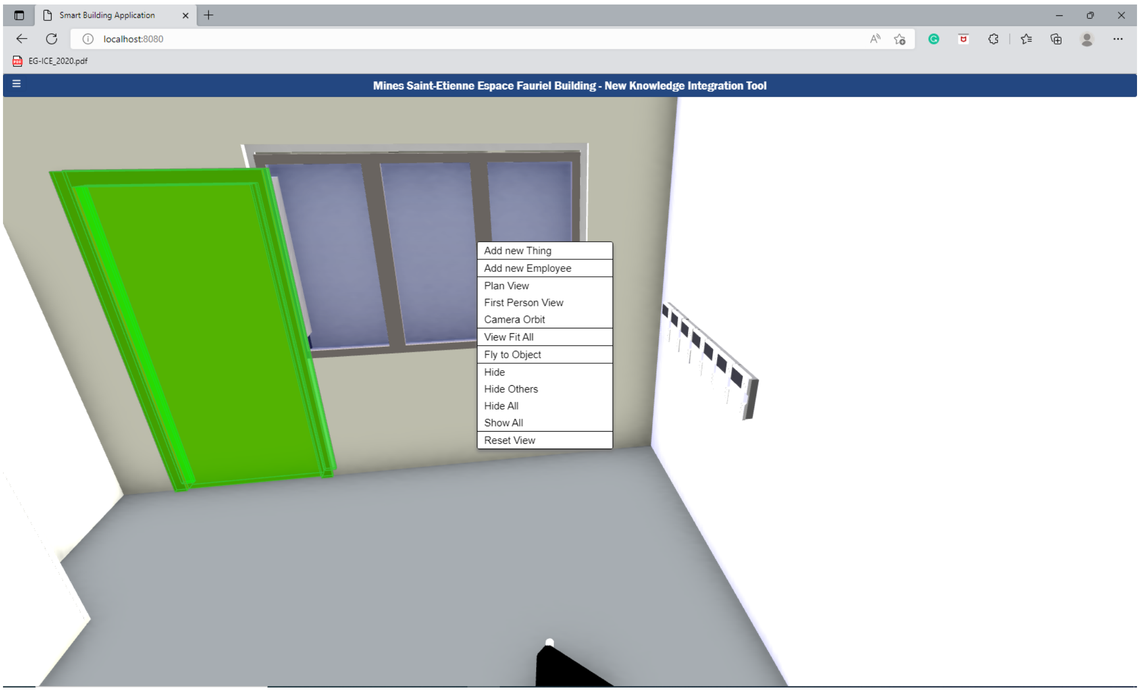

Figure 6.10. Figure of context menu to create thing and employee KG

When we select new thing from the context menu then one popup window will display. There we need to enter the contextual and the metadata of the sensors.We can create and link multiple sensors with a single IFC element. The screenshot of the popup display to add new thing is show in the figure 6.11.

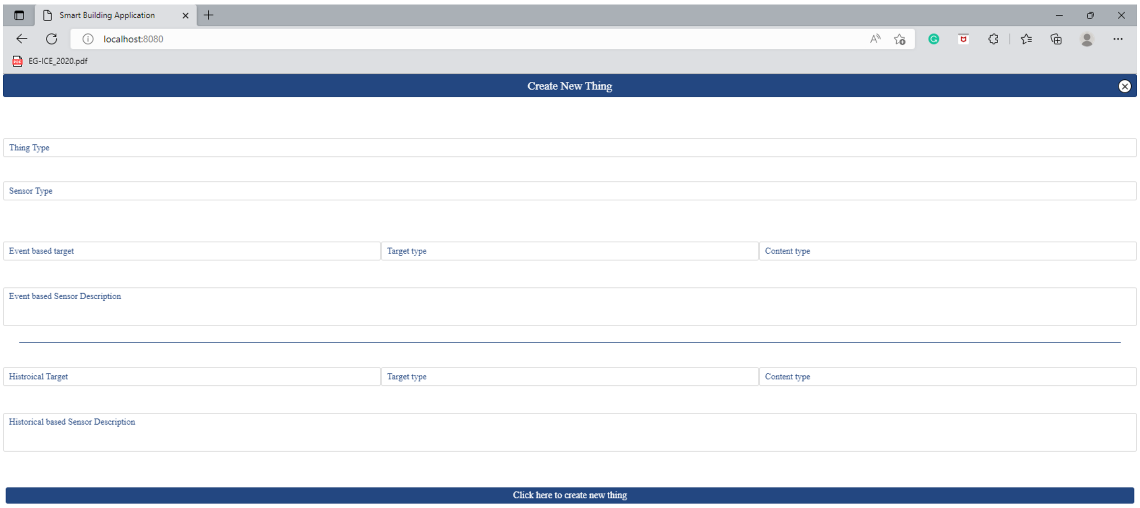

Figure 6.11. Figure to enter thing description of individual IFC elements

The tool automatically created the thing description RDF graph in the thing directory using the information provided in the popup display and publish it via REST API.The listing 6.7 shown the thing description created for a sensor. Also, the created thing description is linked with the corresponding knowledge graphs of the IFC element in the BIM directory by providing the REST API link and vice versa.

```
@prefix dct: <http://purl.org/dc/terms/> .
@prefix hctl: <https://www.w3.org/2019/wot/hypermedia#> .
```

```
@prefix htv: <http://www.w3.org/2011/http#> .
@prefix jsonSchema: <https://www.w3.org/2019/wot/json-schema#> .
@prefix mqv: <http://www.example.org/mqtt-binding#> .
@prefix td: <https://www.w3.org/2019/wot/td#> .

<IfcFlowTerminal/1bDMdL0k55X8oOMH5VK_cb/ESP32> a td:Thing ;
    td:hasEventAffordance [ dct:description "sdf" ;
            dct:name "Relatice␣Humidity␣current" ;
            dct:title "Relatice␣Humidity␣" ;
            td:hasForm [ mqv:controlPacketValue "Subscribe" ;
                    hctl:forContentType "Application/json" ;
                    hctl:hasOperationType td:subscribeEvent ;
                    hctl:hasTarget <MQTT> ] ;
            td:hasNotificationSchema [ a jsonSchema:NumberSchema ] ] ;
    td:hasPropertyAffordance [ a jsonSchema:ObjectSchema ;
            dct:description "sddflhuogdfughyuifdg" ;
            dct:title "Relatice␣Humidity␣-(HTTP␣binding)" ;
            td:hasForm [ htv:methodName "get" ;
                    hctl:forContentType "application/json" ;
                    hctl:hasOperationType td:readProperty ;
                    hctl:hasTarget <http> ] ;
            td:name "Relatice␣Humidity␣history" ] .
```

Listing 6.7. RDF graphs generated for a single sensor using thing description from the tool

Similarly the knowledge graph of employee information is created in employee directory and linked with the knowledge graph of selected IFC element usinf the REST API link of the employee knowledge graph with the help of new knowledge augmentation tool.The figure 6.12 shows the screenshot of the popup display when add new employee was selected from the context menu.

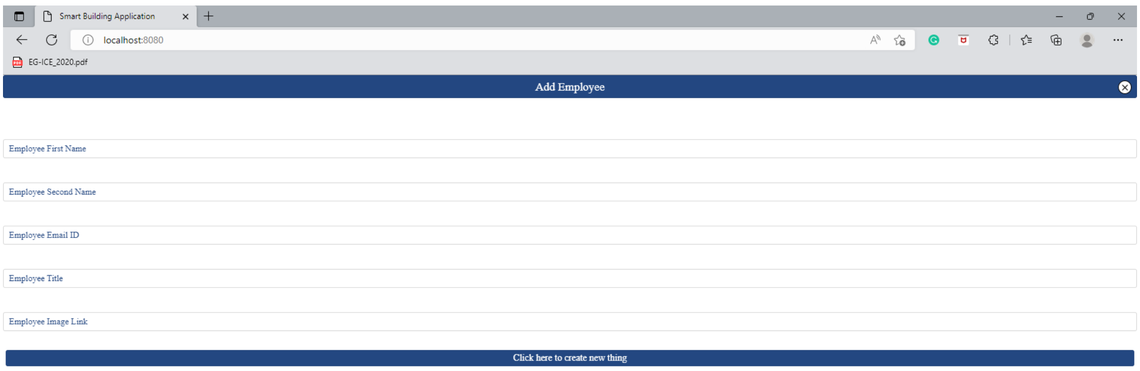

Figure 6.12. Figure to enter employee details of individual IFC elements

In the listing 6.8, shows the knowledge graph of one employee that was created using the tool.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<IfcWindow/1cTfeCS2jF2wOSCMZ2Mxlz/employee> a foaf:Person ;
    foaf:firstName "Arun" ;
    foaf:img "arun.png" ;
    foaf:lastName "Vishnu" ;
    foaf:mbox "arunvishnu40@gmail.com" ;
    foaf:title "intern" .
```

Listing 6.8. Generated knowledge graph for a employee

6.3 Phase 3 - Visualization of Digital Twin in browser

The main objective of this phase is develop a platform to visualise the digital twin of a smart building using the IFC model of building with the help iof data sources created from the first and second phase.

6.3.1 Implementation

This phase has two sub phases that are explained below:

Integration of BIM with a front-end framework We need to integrate BIM with a front-end framework to visualize it in the browser. We selected vue.js as a front-end framework for the development. Nevertheless, the size of the BIM model is the main challenge. Refer table to see the size of some models

So as discussed in section 6.4, converting the IFC into XKT is the primary task of this sub-phase.To do this, a library called xeokit-convert is available as an npm package to convert the IFC model to XKT. Nevertheless, the conversion using the xeokit-convert does not produce accurate output. So we used a platform called BIMdata.io, a cloud platform that helps visualize the IFC model. We need to upload our IFC model to this platform and download it as an XKT file. Also, here we need another tool for converting IFC to JSON metadata[8] to provide the IFC class and IFCguid of each element to the XKT file.

Now we need to integrate the converted XKT file with the vue.js framework for visualizing it in the browser. We used an open-source library called Xeokit-SDK[9] to integrate XKT and JSON metadata with vue.js.

We also implemented some features with the help of Xeokit-SDK for this platform which was explained below.

- Walk through navigation which helps navigate and virtual visit every rooms using the arrow buttons in the keyboard
- Individual floor visualization with sliced top view
- A command window is provided to the user to interact with the IFC model
- Navigate to individual rooms using the context menu
- Get the individual IFC elements properties by selecting the elements

Also there are other features which was explained in details in the appendex section.

The component diagram of the of the whole process to develop the digital twin are illustrated in the figure 6.13.

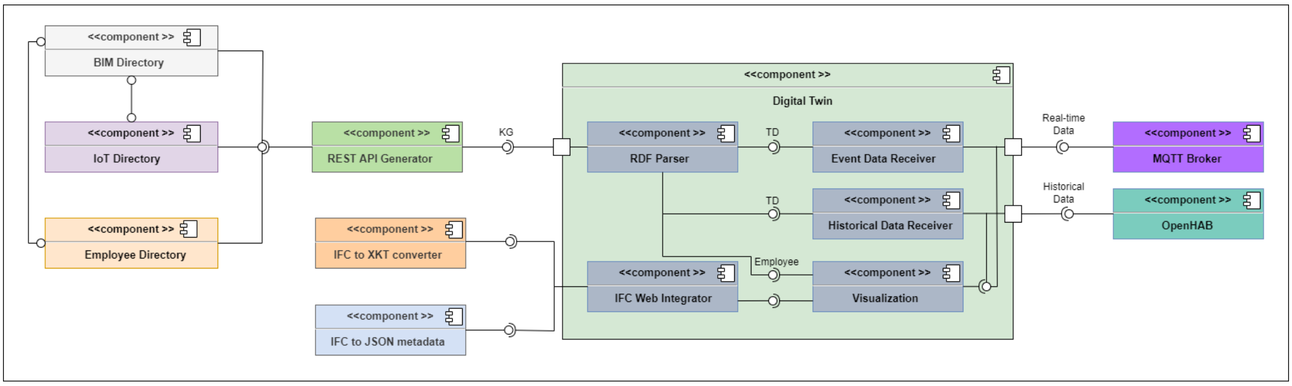

Figure 6.13. Figure to show the component diagram of Digital Twin

8

9 https://xeokit.io/

Find technologies and data sources integrated in the BIM knowledge graphs

There are two methods to find the technologies and data sources integrated in the knowledge graphs of IFC elements in the BIM directory.

(a) With the help rdflib library, we extract the required information from the knowledge graph. We must find the technologies and data sources integrated with the BIM knowledge graphs. From the use cases we defined for this work, mainly thing description of the sensors and the actuator from the thing directory and employee information from the employee directory are the data sources integrated with the BIM knowledge graphs. When we click any element in the IFC model visualized in the browser, then using the IFC class name and IFCguid of the selected element, we get the knowledge graph published via REST API. From the knowledge graph, we get the information of all external sources integrated with the IFC element. If it is a thing description, we will extract the sensor/actuator information and path to access historical and real-time data. Similarly, we extract the employee data linked with the BIM knowledge graphs.

(b) With the help of SPARQL queries also, we can extract information from the BIM knowledge graph. Note - Implementation via SPARQL query method can't be able to perform because currently there was no SPARQL endpoints configured.So we followed method one for implementing phase 3.

Considering the use cases we design some SPARQL queries that are explained below.

SPARQL query to get the total sensors linked with one IFC element:

```
PREFIX saref: <https://saref.etsi.org/core/v3.1.1/>
PREFIX hctl: <https://www.w3.org/2019/wot/hypermedia#>
PREFIX dogont: <http://elite.polito.it/ontologies/dogont.owl>
SELECT ?URI ?localURL ?sensorName ?target
WHERE {
?URI dogont:hasSensor ?localURL .
?localURI saref:hasSensorType ?sensorName .
?localURI hctl:hasTarget ?target .
}
LIMIT 3
```

Listing 6.9. SPARQL query to get the sensors

SPARQL query to get the description of a sensor to access the real-time and historical data:

```
PREFIX mqv: <http://www.example.org/mqtt-binding#>
PREFIX td: <https://www.w3.org/2019/wot/td#>
PREFIX hctl: <https://www.w3.org/2019/wot/hypermedia#>
PREFIX htv: <http://www.w3.org/2011/http#>
PREFIX dct: <http://purl.org/dc/terms/>
SELECT ?URI ?eventForm ?eventTD ?eventTarget ?eventType ?historicalTarget
?historicalForm ?historicalTD ?historicalMethod ?title
WHERE {
    ?URI td:hasEventAffordance ?eventForm .
    ?eventForm dct:title ?title .
    ?eventForm td:hasForm ?eventTD .
    ?eventTD mqv:controlPacketValue  ?eventType .
    ?eventTD hctl:hasTarget ?eventTarget .
    ?URI td:hasPropertyAffordance ?historicalForm .
    ?historicalForm td:hasForm ?historicalTD .'
    ?historicalTD htv:methodName ?historicalMethod .
    ?historicalTD hctl:hasTarget ?historicalTarget .
} LIMIT 3
```

Listing 6.10. SPARQL query

SPARQL query to access the employee URL linked in a IFC element:

```
PREFIX cwrc: <http://sparql.cwrc.ca/ontologies/cwrc#>
SELECT ?url ?targetURL
WHERE {
 ?url cwrc:hasEmployee ?targetURL
} LIMIT 1
```

Listing 6.11. SPARQL query

SPARQL query to access the basic information of one employee:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?url  ?firstName ?lastName ?img ?mbox ?title
WHERE {
   ?url foaf:firstName ?firstName .
   ?url foaf:img ?lastName .
   ?url foaf:lastName ?img .
   ?url foaf:mbox ?mbox .
   ?url foaf:title ?title .
} """
```

Listing 6.12. SPARQL query

6.3.2 Results

Visualization of the IFC model and created widgets to display data integrated with BIM knowledge graphs are explained below:

Visualization of IFC model in browser

The figure 6.14 shows the different screen shots of visualizing IFC model of Espace Fauriel Building in the web browser.

Figure 6.14. Screenshots of IFC model visualized in browser

Monitor window status

The screenshot of widgets create to monitor window open/close status are shown in the diagram 6.15.From the figure red colour indicates the closed state and green color indicates the open state of the window.

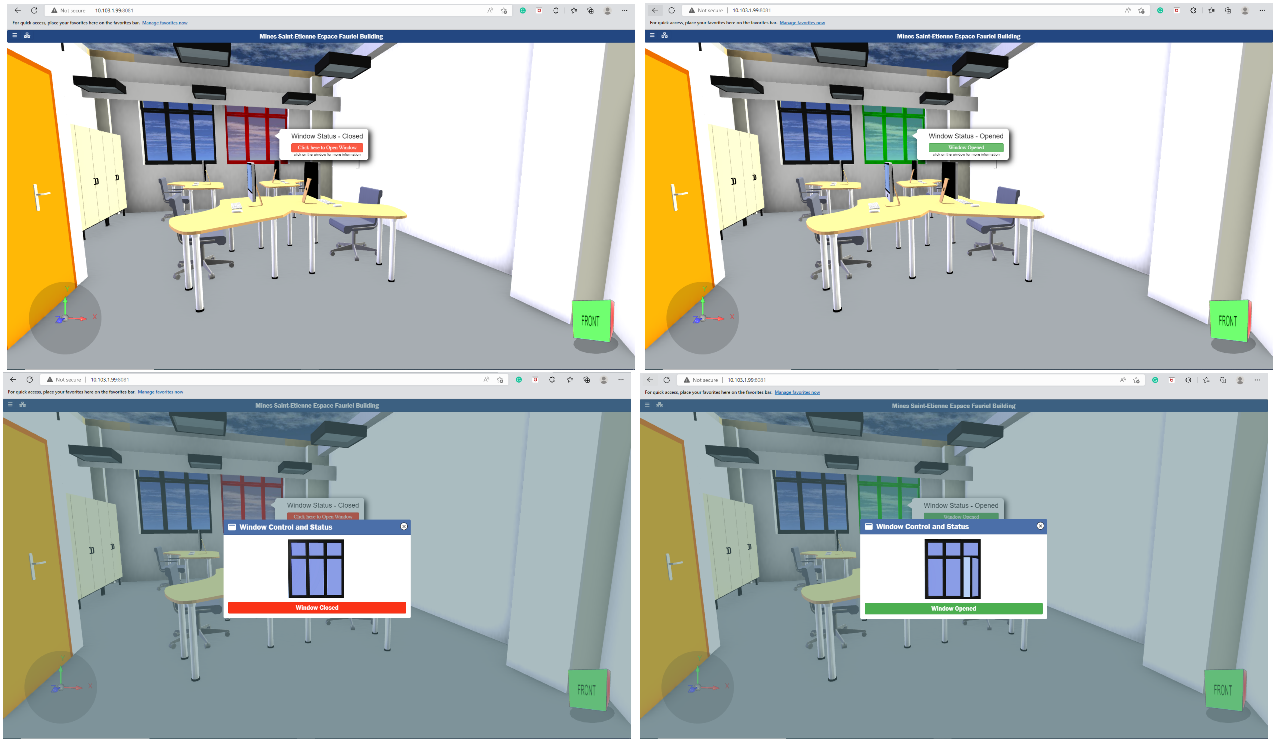

Figure 6.15. Screenshots to monitor window status through IFC model

Monitor Comfort parameters - Real-time data

The figure 6.16 shows the screenshots of the widgets created to monitor real-time data of the comfort parameters.We used a JavaScript library called EChart.js to create the widgets.

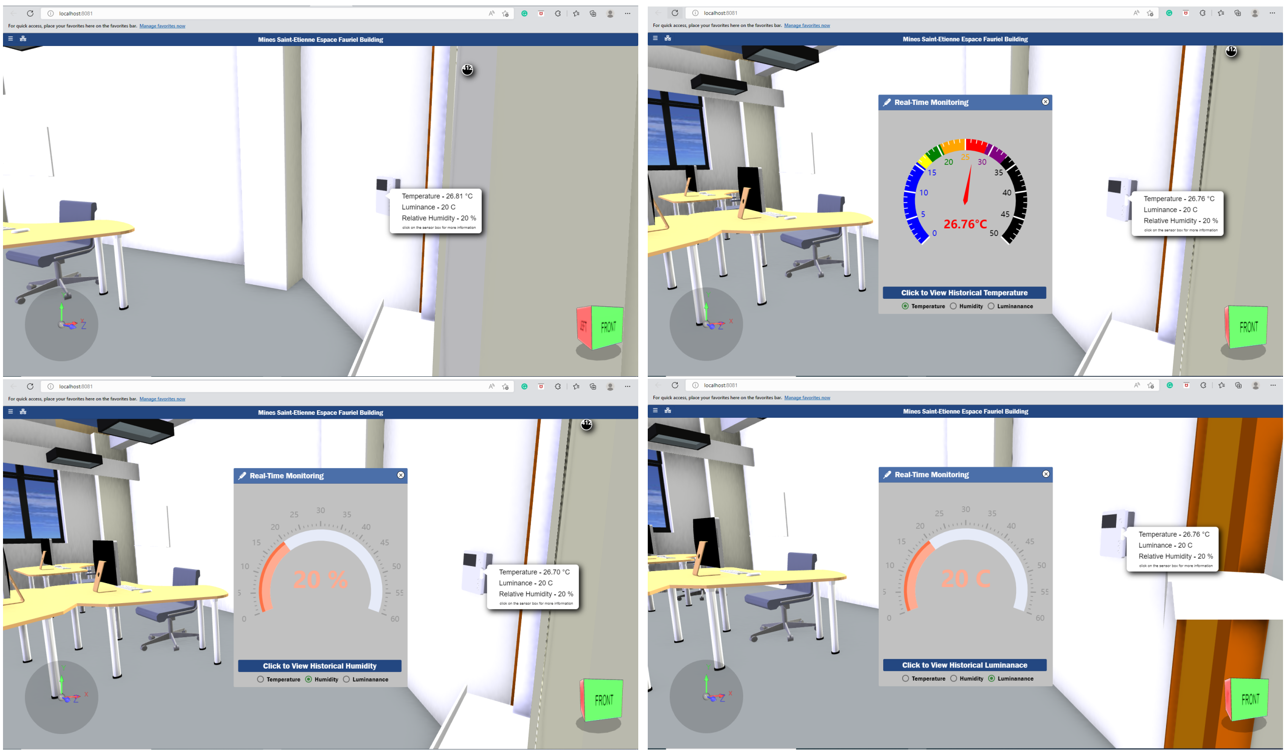

Figure 6.16. Screenshots to monitor real-time data through IFC model

Monitor Comfort parameters - Historical data

The widget to visualize the historical data of comfort parameters are shown in the figure 6.17.

Figure 6.17. Screenshots to monitor historical data through IFC model

Occupants Monitoring

The figure 6.18 shows the screenshot of widgets for occupant monitoring.

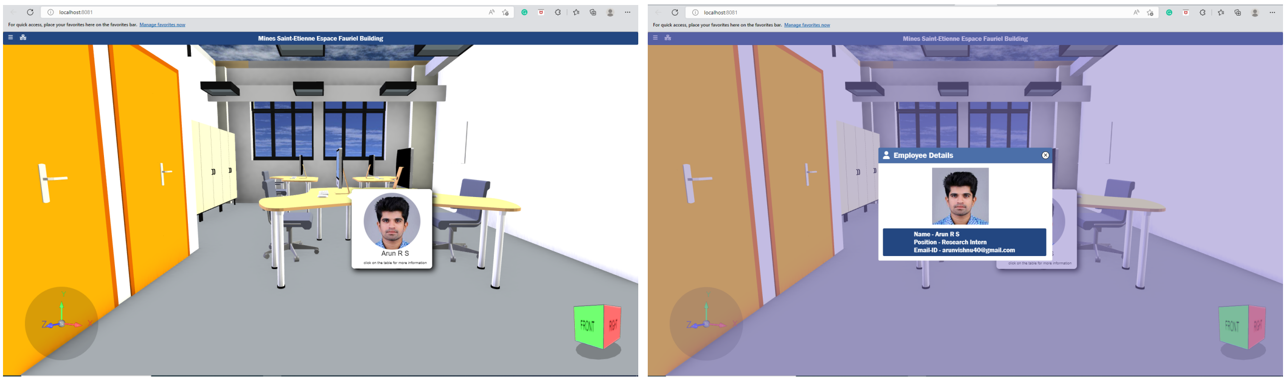

Figure 6.18. Screenshots to monitor occupant's data through IFC model

Discussion

In this work, we tried to find a method to integrate BIM data with other technologies and data sources and develop the digital twin of a smart building. Considering the above objective, we defined one main research question and three sub-research questions. We answered all the three research questions defined and created a framework for developing a digital twin by integrating other technologies and data sources with the BIM model of a smart building.

No.	Sub-Research Question	Answer
1	SRQ - 1	Phase1 - BIM to Modular knowledge graphs
2	SRQ - 2	Phase2 - New Knowledge Augmentation
3	SRQ - 3	Phase3 - Digital Twin visualization in browser

Table 8.1. Sub-Research questions and Answer

8.1 Comparison with existing works and overview of results

In this section, we tried to compare our framework with other existing work with the help of different tables. In the related work section, we reviewed the papers for the three phases of this research work(refer to chapter 4). In the same way, we compare the three phases individually with other related works.

8.1.1 Phase 1 - BIM to Modular Knowledge graphs

In this phase we compare the existing tools which convert BIM to RDF and platforms to access contextual building data from BIM.

Existing tools - The table 8.2 compares the differences of the existing tools and the tool developed by us to convert BIM to RDF:

	IFC to RDF	IFC to LBD	IFC to Modular RDF(our tool)
Modular Desgin	No	No	Yes
Contains IFC property sets	No	Yes	Yes
Publish generated RDF via REST API	No	No	Yes
Contain IFC element relationships	Yes	Yes	Yes
Exact RDF serialization of the IFC schema	Yes	No	No

Table 8.2. Comparison of existing tools with our tool for IFC to RDF conversion

Similarly, there are several platforms were introduced to exchange building data globally. The table 8.3 compares these platform with the phase one of this research work:

	BIMServer	DRUMBEAT	SCOPE	LBDserver	OurPlatform
IFC format as input	Yes	Yes	No	Yes	Yes
Get Building Data via REST interface	No	Yes	Yes	No	Yes
Query service to get building data	Yes	No	Yes	Yes	No

Table 8.3. comparison of existing platforms for accessing buidling data through BIM w

Result Overview -

(a) Developed a tool to convert BIM to modular knowledge graphs and save the knowledge graphs as turtle files in BIM directory

(b) Published all knowledge graphs via REST API.

8.1.2 Phase 2 - New Knowledge Augmentation

Phase 2 fully focuses on creating distributed data sources and integrating them with modular knowledge graphs of BIM through linked data principles. We created two distributed data sources from the selected use cases for accessing data from IoT devices called Thing Directory and storing employee information called employee directory. Thing Directory used the first two building blocks of W3c's WoT architecture, i.e Thing Description and Binding Templates.

The table 8.4 compares some existing works with phase 2:

	[24]	[27]	[25]	[26]	Phase 2
Used WoT Architecture	No	No	No	Yes	Yes
Digital Twin with Distributed System Architecture	No	Yes	No	No	Yes
BIM - IoT Integration Based on Linked Data	Yes	Yes	Yes	Yes	Yes
BIM Visualization with IoT data	No	Yes	No	No	Yes

Table 8.4. Phase 2 Comparison

Result Overview

(a) Provided a tool to see the knowledge graphs and REST API of individual IFC elements through a 3D visualization platform.

(b) A feature to generate knowledge graphs of thing description of sensor and employee information automatically and integrate with the knowledge graph of corresponding IFC element in the BIM directory with the help of the tool.

8.1.3 Phase 3 - Digital Twin Visualization in browser

The table 8.5 compares the feature of this phase 3 with other similar works:

	[28]	[29]	[27]	[25]	[30]	Phase 3
BIM Visualization in browser	No	Yes	No	No	Yes	Yes
Visualization of Large BIM model in browser	No	Yes	No	No	No	Yes
Visualization of real-time building sensor data through BIM	Yes	Yes	Yes	No	Yes	Yes
Visualization of historical building sensor data through BIM	Yes	Yes	Yes	No	No	Yes

Table 8.5. Phase 3 Comparison

Result Overview

(a) Visualization of very large BIM models in a browser

(b) Visualization of real-time and historical monitoring of building sensor data through BIM assistance.

(c) Monitoring door and window status through BIM

(d) Visualization of occupants information in a building through BIM

8.2 Advantages

(a) Modular design pattern was followed.

(b) Distributed based architecture(different data sources and technologies are maintained on different databases and system and all these are connected together through linked data).

(c) Visualize the digital twin of a building in a browser.

(d) Any external data sources and technologies can be integrated with BIM data.

8.3 Limitations

(a) Triple store is not used for storing Knowledge graphs.

(b) Lack of Security and Trust.

Chapter 8

Future plan

As this work is only an initial step for developing digital twins from BIM models, future research can go in various directions. Some of the ideas that can push the work further are listed below:

Research Ideas

(a) Find more use cases related to decision-making, predictive algorithms, simulations, etc., and integrate with BIM using the framework.

(b) Security and Trust is the main concern of this research framework. So upgrade the proposed framework by using Blockchain technology for sharing and integrating knowledge.

Future Development ideas

(a) Currently, knowledge graphs are created and maintained as turtle files. Instead of that, provide a triple store.

(b) Create separate micro-services to access data from every technologies and data source integrated with the BIM directory. This will improve the performance of the digital twin visualization further.

Chapter 9

Conclusion

This work presented a workflow to develop the digital twin of smart buildings from the selected BIM model. To reduce the complexity, we categorized the workflow into three phases. In the first phase, BIM to Modular RDF graphs, we provided a tool to convert the BIM into modular knowledge graphs and publish the knowledge graphs via REST API. Similarly, in the second phase, new knowledge augmentation, we completed a proof of concept by integrating new knowledge into the BIM knowledge graphs by considering the use cases defined with the help of a tool. Also, In the third phase, we developed platform to visualize the BIM in the browser with the assistance of the knowledge graphs generated in the previous steps. Also, we visualized the real-time and historical data coming from the building sensors through platform and constructed the digital twin of a smart building.

Even though we need to address some concerns like security issues, performance of the platform etc. Also, we need to implement more smart use cases by following the workflow proposed by this work.

Bibliography

[1] Dimitrios D. Vergados Emmanouil Skondras, Theodoros Mavroeidakos. The convergence of blockchain, internet of things (iot) and building information modeling (bim): The smart museum case. In *Wireless Telecommunications Symposium*, April 2019.

[2] K. Barlish and K. Sullivan. “how to measure the benefits of bim—a case study approach. In *Automation in Construction*, vol. 24, pages 149–159, 2012.

[3] Rafael Sacks Kathleen Liston Chuck Eastman, Paul Teicholz. “bim handbook: A guide to building information modeling for owners, mangers, designers, engineers, and constructors. In *Automation in Construction*.

[4] Jdesigningbuildings. Digital model. In *designingbuildings.co.uk/wiki/DigitalModel*, 23 Dec 2021.

[5] Charles Day Chris Barlow Aidan Fuller, Zhong Fan. Digital twin:enabling technologies , challenges and open research. In *IEEE*.

[6] IGI Global. Digital shadow. In *https://www.igi-global.com/dictionary/digital-twin-for-smart-school-buildings/97586*, 2021.

[7] ISOl. Digital twin. In *https://www.iso.org/obp/uiiso:std:iso:6707:-4:ed-1:v1:en*, 2021.

[8] Johnson controls. Smart building architecture. In *https://www.johnsoncontrols.com/digital-solutions/smart-buildings*, 2021.

[9] Autodesk Revit. Autodesk revit ifc instruction manual.

[10] BY THOUGHTWIRE. Smart building digital twin. In *https://blog.thoughtwire.com/what-is-a-smart-building-digital-twin*, 2021.

[11] Qingcheng Zeng Yuhui Sun Ang Yang, Mingzhe Han. Adopting building information modeling(bim) for the development of smart builidng: A review of enabling applications and challenges. In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 1*, BCS-HCI ’08, pages 13–22, Swinton, UK, UK, 2008. British Computer Society.

[12] Buildingsmart international - ifc open standard. 2013.

[13] Carlo Rainieri Giovanni Fabbrocino Salvatore Gerbino, Luigi Cieri. On bim interoperability via the ifc standard: An assessment from the structural engineering and design viewpoint. In *MDPI*, 22 November 2021.

[14] Walter Terkaj Pieter Pauwels. Express to owl fro construction industry: Towards a recommendable and usable ifcowl ontology. In *Automation in Construction*, March 2016.

[15] Maxime Lefrançois Georg Ferdinand Schneider Mads Holten Rasmussen and Pieter Pauwels. Bot: the building topology ontology of the w3c linked building data group. In *Semantic Web*, 27 November 2016.

[16] Arka Bhattacharyab Jingkun Gao Joshua Gluck Dezhi Hongd Aslak Johansene Jason Kohf Joern Ploennigsg Yuvraj Agarwalc Mario Bergésc David Cullerb Rajesh K. Guptaf Mikkel Baun Kjærgaarde Mani Srivastavaa Kamin Whitehouse Bharathan Balaji, Arka Bhattacharyab. Brick:towards a unified metadata schema for buildings. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built*, 27 November 2016.

[17] Per Karlberg David Hälleberg Karl Hammar, Erik Oskar Wallin. The realestatecore ontology. In *International Semantic Web Conference,*, The Semantic Web – ISWC 2019. LNISA,volume 11779, 2019.

[18] Ruben de Laat Pim van den Helm Jakob Beetz, Léon van Berlo. Bimserver.org - an open source ifc model server. In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 1*, BCS-HCI ’08.

Proceedings of the CIP W78 conference, 2010.

[19] N.V. Hoang and S. Törmä. Drumbeat platform—a web of building data implementation with backlinking. In *E eWork and eBusiness in Architecture, Engineering and Construction*. CRC Press, 2017.

[20] A. Wagner M. Shi A. Hoffmann W. Sprenger T.-J. Huyeng, C.-D. Thiele and U. Rüppel. An approach to process geometric and semantic information as open graph-based description using a microservice architecture on the example of structural data. In *EG-ICE 2020 Workshop on Intelligent Computing in Engineering*, 2020.

[21] Jakob Beetz Jeroen Werbrouck, Pieter Pauwels and Erik Mannens. Lbdserver - a federated ecosystem for heterogeneous linked building data. In *E eWork and eBusiness in Architecture, Engineering and Construction*. IOS Press, 2021.

[22] Pieter Pauwels3 Maarten Vergauwen1 Mathias Bonduel, Jyrki Oraskari and Ralf Klein1. The ifc to linked building data converter - current status. In *Proceedings of the 6th Linked Data in Architecture and Construction Workshop*, 2017.

[23] Michael Lagally Toru Kawaguchi Kunihiko Toumura Kazuo Kajimoto Matthias Kovatsch, Ryuichi Matsukura. Web of things (wot) architecture. In *W3C Recommendation*, 2020.

[24] Charles M Eastman Pardis Pishad-Bozorgi Zinghua Gao Shu Tang, Dennis R Sheldon. A review of building information modeling and the internet of things devices integration:present status and future trends. In *Automation in Construction*, BCS-HCI '08, pages 13–22, Swinton, UK, UK, 2019. British Computer Society.

[25] Jianli Hao Ph.D Fangyu Guo Ph.D. Shu Tang Ph.D, Cheng Zhang Ph.D. A framework for bim, bas, and iot data exchange using semantic web technologies. In *Construction Research Congress 2022*, 2022.

[26] Nursi Karoglan Marco Niemeyer Ingo Kunold Hendrik Wohrle Markus Kuller, Fabian Kohlmorgen. Conceptual design of a digital twin based on semantic web technologies in the smart home context. In *IEEE work shop*, CANDO$_EPE2020, pages167--172, Swinton, UK, UK, 2020. British Computer Society.$

[27] Tamer El-Diraby Soroush Sobhkhiz. Developing bim-based linked data digital twin architecture to address a key missing factor: Occupants. In *Construction Research Congress 2022*, 2022.

[28] W. Natephraa and A. Motamedib. Live data visualization of iot sensors using augmented reality (ar) and bim. In *2019 Proceedings of the 36th ISARC, Banff, Canada*.

[29] Yingying Zhang and Jakob Beetz. Building-cps: Cyber-physical system for building environment monitoring. In *CIB W78 - LDAC 2021*, 2021.

[30] Mahmoud Badri Timo Ojalo Matti Pouke, Juho-Pekka Viretanen. Comparison of two workflows for web-based 3d smart home visualization. In *2018 IEEE International Conference on Future IoT Technologies*, 2018.

Propdef Ontology

Propdef ontology is light weight ontology to define the property set of different IFC classes.

(a) :belongToLevel
- isDefinedBy - A part of a building comprising all the rooms that are on the same level.
- subPropertyOf - bot:Element

(b) :hasIFCguid
- isDefinedBy - The IFC specification uses an unique identifier for object instances that follows the universal unique identifier standard UUID with its implementation as a globally unique identifier GUID. The generated GUID is compressed for exchange purpose following a published compression function.
- subPropertyOf - bot:Element

(c) :hasReference
- isDefinedBy - Reference is a relationship between objects in which one object designates, or acts as a means by which to connect to or link to, another object.
- comment -show the reference of every element
- subPropertyOf - bot:Element

(d) :hasThermalTransmittance
- isDefinedBy - Thermal transmittance (U-value) defines the ability of an element of structure to transmit heat under steady-state conditions. It is a measure of the quantity of heat that will flow through unit area in unit time per unit difference in temperature of the individual environments between which the structure intervenes.
- comment - unit of thermal transmittance is W/m^2K
- range - decimal
- subPropertyOf - bot:Element

(e) :hasHeight
- isDefinedBy - The measurement of the distance of an object from the base to the top.
- comment - Measurements in Meters
- range - decimal
- subPropertyOf - bot:Element

(f) :hasWidth
- isDefinedBy - The distance from side to side.
- comment - Measurements are in Meters
- range - decimal
- subPropertyOf - bot:Element

(g) :hasHeadHeight
- isDefinedBy - The position of something when level with the height of a person's head.
- range - decimal
- subPropertyOf - bot:Element
- comment - Measurements of the head height in Meters.

(h) :hasGrossVolume

- isDefinedBy - Total gross volume of the object. Openings, recesses, enclosed objects and projections are not taken into account.
- range - decimal
- subPropertyOf - bot:Element
- comment - Measurements of the head height in Meter cube.

(i) :hasGrossSurfaceArea

- isDefinedBy - Total gross area of the object, normally generated as perimeter * length + 2 * cross section area. It is the sum of OuterSurfaceArea + (2 x CrossSectionArea) and shall only be given, if the OuterSurfaceArea and CrossSectionArea cannot be established separately.
- range - decimal
- subPropertyOf - bot:Element
- comment - Measurements of the head height in Meter square.

(j) :hasRoughness

- isDefinedBy -
  A measure of the vertical deviations of the surface.
- range - decimal
- subPropertyOf - bot:Element

(k) :hasBendRadius

- isDefinedBy - The radius of bending if circular arc or zero if sharp bend.
- range - decimal
- subPropertyOf - bot:Element
- comment - Measurements of the head height in Meter.

Similarly, there are more vocabularies in this light weight ontology.

Click here to see all - `https://gitlab.emse.fr/isi/internships/master-raveendran-nair/-/blob/main/ontology.ttl`

Digital Twin Visualization Features

In this section we explained about the features implemented in the digital twin platform:

Context Menus

We give several command options via context menus to interact with the BIM model.We have two context menus which will popup when we do right-click when the pointer is in the BIM model and the empty space.

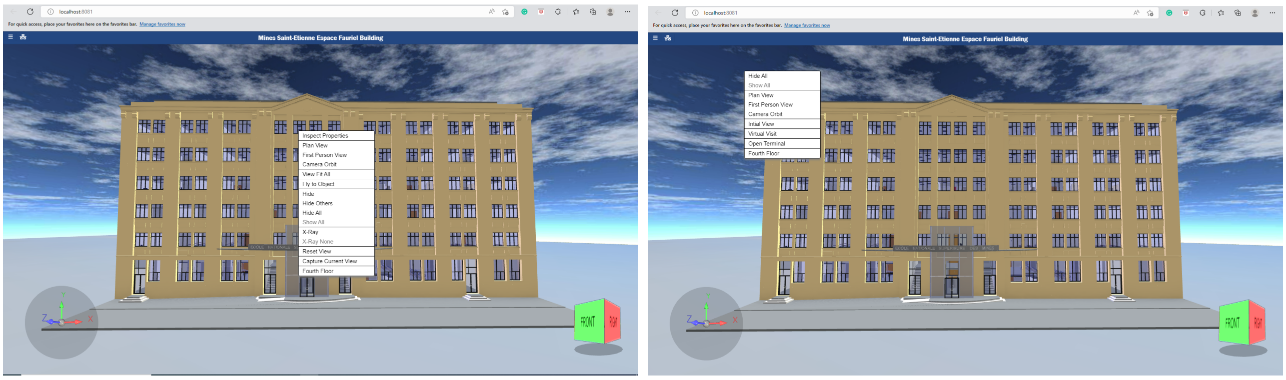

Figure B.1. IFC model of Small Office building

Slice Every Floors

This feature helps to slice and visualise the individual floors.

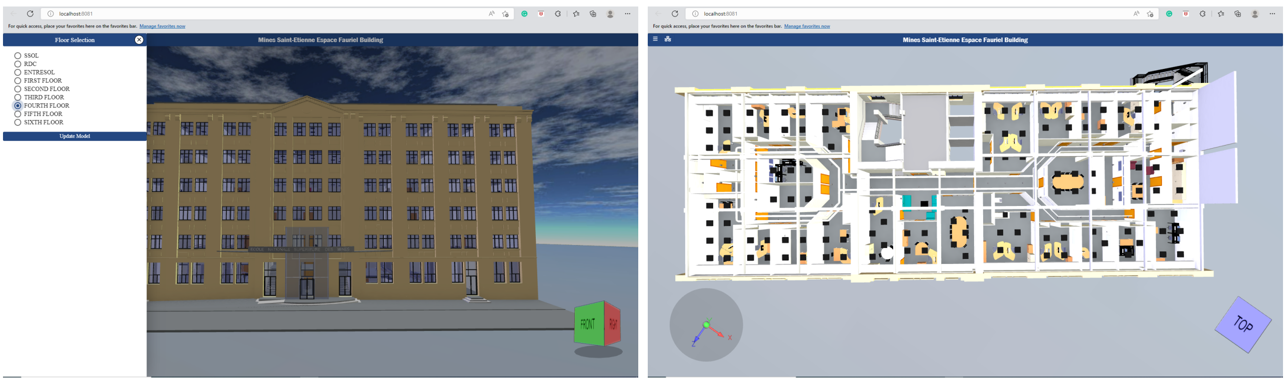

Figure B.2. IFC model of Small Office building

Terminal Box for Digital Twin interaction

A terminal box was provided in the platform to interact with digital model more efficiently.

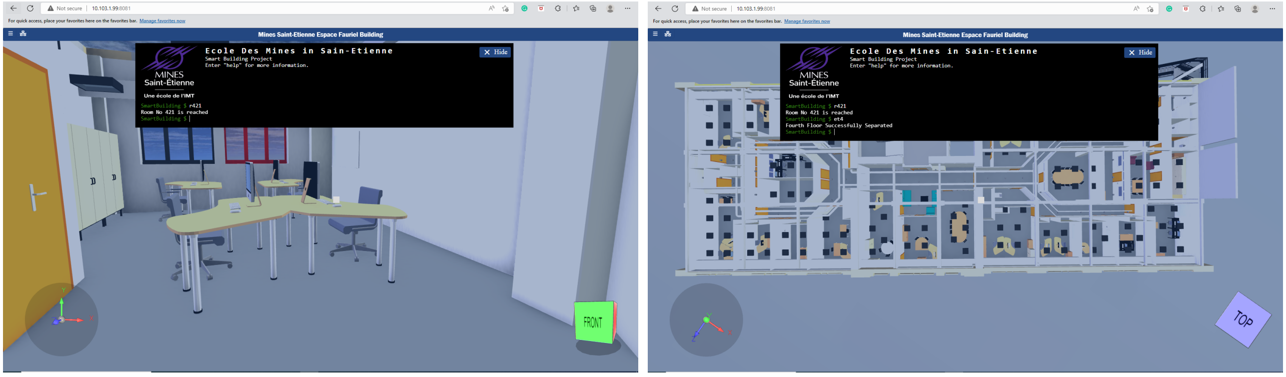

Figure B.3. IFC model of Small Office building

Walk-through Navigation

Walk-through navigation helps the user to visit the building virtually with the help of arrow key in the keyboard.

SPARQL Queries

SPARQL query to delete a thing linked with BIM directory

```
PREFIX dogont: <http://elite.polito.it/ontologies/dogont.owl>
PREFIX saref: <https://saref.etsi.org/core/v3.1.1/>
PREFIX hctl: <https://www.w3.org/2019/wot/hypermedia#>

DELETE DATA
{
  <http://127.0.0.1:3000/IfcWindow/1cTfeCS2jF2wOSCMZ2MxhR> dogont:hasSensor
             "http://127.0.0.1:3000/IfcWindow/1cTfeCS2jF2wOSCMZ2MxhR/Humidity#" .

  <http://127.0.0.1:3000/IfcWindow/1cTfeCS2jF2wOSCMZ2MxhR/Humidity#> saref:hasSensorType "Humidity" ;
    hctl:hasTarget <http://127.0.0.1:3000/IfcWindow/1cTfeCS2jF2wOSCMZ2MxhR/Humidity> .
}
```

Listing C.1. SPARQL query to delete linked thing

Appendix D

Conversion procedure from IFC to XKT

We used a online platform called BIMdata.io to convert the IFC model into XKT.

Procedure for converting IFC to XKT:

(a) Go to https://platform.bimdata.io/

(b) Click add model to upload the selected IFC model

(c) Finally download the converted XKT model from the network tab of the inspect window.

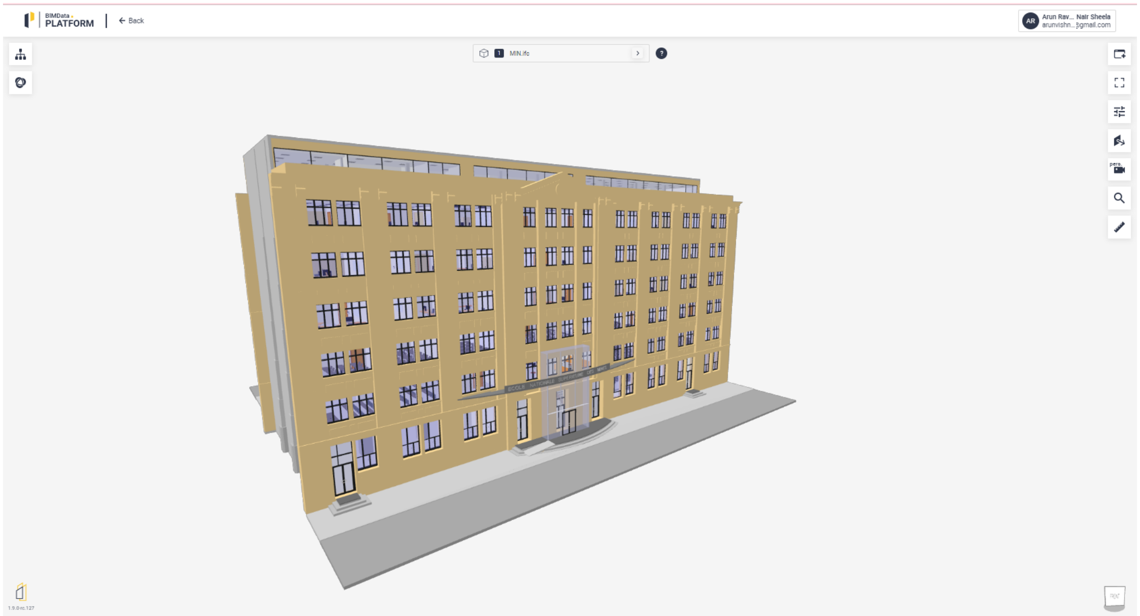

Figure D.1. BIMdata platform

No.	IFC model name	Size of IFC file	Compressed format	Size of compressed format
1	Espace Fauriel Building	214 MB	.xkt	11 MB

Table D.1. Sublayers of IoT and Data Analytics