# A web-based application to integrate Building Management System sensor data and Building Information Model data to support facility management tasks

Angela de Barros Lima
Msc Construction Management and Engineering
2018-2020

## Colophon

| | |
|---|---|
| Thesis title | A web-based application to integrate Building Management System sensor data and Building Information Model data to support facility management tasks |
| Date of final presentation | 22th of September 2020 |



| | |
|---|---|
| Author | Angela de Barros Lima |
| Student number | 1294814 |
| E-mail | angeladebarroslima@hotmail.com / a.de.barros.lima@student.tue.nl |
| University | Eindhoven University of Technology |
| Master Program | Construction Management and Engineering (CME) |
| Chair | Information System in the Built Environment (ISBE) |



### Graduation Committee

| | |
|---|---|
| Chairman | prof.dr.ir. Bauke de Vries<br>Full Professor, Department of the Built Environment, Information Systems in the Built Environment |
| 1st Supervisor | dr.ir. Pieter Pauwels<br>Associate Professor, Department of the Built Environment, Information Systems in the Built Environment |
| 2nd Supervisor | dr.ir. Rianne Appel-Meulenbroek<br>Associate Professor, Department of the Built Environment, Real Estate and Urban Development |
| Advisor | ir. Annemieke Pelt-Thissen |

## Preface

One of my first jobs in the AECO industry was to register building users' maintenance and repair claims for a local real estate developer. Back in those years, BIM was starting to replace traditional CAD files in the design stage but was still a far reality to support the operation and management procedures mostly based on paper plans and excel sheets. This thesis gave me the opportunity to revisit this stage of the building lifecycle and evaluate the state of the art solutions to support facility management tasks. In addition, It was my first contact with software development and coding two things I was always curious about but never had a concrete chance to study.

I would like to thank everyone that somehow contributed to this final report. Professor Pieter Pauwels for guiding me through this process in our bi-weekly meetings. Professor Rianne Appel-Meulenbroek for the literature insights and articles recommendations. Annemieke Pelt-Thissen for providing the Zero Emission Lab documentation and a better understanding of the TU/e facility management department. Nasir Abed for the clarifications about the Atlas Living Lab infrastructure, the orientation to get the Ethical Review Board approval, and for helping me find an alternative solution when the lab activities got suspended.

Moreover, this thesis would not be possible without the support of my parents, my biggest fans and sponsors, and my sister that went back home to take care of them so I could focus on my graduation. Also, my many thanks to Oscar Robles for all the brainstorming sections, and for lending me his computer when my own also decided to take some quarantine time. I also want to mention my CME colleagues in special Siddharth Panjwani, Jianli Zhai, and Haoke Ji. Thank you for the knowledge sharing and academic support in these past two years. Finally, thanks to all my friends from the other side of the Atlantic that gave me some virtual human contact in this very unusual time, I can't wait to see you all in our warm country hopefully under a better political and economic situation.

Angela Lima
Eindhoven, September 2020

## Contents

## Summary

The operation and maintenance (O&M) phase of a building is the longest and most expensive stage in its lifecycle (Sattenini et al., 2011). Hence, interest has grown in applications to support the tasks performed in it leading to the development of information technology (IT) tools specific for this stage as a way to enable both long term strategic decisions and support for daily activities based on real-time data (Carbonari, 2016).

However, the explosion of software dedicated to O&M generates isolated islands of information that are difficult for facility managers to combine, and achieve actual cost savings and user satisfaction (Codinhoto & Kiviniemi, 2014). On the other hand, there is the building information model (BIM) delivered as part of the handover documentation. It is a 3D database with information accumulated throughout the design, engineering, and construction process that can be used to support the O&M processes (Curry, et al., 2013). Because of that, BIM is getting increasing attention as a reference for the information generated by the IT tools for O&M since the building is the common ground for these applications (Codinhoto & Kiviniemi, 2014).

Combining the potentials of BIM data with O&M applications has several benefits for the maintenance and management of facilities. Some examples from the literature are the auto-population of IT tools for facility management, less time spent to find information about a building resource, and improved visualization capability (Pärn et al., 2017). However, some challenges for implementation are reported like modelling problems, uncertainty about the information to be included in the model, and interoperability issues (Pishdad-Bozorgi et al., 2018). Moreover, the Industry Foundation Classes (IFC) is a neutral format and open ISO standard that brings advantages for BIM software interoperability but it is difficult to extend, has a large and complex structure, is not easily queried, and is not web compliant (Bonduel et al., 2018).

Considering the current knowledge gap, this thesis aims to analyse the possibilities of web technologies to create a web app that integrates O&M data and BIM data in a decentralized approach to bridge these islands of information. Inside this main objective, it also evaluates the use of semantic web technologies to make BIM data accessible on the web. Therefore, the main research question of this thesis regards the challenges of implementing this application for a BIM model and a Building Management System (BMS) and the potential it can have to support facility manager tasks in the O&M phase. To answer this question, this study investigates potential use cases for the application and what kind of O&M data could be integrated into it. Moreover, it defines the system requirements for the proposed approach and explores how semantic web and web technologies can be used to implement them. Furthermore, it aims to identify the BIM challenges for this stage of the building lifecycle.

A literature review in the topics of facility management and IT tools for O&M, BIM for O&M, semantic web, and web technologies were performed to achieve these goals. This review was complemented by two study cases, the Zero Emission Lab (ZEL) and the Atlas building, both in the TU/e campus. The potential use cases for the application were based on the facility management tasks from the study cases, and the potential uses for BIM data for O&M found in the literature. In addition, the system requirements were defined and implemented in a proof of concept application that integrates the BIM model of the 8th and 9th floors of the Atlas building with data from a lighting BMS. The infrastructure to access this BMS is provided by the Atlas Living Lab. It was used theoretically to investigate the system requirements to retrieve data from the building sensors since the laboratory was not operational during this study due to Covid-19 preventive measures. Finally, the ZEL architecture and MEP BIM models and their modelling protocol for the design stage were analysed to disclose BIM modelling issues to be improved for the handover model and to complement the literature review in this topic.

The opportunities identified are in the fields of maintenance, Logistics, space management, safety, indoor conditions, and cost management. They require the connection of data from these areas to the building elements like the maintenance records, work orders, asset tracking data, and sensor data from HVAC, Lighting, or alarm systems. This integration in a web application provides a single platform to visualize different datasets and support:

- the evaluation of the building elements' historical condition;
- the location and status of assets in the building;
- the use of building data for spatial analysis, refurbish, and retrofit processes;

- the monitoring of safety thresholds and indoor conditions;
- the investigation of opportunities for resource consumption optimization.

From these potential use cases, the current layout of the application works best to monitor sensors measurements for indoor conditions and energy consumption. It enables the facility manager to retrieve data about the building elements from both the BIM model and the BMS system that could also be consulted during site inspections in smartphones and tablets. The key system requirements for the app are:

- Have a user interface that is independent of the BIM authoring tool (e.g. Revit, ArchiCAD);
- Have an intuitive design that can be adapted for portable devices;
- Display data from the BMS and BIM in a format easy to understand;
- Don’t require technical knowledge to query the database.

To implement these functionalities the Angular framework, and the Chart.js and the Angular Material libraries were used to retrieve, process, and display data taking into consideration a responsive web design scenario. Moreover, the Linked Building Data ontologies (LBD) were used to define the RDF graph of the BIM model, which was stored in a GraphDB web database to enable communication with the app. The resulting application allows its users to navigate back and forward in the building hierarchy and select elements to find information about it. The user interface also has a filter field where the facility manager can type the element name or code to locate it faster. Moreover, the BIM data is displayed as a table with simple property/values pairs and the sensor energy metric report appears as a line chart with different colours for average, maximum, and minimum values.

In the app background, the application programming interface (API) is handling the HTTP requests, using SPARQL to query the triplestore every time an element is selected, processing XML and JSON responses, and synchronizing the sensor data with the building element it refers to. Other system requirements were found and implemented in the literature using web technologies. One is the inclusion of a 2D and 3D viewer for better building navigation, which brings opportunities for research in the level of detail of the 3D geometry and application performance, especially for lower-powered devices (Hu et al., 2018; Mccaffrey et al., 2015) . Another is the representation of O&M data directly in the model using colour schemas to support assessments of indoor quality, and thermal and energy performances, which can be improved by simulation algorithms (Chang et al., 2018; Yan et al., 2013). Other examples are report generation for the O&M data with a customizable timeframe and pop-up alerts for critical thresholds (Mccaffrey et al., 2015; Riaz et al., 2014).

Regarding the BIM data, the semantic web technologies used solved the limitations of the model to be implemented in the app with minors semantic conflicts and data loss. However, the quality of the BIM data will rely on good modelling processes and well-defined requirements with the involvement of the facility management department. Furthermore, the RDF graph generated can be linked to the O&M data to improve access to hard copy or digital files commonly delivered in this stage, although some level of data transformation and digitalization will be required. This brings opportunities for investigation of domain-specific ontologies for O&M and the knowledge generation potential of this linked data database (Kim et al., 2018).

Thus, the developed web application provides a simple user interface to integrate BIM data with BMS data, and the API deals with the basic feature to enable this connection. For this use case scenario, the app could be used to fast retrieve data about the building elements and the sensor readings using the building structure to navigate inside floors and spaces. Information about different sensors could be shown on the same platform to better monitor energy consumption, occupancy status, humidity, and temperature conditions. Moreover, other functionalities and datasets can be included with semantic web and web technologies bringing more research opportunities in the field of BIM for O&M.

## Abstract

The facility managers are responsible for keeping the building conditions and services at a satisfactory level for the facility users while aiming to decrease the operational and maintenance cost. Due to the diversity of data sources necessary to achieve these goals, several IT solutions exist to aid the information management and facility management tasks. Moreover, BIM models are being popularized as part of the handover documentation, which brings possibilities for an object-oriented and semantic approach to deal with building data management and better visualization capability of the building systems. However, these IT tools from different vendors are not easily integrated what creates islands of information, preventing the facility manager from quickly obtaining the overall status of the building. Therefore, this thesis targets the problem of information management to support facility management tasks in a decentralized approach using web technologies to loosely-coupled integrate BIM data with BMS data. This topic was explored through two study cases and the development of a proof of concept application. The evaluation included the BIM requirements to be web compliant using semantic web technologies, the common problems found in the BIM models delivered in the handover stage, the system architecture to enable this integration including the process and display of BIM data and BMS data, and the potential use cases of this approach for the facility managers. The results showed that a web app has the potential to retrieve the information from different O&M data sources, associate them to the building elements using the BIM data as its central reference, and display them in an easily readable way. This can be used to support tasks in the maintenance, logistics, spatial management, safety, indoor conditions, and cost management areas. Although, the current app layout is better developed to support the visualization of indoor conditions and energy consumption sensors. The main system requirements are an easy to use interface, adaptable for portable devices, that don't rely on technical knowledge from the facility manager. Moreover, the system preconditions are that the BIM model needs to be converted to a RDF graph and included in a triplestore, and that the BMS handles the sensor data storage and its API allows third-party access to this information.

---

**Keywords**
BIM for O&M, BIM and BMS integration, Web technologies for Facility management, Semantic web technologies for BIM.

## List of Abbreviations

| | |
|---|---|
| API | Application programming interface |
| AECO | Architecture, engineering, construction, and operation |
| BEMS | Building Energy Management System |
| BIM | Building Information Model |
| BMS | Building Management System |
| BOT | Building Topology Ontology |
| BPO | Building Product Ontology |
| CAD | Computer-aided Design |
| CAFM | Computer-aided Facility Management |
| CCMS | Computer Control and Monitoring Systems |
| Chrome DevTools | Google Chrome Development tools |
| CMMS | Computerized Maintenance Management Systems |
| COBie | Construction Operations Building Information Exchange |
| CSS | Cascading Style Sheets |
| CSV | Comma-Separated Values |
| DALI | Digitally Addressable Lighting Interface |
| DOM | Document Object Model |
| EMS | Environmental Management System |
| EAM | Enterprise Asset Management |
| FM | Facility Management |
| GUID | Globally unique identifier |
| GIS | Geographic Information System |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transfer Protocol |
| IEQ | Indoor Environmental Quality |
| IFC | Industry Foundation Classes |
| IFMA | International Facility Management Association |
| IT | Information Technology |
| IWMS | Integrated Workplace Management System |
| JSON | JavaScript Object Notation |
| LBD | Linked Building Data |
| MEP | Mechanical, Electrical, plumbing |
| MVD | Model View Definitions |
| O&M | Operation and maintenance |
| OPM | Ontology for Property Management |
| OWL | Web Ontology Language |
| PRODUCT | Building Product Ontology |
| PROPS | Building Properties Ontology |
| RDF | Resource Description Framework |
| RDF4J | Java framework for working with RDF data |
| REALL | Research Environment Atlas Living Lab |
| REST | Representational State Transfer |
| RFID | Radio-frequency identification |
| SQL | Structured Query Language |
| SPARQL | SPARQL Protocol and RDF Query Language |
| STEP | Standard for the Exchange of Product Model Data |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| W3C | World Wide Web Consortium |
| W3C LBD-CG | W3C Linked Building Data Community Group |
| XML | Extensive Markup Language |
| XSD | XML Schema Definition |

## List of figures

## List of tables

# 1 Introduction

This chapter introduces the problem assessed in this study. Considering the presented context, it describes the goals of the thesis and the research questions it aims to answer. Furthermore, it includes the research design that explains the strategy to achieve these targets, which is complemented by the reading guide, to describe how this study is documented.

## 1.1 Problem definition

The international facility management association (IFMA) describes facility management as “a profession that encompasses multiple disciplines to ensure functionality, comfort, safety, and efficiency of the built environment by integrating people, place, process, and technology” (IFMA, n.d.). Khodeir (2008) complements this definition stating that, as a multidisciplinary professional, the facility manager needs knowledge among others in engineering, architecture, accounting, finance, management, and behavioral science. In this context, technology has risen in importance to gather information about the building conditions, its users, and its daily operation, and was included in the definition of facility management in 2003 with the growth of IT tools for this industry segment (FMLink, 2003).

This technological advancement generated an expectation for the future of the facility management, for some called e-FM, where labor-intense functions and repetitive processes, as collecting and processing data, are solved by technology (Smith, 2011). On one hand, the facility managers spend less time searching for the information, and one the other, they have access to new data sources that allow them to make strategic long-term decisions as well as improve day-to-day operations. The IT tools for operation and management (O&M) are commonly classified in computer-aided facility management (CAFM) software, computerized maintenance management systems (CMMS), and building management systems (BMS).

A CFAM application stores and manages the building data inputted in the software like CAD files, inventory, manuals, and warranties (Alileche, 2018). CMMS is a type of CAFM specialized in maintenance tasks such as work order management, maintenance-related cost, and preventive maintenance plans (Alileche, 2018). BMS is a system that controls mechanical and electrical services, the combination of sensors and internet access allows the facility manager to monitor and manage the system with commands in a computer and to get data analytics about them (Wong et al., 2018). Hence, these systems and applications support specific facility management tasks such as managing the repair and replacement execution and planning preventive maintenance schedules, organizing the supplies logistic stream, monitoring building indoor conditions and coordinating improvements in the facility spaces, allocate people and activities, make risk assessments for system failure and investigate options to optimize resource consumption.

In parallel with these IT tools for facility management, another important development is the building information model (BIM) introduced as part of the handover documentation. It has the potential to become a structured 3D database full of information accumulated throughout the design, engineering, and construction process that can be used to support the O&M phase (Curry, et al., 2013). Traditionally, the data about these stages is delivered in hard copies or unstructured digital files (Hu et al., 2018). That makes it difficult for facility managers to find and utilise essential information such as up to date as-built plans, product data sheets, operation manuals, and safety instructions (Wong, et al., 2018). Moreover, a BIM model enables the visualization of the building systems in 3D.

IT tools for O&M and BIM complement each other. If BIM software can structure information in an object-oriented style and give visual geometric information about the facility, FM-oriented IT tools can better handle the O&M specific tasks they are designed to perform. However, more data demands better information management to not create islands of information that makes it harder for facility managers to transform data into actual knowledge (Smith, 2011). In this context, BIM data is rising as an option to associate the information generated by CAFM, CMMS, and BMS to the building elements (Codinhoto & Kiviniemi, 2014). In addition, Dibley (2011) points out that the potential of BIM to increase productivity and efficiency in O&M is related to its capacity to be connected to real-time data collection systems (e.g. wireless sensor systems) in an automatized way. This is complemented by Parn et al. (2017) who state that when BIM is not fully exploited for decision support in O&M the opportunity to enhance the building performance with the rich semantic data in it is lost. Edirisinghe et al. (2017) cites as application area of BIM-enabled facility management, among others, asset tracking, indoor navigation, lifecycle assessment, renovation planning and execution, and space management.

Nonetheless, this integration has several barriers. O&M applications are developed in different proprietary software, with incompatible communication protocols or hardware, and distinct data formats (Smith, 2011). Furthermore, the integration of BIM data with IT tools is hard to achieve as the IFC file format, created to exchange BIM data between applications in the AECO industry, is not web-compliant and cannot be reached by these facility management tools (Shen, et al., 2010). Other challenges for the adoption of BIM data in this stage is the lack of clear requirements for the BIM model early in the project development what results in the data from the model being incomplete in the handover stage, with no connection between its assembly and O&M processes and elements with wrong semantic relationships (Codinhoto & Kiviniemi, 2014).

Therefore, although there is potential to move toward an e-FM and BIM-enabled O&M, more convincing results and research are needed to transposes the integration barriers (Edirisinghe, et al., 2017). One approach for the lack of compatibility is the possibility to integrate these data sources in a loosely-coupled way by using web technologies. This solution aims to comply with the fragmented nature of the industry by not relying on only one repository for all project information but rather take advantage of the individual strength of the applications and at the same time overcome its isolation (Shen, Hao, & Xue, 2012). In this study, this decentralized approach was implemented as an intermediate web application to display the data from one BMS together with the information from a BIM model without changing the underlying architecture of both data sources. This method has some advantages for O&M such as the scalability of the solution, since theoretically it can be expanded with the inclusion of more applications, and portability, since a fully deployed web application could be used in any device with access to an internet browser.

Moreover, regarding the access of BIM data on the web, semantic web technologies are getting increasing attention to elevate the IFC format to a web compliant state. One of the approaches is to use the Resource Description Framework (RDF[1]) standard to generate a RDF graph that enables data linking and sharing on the web (Curry, et al., 2013). IfcOWL[2] is a standard the provides Web Ontology Language (OWL) representation for the IFC EXPRESS schema (BuildingSMART, n.d.-a). However, it aims to mirror the IFC EXPRESS schema fully in an OWL ontology resulting in a large and complex RDF file which makes it difficult to retrieve information from it, since query writing becomes longer and more intricate (Bonduel, et al., 2018). Thus, in this study, the Linked Building Data ontologies were applied as they cover the problems of the ifcOWL and have important developments in converting tools, such as the IFCtoLBD converter [3] and Revit-bot exporter [4].

Considering the current knowledge gap, this master thesis aims to contribute to the existing research under a decentralized combination of BIM data and O&M applications to support facility management tasks. This includes an investigation of the common BIM modelling problems for facility management, and its repercussions to a possible integration with other data sources, as well as its conversion to RDF, with the LBD ontologies. Moreover, it comprises the development of a proof of concept web application to connect this RDF graph with data from a BMS. This aims to assess the process of retrieving data, processing it, and displaying it to the end user under the scope of a use case scenario.

## 1.2 Research goals and questions

This research aims at analysing whether a web-based application, which visually integrates the information generated by a building management system with the data from a BIM model, has the potential to be a strategic decision support tool for the O&M tasks. The end user of this application is the facility manager and the task target are the location of assets and defective building elements and systems, support for planning punctual interventions, refurbishment and retrofit process in the facility, analyses of indoor environmental conditions and user occupancy patterns, critical thresholds monitoring, and simulations for better management of resources and building users.

[1] https://www.w3.org/RDF/
[2] https://www.w3.org/OWL/
[3] https://github.com/jyrkioraskari/IFCtoLBD
[4] https://github.com/MadsHolten/revit-bot-exporter

Thus, this thesis explores the topic of BIM and IT tools for facility management under the context of web technology. Hence, not only the deficiencies and potentialities of BIM to support the O&M tasks need to be assessed but also how this structured database can be used in a web application. For the O&M IT tools, it means understanding how a web server can request and retrieve data from them and how it should process this data to be useful for the end-user.

With this scope, the main research question of this study can be defined as:

- What are the challenges and potentialities of implementing a web-based application with building data from a BIM model and a BMS to support tasks in the O&M phase?

And the related sub-questions for this research question are:

- What are the potential use cases for the application and what kind of O&M data is needed to enable them?
- What functionalities the system architecture should enable to support O&M tasks and how they can be achieved?
- What challenges BIM data need to overcome to support these use cases?

## 1.3 Research design

This thesis was structured in three main stages (Figure 1). The first stage comprises the literature review and the research about the two study cases. The scope of the literature review is associated with the research question and sub-question, namely, IT tools for O&M, BIM-enabled facility management, linked data approach for building data, and data decentralized integration using web technologies.

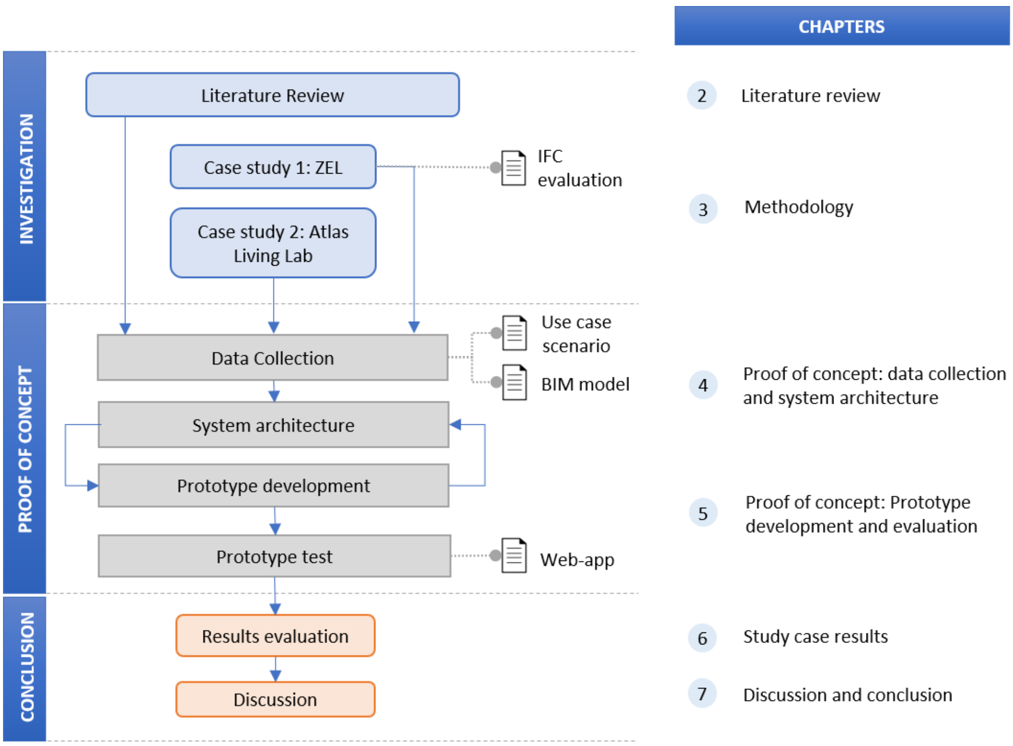


*Figure 1. Research design*

The first study case is the Zero Emission Lab (ZEL), a laboratory in the TU/e campus. It complements the literature review in the investigation of the common problems of using BIM models made in the design phase in the O&M stage by providing a real BIM model and related documentation. The second study case is the Atlas Living Lab, an infrastructure to support researches about the use of the lighting system of the Atlas building, also on the

TU/e campus. It gives access to building management system data that enable the development of the proof of concept.

The development of the prototype application as a proof of concept aims at a practical approach to the semantic web and web technologies. It follows a typical procedure for app development which includes data collection, the definition of the system architecture, prototype development, and testing. In the data collection, the insights from the literature review and the two study cases were used to create a BIM model for the $8^{th}$ and $9^{th}$ floors of Atlas, which was used as the second data source in the developed application, and the use case scenario, that defined the scope of the basic system architecture.

The conclusion stage of this thesis comprises the evaluation of the results obtained which complements the investigation phase to answer the research questions and sub-questions. It culminates in a final discussion about implementation areas for the application, including suggestions for the prototype improvement and future research opportunities in the subject.

## 1.4 Reading guide

This thesis is organized in seven chapters, the first being this introduction to the knowledge gap of BIM data for O&M and its lack of integration with the IT tools used in this phase and how this thesis aims to contribute to it. The second chapter is the literature review. It comprises the gathering of information to base the answer to the research question and sub-questions. This includes the potential use case for BIM data in the O&M considering the proposed approach, the BIM challenges for this stage, and the common system requirements to support facility management tasks.

The methodology of this study is divided into three chapters. The first brings an overview of the applied methodology and its justification, and the exploration of the two study cases. The second further developed the data collection and the system architecture stages in the proof of concept application. The third one is focused on the development stage, namely the data gathering, processing, and displaying. It also explains how the application was alfa tested.

The sixth chapter comprises the critical evaluation of the results obtained. It deliberates about the positive aspects for facility managers, the limitations of the application, and the challenges to achieving a fully deployed web solution. This chapter is followed by the discussion and conclusion chapter which finalizes this thesis with a reflection about the research questions and sub-questions, future possibilities, and research limitations.

# 2 Literature review

This literature review chapter aims at describing key concepts for the topics covered by this master thesis to enable the discussion proposed in the research questions. It gives a broad view of the facility manager role and the new demands for the operation and management of facilities, as well as an overview of the IT tools developed to support this industry's needs. Moreover, the use of BIM models and processes to support facility management tasks are explored regarding the potentials and challenges reported in the literature and complemented by a non-exhaustive examination of research approaches to implement these potentials and overcome these barriers. Furthermore, since the prototype applies semantic web and web technologies elements, this chapter presents the basic concepts in these topics and how they were used in this thesis.

## 2.1 Facility Management

As mentioned in the introduction chapter of this thesis, facility management is a multi- and interdisciplinary activity. The O&M processes should assure a sustainable strategy for the management of the building infrastructure and services along with providing a qualitative environment for the human activities (Parn, et al., 2017). Sustainability is one of the key aspects of facility management since the O&M phase is the longest in the building life cycle. It ends up being the most expensive and the one with most waste of resources, like water, and energy (Hu, et al., 2018). Facility management can be associated with the generation of revenue, in this case, the facility manager is also responsible to maximize the profit by providing the best experience to the building users and the best environment for them to develop their activities (e.g. office buildings, universities, hospitals).

In this context, the scope of facility management is growing throughout the years to include more integrated services, long-term contracts, and strategic activities (Talamo & Bonanomi, 2016). This reflects the paradigm shift of the industry from a focus in merely cost reduction to a gradual focus in strategic facility management that adds value to the core business, corporate real estate management, and business-to-business marketing (Jensen, et al., 2012). Jensen (2010) in his facility management value map emphasizes the different ways in which facility managers can create and add value beyond cost reduction with effects in the main businesses, such as user satisfaction, productivity, reliability, adaptation and company culture, as well as in related business areas, like economic, social, spatial and environmental areas. The facilities, activities, real estate, technology, manpower, and the knowhow are inputs for facility managers that enable planning, coordinating, controlling, and improving processes to deliver products, space, services, development, and relations (Jensen, 2010).

Consequently, the organization model increases in complexity, including new areas of interest and disciplines, and demands the creation of new roles with new competencies and support tools to comprise with the expectations of improved O&M processes (Talamo & Bonanomi, 2016). Rondeua et al. (2006) present the jobs' responsibilities of the facility managers in nine major functional areas according to the IFMA definition of the facility management tasks:

- Long-range facility planning
- Annual facility planning (tactical planning)
- Facility financial forecasting and management
- Real estate acquisition and/or disposal
- Interior space planning, work specification, and installation and space management
- Architectural and engineering planning and design
- New construction and/or renovation work
- Maintenance and operations of the physical plant
- Telecommunications integration, security, and general administrative services (food services, records management reprographics, transportation, mail services, etc.)

Which are compatible with the basic categories of demand for facility managers defined in the EN 15221-1:2007 and EN115221-4:2011 standards for facility management described by Talamo & Bonanomi (2016) in two main domains:

1) Space and infrastructure
- Accommodation (space)

- Workplace (working environment)
- Technical infrastructure (utilities)
- Cleaning (hygiene and cleanliness)
- Outdoor (land, site, lot parking)

2) People and organization
- Health, safety, and security
- Hospitality (supports for hospitable working environments)
- Information and communication technologies

In this thesis, the Atlas building, an office and educational building, and the Zero Emission Lab, a combustive engine laboratory, were used as study cases to define the main areas of interest of the facility managers for the application and their associated tasks that were grouped according to Table 1.

*Table 1. Facility managers areas of expertise and related tasks*

| **Areas** | **Subareas** | **Related tasks** |
|---|---|---|
| Maintenance | Garden and indoor plants watering<br>General repair and replacement<br>Indoor and outdoor cleaning<br>Waste management (e.g. recyclable, chemical and biomedical) | • Register users complains<br>• Assign personnel to solve maintenance issues<br>• Make preventive maintenance plans<br>• Identify recurring issues and perform root cause analysis<br>• Monitor cleaning schedules<br>• Ensure the correct destination of waste |
| logistics | Office equipment supplies (e.g. printers, coffee and snack machines)<br>Special supplies (e.g. laboratory supplies)<br>Asset tracking | • Schedule order supplies<br>• Check if the order is correct<br>• Perform product quality checks<br>• Coordinate tender procedures<br>• Provide information about an asset's location<br>• Plan the storage of assets that are not current in use |
| Space management | Workspace and meeting room allocation<br>Parking space allocation<br>Renovation and refurbishment | • Verify if the users have an available space to perform their activities in the correct time<br>• Monitor the increase or decrease in demand for space<br>• Evaluate if the existing condition of the spaces support the activities developed in it<br>• Analise how renovation and refurbishment activities impact the building users |
| Safety | Fire safety<br>Access control<br>Closed-circuit television (CCTV) monitoring<br>Asset protection<br>Risk management | • Analysis of the operational safety of the building.<br>• Make probabilistic risk assessments<br>• Evaluate potential failures of the building systems<br>• Monitor who has access to the building |

| | | |
|---|---|---|
| | | • Define extra safety measures for valuable assets |
| Indoor conditions | Indoor environment quality<br>User satisfaction and performance | • Verify the compliance of legal standards (e.g. Dutch working conditions legislation, WELL building standard)<br>• Monitor and adjust the ventilation, lighting, heating, and humidity systems |
| Cost management | Resource optimization<br>Acquisition cost optimization | • Monitor the consumption of natural resources<br>• Define long-term plans to optimize operational costs<br>• Make strategic purchasing decision to minimize cost |

### 2.1.1 IT tools for facility management

The support tools for facility management evolved in time and gained more functions with the advancement of information technology (IT). IT involves the use of computer-based information systems to collect, store, process, and transmit information (Sooriyarachchi & Karunasena, 2010).

There are several classifications for the facility management IT tools. Some considerer CAFM as an umbrella term to all applications that use IT software to manage building data collections and classify other applications as types of CAFM that handle specific data sources (e.g. environmental management systems (EMS) and Computer control and monitoring systems (CCMS) (Smith, 2011; Parn, et al. 2017). Some consider CMMS and CFAM as two IT solution branches, as the first computer-based information systems used were related to maintenance tasks (Aziz et al., 2016). Some see it as a hierarchical chain that starts with CAFM and as the technology evolves aggregates more tasks and change category to CMMS, then enterprise asset management (EAM), until an integrated workplace management system (IWMS) (Mohanta & Das, 2015; Figure 2).

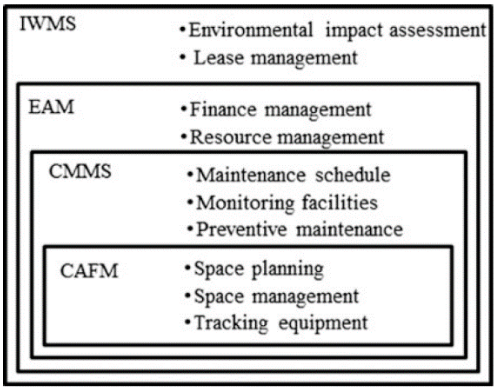



*Figure 2. IT tools for FM based on level of features (Mohanta & Das, 2015)*

Whatever the classification approach is, it seems to have a clear distinction between this first generation of IT tools and BMS. These Building Management Systems, also called Building Automation Systems, go beyond the management of data collections to enable the control and monitoring of the building system through an IT tool in real-time (Wong, et al., 2018). Common building services associated with BMS are power, heating, ventilation, air-conditioning, access controls, pumping stations, elevators, and lights (Alileche, 2018). Building Energy Management Systems (BEMS) are often classified as a subset of BMS that are specific to energy systems. BEMS gained popularity as a way to promote energy-efficient buildings, and to better understand the gap between the predicted versus actual values of energy consumption (Ock et al., 2016).

Making a parallel with the tasks described in the previous topic, there are clear benefits in applying this IT tool to support the operation and maintenance of facilities. For example, CMMS provides a platform to register and organize maintenance procedures and work orders. With the correct information about the problem and the systems affected, the involved costs, and the time to solve them, it becomes a powerful database to perform critical analysis of potential failures of the building systems and its repercussion in cost and people safety. Maintenance based in risk assessments is very important when handling toxic or hazardous material, since failure in the systems for producing or transporting these substances, or in its monitoring sensors, can have severe consequences (Arunraj & Maiti, 2007). Moreover, modern BMS systems also provide predictive maintenance capability, they can inform future demands for repair and replacement based on the lifespan of its components and in their degraded state due to fluctuations in use or weather conditions, preventing disruption in the service (Bumblauskas et al., 2017). This also has repercussions for cost management, since the future demand can be included in the bidding process to bargain better prices.

CAFM tools for space management can generate analytics about building occupancy what is valuable information for the long-term strategy for real state disposal or acquisition (Lavy, 2008). These tools can be combined with Radio-Frequency Identification (RFID) tags placed in the university assets to monitor in real-time their location and status (Wong, et al., 2018). The valuable asset's location can be used to map strategic areas that can be protected by sophisticated access systems that use a role-based, attribute-based, or a capability-based approach to give only authorized people access to these sensitive areas (Zhang et al., 2019). The CCTV and alarm systems can use sensors to detect intruders or dangerous situations, like heat increase and smoke, store pictures and videos of the scene, and send alerts to authorities.

Also, BMS and BEMS use sensors to monitor in real-time room conditions providing data for indoor environment quality assessments (Kallio et al., 2020). Several of these systems come with options for user personalization, this generates analytics to understand better user behavior and preferences (Kallio et al., 2020). Furthermore, the sensors in smart HVAC systems enable automation capabilities to optimize energy consumption by enabling modulating control of the building temperature, automatically turning off the system out of business hours, or preheating or pre-cooling rooms before occupation (Durier, 2017). Other examples of resource optimization are responsive façades that adapt according to the daylight level to maximize user comfort and balance energy costs (Hosseini, et al., 2019), energy systems that optimize the use of the energy generated by the building (e.g. Photovoltaic generation and energy storage) back in its operating system or the electric grid using learning algorithms (Di Santo et al., 2018), and identification of heat loss problems (Yan et al., 2013).

However, The existence of so many specialized applications with different classification systems for the building elements, together with hardy copy data and unstructured digital files makes it difficult for the facility manager to monitor the overall status of the building and correlate information (Peng et al., 2017). Another issue is that most of the supporting tools for facility management depend on some level of manual input of data, and hence are error-prone and time-consuming (Peng, et al., 2017). Even BMS needs that the user implements a project with the building spatial structure in the software so that it can relate the sensors in the real world to the digital representation of the building.

Moreover, commonly the IT tools for facility management have poor geometry visualization capabilities, generally a 2D visualization of a CAD file or a JPEG drawing (Aziz, et al., 2016). This prevents facility managers to be able to define the exact location of an issue (Aziz, et al., 2016), and to be better prepared for the site conditions when executing a maintenance service (Liu & Issa, 2016). Both can represent a significant waste of time. This context opens opportunities to use BIM data to support O&M processes, the next topic explores what are these opportunities and the challenges associated with them.

## 2.2 Towards a BIM enabled FM

The acronym BIM is commonly associated with Building Information Model or Building Information Modelling. The BIM model refers to the digital representation of the building with information depth, and BIM modelling refers not only to the stage of creating the model but also the processes of maintaining, using and exchanging it throughout the building lifecycle (Borrmann et al., 2015). Because of this expansion of the BIM modelling concept, BIM is also associated with Building Information Management. BIM management focuses on the capacity of BIM to be a source of knowledge and innovation diffusion in the project/company level as well as in the AECO industry level (Lindgren & Wide, 2018).

A BIM model is object-oriented. The components of the building have embedded graphic and data attributes and are controlled by parametric rules that allow them to be manipulated consistently through the project database (Sacks et al., 2018). A BIM model is also semantic. The data of a BIM model follows the building hierarchical structure containing the relationship between the building components and other physical or logical entities, for instance, spaces and zones (Borrmann, et al., 2015). BIM as a modelling technology is used to produce, communicate and analyze accurate building models, improving the quality and productivity of traditional manual or CAD-based processes (Sacks, et al., 2018). Because of these characteristics, BIM is in the core of solutions to improve inter- and intraorganizational collaboration in the construction industry (Ghaffarianhoseini, et al., 2017).

However, due to the fragmented nature of this industry, a key aspect for this collaboration, and hence information exchange and knowledge generation, is interoperability. It refers to the capability of different systems (hardware or software) to work together (Epstein, 2012). Ideally, this collaboration should be based on open standards to support data flowing between applications and people without losing information integrity (Epstein, 2012). Although full interoperability can never be achieved, the most consolidated effort in this direction is the IFC schema. The Industry Foundation Classes schema is a standardized and vendor-neutral digital description of the built environment (BuildingSMART, n.d.-b). It is developed by BuildingSMART together with other standards to facilitate IFC BIM workflows. Those other standards are the Model View Definition[5] (MVD), a subset of IFC tailor to a specific process, the BIM Collaboration Format[6], to exchange information about issues in the project, and the Information Delivery Manual[7], that document the BIM process and the information exchanged in them. The most recent release of IFC is IFC4.1[8].

Although the IFC format represented an important step to allow the exchange of data between design and construction management applications, it has several flaws to expand BIM processes to other stages of the building lifecycle. The next topic introduce present barriers and potentials expected of BIM data to be implemented in this field.

### 2.2.1 Current status of BIM for O&M

The literature review on the topic of BIM for O&M aimed at defining the current status of BIM data quality and usability for facility management. 14 documents were analyzed, six literature reviews from 2011, 2014, 2016, 2017, and 2019, three questionnaire surveys with industry practitioners from 2016 and 2019, and five case studies from 2013, 2014, 2017, and 2018. The main topics and subtopics regarding potentials for BIM in facility management are depicted in Table 2.

The articles were obtained by searching the TU/e library database with the keywords *BIM* combined with *facility management*, *operation and maintenance*, and *BIM 6D* with a time frame starting from 2010. The results were narrowed down after evaluating the articles' abstracts with a focus on studies that analyzed the potential of BIM to support the FM process and discussed the challenges associated with them. Since these potentials and challenges overlap in several of them, the selected articles were chosen to represent the universe of studies in the topic. The literature reviews were selected based on the time of publication, with an emphasis in the last five years, and the number of articles reviewed. Moreover, surveys and case studies complemented them, bringing areas that were superficially addressed by the literature reviews such as BIM implementation in existing buildings, impact of different levels of model accuracy for decision making, and integration with web tools for O&M.

[5] https://technical.buildingsmart.org/standards/ifc/mvd/
[6] https://technical.buildingsmart.org/standards/bcf/
[7] https://technical.buildingsmart.org/standards/information-delivery-manual/
[8] https://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML/

*Table 2. BIM potential for FM in literature review*

| Topics | | Subtopics | Authors [a] |
|---|---|---|---|
| 1) | Improvement in manual process for information hand over | Auto population of IT tools and Improvement in data accuracy | [1], [3], [4], [5], [8], [9] |
| | | Increase in speed for data retrieval | [1], [2], [3], [4], [5], [6], [7], [8], [9], [14] |
| | | Better organization and geometric information of the building plans | [1], [2], [3], [4], [5], [8], [9], [13] |
| 2) | Visualization of 3D data | Indoor navigation | [2], [4], [5], [6], [8], [9], [11], [14] |
| | | Fast location of building components and areas | [1], [2], [4], [5], [6], [8], [9], [11], [14] |
| | | Marketing and satisfaction survey | [5], [6], [8] |
| | | Simulations | [7], [8], [9], [11] |
| 3) | Maintenance | Efficiency in work orders execution and maintenance planning | [1], [2], [4], [5], [8], [14] |
| | | Checking design maintainability | [1], [2], [4], [5], [6], [7], [8], [9], [10], [12] |
| 4) | Asset tracking | Mobile location of building resources with RFID integration | [5], [7], [8], [9] |
| 5) | Integrated data environment | real-time data access and sensor integration | [5], [6], [8], [9], [11] |
| | | Controlling and monitoring building systems | [6], [8], [9] |
| | | Aid for decision making | [1], [3], [5], [8], [9], [11], [13] |
| 6) | Sustainability assessment | Lifecycle assessment and cost | [5], [9], [10], [11] |
| | | Carbon foot-printing | [5], [10], [11] |
| | | Potential recycling rate | [5], [10], [11] |
| | | Planning for disassembly of components for reuse | [11] |
| 7) | Space management | Planning and feasibility studies for space use | [5], [6], [7], [8], [11] |
| | | Emergency and safety planning | [5], [6], [7], [8], [11], [14] |
| 8) | Renovation and retrofit | Renovation and retrofit planning and cost-saving | [5], [7], [8], [11] |
| | | Deconstruction planning | [11] |

[a] [1] Parn et al., 2017; [2] Liu & Issa, 2016; [3] Pishdad-Bozorgi et al., 2018; [4] Kelly et al., 2013; [5] Edirisinghe et al., 2017; [6] Carbonari, 2016; [7] Nical & Wodynski, 2016; [8] Matarneh et al., 2019; [9] Codinhoto & Kiviniemi, 2014; [10] Ashworth et al., 2019; [11] Volk, et al., 2014; [12] Leite, et al., 2011; [13] Peng, et al., 2017; [14] Rasys, et al., 2014.

From this analysis, it was possible to conclude that the potential attributed to BIM derivates from the 3D visualization of the building geometry and the possibility to input data in the object level, and to extract this data in a structured way using BIM tools (e.g. list of components, quantities, floor area).

The popularization of BIM models and processes for design and construction enables the encapsulation of rich semantic data in the model early in the building life cycle (Parn, et al., 2017). Hence, it can be delivered as part of the handover documentation and prevents extra efforts to collect and organize a profusion of useful data generated during the production process (Liu & Issa, 2016). Besides, associate O&M data with the geometrical data retrievable from a BIM model reduces the time spending on finding relevant information about equipment and building materials (Parn, et al., 2017). This data can be used to fast populate the IT tools for O&M with building data, which saves precious time for the facility managers and elevate the accuracy level of this information (Pishdad-Bozorgi et al., 2018).

Moreover, a BIM model provides the visualization of the relationships among the building components as ‘is contained in’, ‘is adjacent to’ and ‘is part of’ in an accurate geometrical context (Kelly et al., 2013). The visualization benefits of BIM models assist the O&M personnel in several ways as in the fast location of building components and rooms (Edirisinghe, et al., 2017). Carbonari (2016) also points out the advantages of the 3D visualization for market intelligence and satisfaction survey.

In the field of maintenance, these characteristics are expected to enable accurate defect detection and report (Edirisinghe, et al., 2017), in opposition to the traditional process of using a paper-based facility map and subjective judgment based on experience (Nical & Wodynski, 2016). A digital twin of a facility enables facility managers to precisely locate maintenance work orders and to identify in advance the working conditions and the systems affected, saving time and money during the execution (Kelly, et al., 2013). Moreover, a BIM model can be used to identify maintainability problems in the design phase making them easier and cheaper to solve (Liu & Issa, 2016).

Asset tracking is another field optimized by BIM with the integration of real-time resource location technologies, namely RFID (Nical & Wodynski, 2016). Asset management depends on reliable information on the asset inventory, condition, and performance and can benefit a lot by the visualization power of BIM data (Parn, et al., 2017). Furthermore, Matarneh et al. (2019) mention the potential of BIM data to be a central reference for various data sources in the O&M stage. They indicate several approaches for BIM data and O&M data integration such as agent-based web services, linking data in the cloud, and relational databases (Matarneh, et al., 2019)

Some emphasis is given in BIM for sustainability as a better way to get precise information about the materials to calculate the life cycle costs, to plan the implementation of green buildings and analyse carbon reduction (Ashworth et al., 2019). BIM models can be used to more accurately calculate recycling rates, analyse the disassembly of components connections, verify the separability of the material’s layers and composites and detail the pollutant emissions calculations (Volk et al., 2014). Additionally, BIM-based systems enable complex energy evaluations and simulations (Nical & Wodynski, 2016).

Moreover, space management, renovation, and retrofit are areas considered improved by BIM processes. BIM brings opportunities for automated space management in 3D (Pishdad-Bozorgi, et al., 2018). This is particularly important to model emergency scenarios and safety operations processes and can be used to train people to prepare for these circumstances (Edirisinghe, et al., 2017). Similarly, BIM models aid decision making in refurbishment processes for reduced costs, safer sites, and enhanced collaboration (Volk, et al., 2014). Other uses are assessing maintenance impacts, scenario planning, and execution planning (Edirisinghe, et al., 2017). For deconstruction, BIM can be used to plan and to evaluate vulnerabilities in the construction and make collapses analyses (Volk, et al., 2014).

However, each potential is associated with challenges for its full implementation. Table 3 brings an overview of the barriers reported in this investigation. There are other challenges beyond the ones discussed here, namely internal factors (e.g. organization resistance, lack of personnel able to use BIM) and external factors (e.g. social, economic, legislative and regulative, local culture and market context) (Parn, et al., 2017), that were not included as they are outside the scope of this thesis. Thus, the focus is on the BIM data available and its integration with IT tools for facility management.

*Table 3. BIM challenges for FM in literature review*

| Topic | Subtopics | Authors [a] |
|---|---|---|
| 1) Modelling problems | Model incompletion at handover stage | [3], [9], [11] |
| | Lack of clarity regarding model assembly and its relationship with O&M processes | [1], [2], [4], [5], [7], [9], [10], [11] |
| | High effort to model existing buildings | [5], [6], [11] |

| | | | |
|---|---|---|---|
| | | Inadequate LOD for the purpose of the model | [7], [10], [11], [12] |
| 2) | Uncertainty about the level of information in the model | Unclear requirements for information to be included in the model, or lack of them | [1], [3], [5], [6], [7], [8], [9], [10], [11] |
| 3) | Interoperability issues | Lack of interoperability between BIM and Facility management technologies | [1], [3], [4], [5], [7], [8], [9] |
| | | Lack of requirements for specific system | [1], [3], [4], [5], [8] |

[a] [1] Parn et al., 2017; [2] Liu & Issa, 2016; [3] Pishdad-Bozorgi et al., 2018; [4] Kelly et al., 2013; [5] Edirisinghe et al., 2017; [6] Carbonari, 2016; [7] Nical & Wodynski, 2016; [8] Matarneh et al., 2019; [9] Codinhoto & Kiviniemi, 2014; [10] Ashworth et al., 2019; [11] Volk, et al., 2014; [12] Leite, et al., 2011; [13] Peng, et al., 2017; [14] Rasys, et al., 2014.

Edirisinghe et al. (2017) reported study cases in which the analysed design model revealed missing geometry, problems with the level of information about components and systems, level of detail issues, and other modelling problems. This situation is worse in existing buildings since the high effort to capture the existing building data in semantic BIM objects make BIM implementation rarer in these cases (Volk, et al., 2014). To improve the level of geometry accuracy of the as-built and as-is BIM models, 3D laser scanning is rising in use but the time, cost, scanning range, and the precision rate are still limitations for this technology (Carbonari, 2016).

Moreover, although a BIM model is great for data integration, it may not be presented in a pertinent semantic format for facility management (Parn, et al., 2017). The BIM deliverables need to be assessed regarding if the required data is modelled, with their correct format and level of quality to meet the requirements of the data user (Pishdad-Bozorgi, et al., 2018). A BIM model is only useful for cost estimating, energy simulation, and creation of fabrication drawings, among others if it has enough details about the building components and systems (Leite et al., 2011). Volk et al. (2014) complements these issues stating that the high information level that needs to be included in the model for detailed maintenance or deconstructions considerations generally is not compatible with the current time or cost restrictions in the AECO and demolition sectors.

Furthermore, Parn et al. (2017) reports inconsistent naming conventions, specific facility management requirements missing, inadequate data categorization, and poor information synchronization between BIM and CAFM systems. Regarding the information to be included in the model, clear and well-documented requirements can address this issue, but most of the building owners have no document in place to state their needs for operation and maintenance phases (Edirisinghe, et al., 2017). Moreover, the lack of a systematic process to capture data for O&M in the construction and design phases results in facility managers not using BIM data to support their process because it does not contain the information needed or because it contains too much superfluous information for this stage (Pishdad-Bozorgi, et al., 2018).

Other Information issues that are documented in the literature are the problems related to the update of the O&M data sources. When no automation is used to update information about the fixed or movable assets of the building, the accuracy level of the O&M and BIM data integration relies on updates made by the O&M team (Parn, et al., 2017). Another side of this problem is the information management of large collections of data, Since the amount of data aggregated in the O&M stage can grow very fast with the inclusion of the daily management data of building components, like checking lists, operation history, indoor and outdoor sensor data (Peng, et al., 2017).

The last group of issues detected regards the lack of interoperability between BIM data and IT tools for facility management. As mentioned before, facility managers have a lot of applications to aid the operation and maintenance of the facilities but the data delivered from the design and construction phase is not integrated or compatible with these systems and hence doesn't support facility management practices (Edirisinghe, et al., 2017). Data problems can occur due to different units of measure, interpretation of the values or reliability of the values, and schema conflicts can happen with identifiers and naming mismatches, different attributes used to define the same information, and other structural inconsistencies from the divergent schema design (Rasys, et al., 2014). Thus, several solutions are in study to bridge the gap between syntax, schema, or semantics differences between BIM data and IT tools for facility management (Edirisinghe, et al., 2017).

One of them is the Construction Operations Building Information Exchange (COBie) standard. COBie provides a Microsoft Excel Spreadsheet-based structure for O&M data which allows the exchange of a vast amount of information embedded in the model (Edirisinghe, et. al, 2017). However, as mentioned, the industry generally doesn't have clear requirements to guide designers when including information in the BIM model to be extracted using Cobie (Kelly, et al., 2013) and filling all data fields offered by the standard would be overwhelming (Pishdad-Bozorgi, et al., 2018). Besides, most applications for facility management were not adapted to import COBie and use it to automate data input.

In this context, decentralized approaches for data or system integration gain force with studies growing in the field of semantic web technologies, mobile BIM, and cloud computing what bring challenges for the management of temporal data, transaction management, and synchronization (Volk, et al., 2014). Moreover, data loss is natural problem of integration that happens when information is converted to other data types to be exchanged between applications. This leads to gaps in the data collection or duplication of information and may result in data having to be re-entered manually (Codinhoto & Kiviniemi, 2014).

### 2.2.2 Potential uses for the integration of BIM and O&M data

Crossing the founds in the literature and the facility management tasks from the Atlas and ZEL study cases, it is possible to disclose potential use cases for the integration of BIM data and IT tools for Facility management (Table 4).

*Table 4. Potential use cases for the application*

| Areas | Data sources | Potential use cases |
|---|---|---|
| Maintenance | BIM data + Maintenance records | Fast retrieval of the maintenance history of a defective building element |
| | | Check maintainability conditions and plan interventions on the building |
| | | Aggregate work order execution per building areas |
| Logistics | BIM data + Asset tracking data | Fast location of assets in its spatial context |
| | | Aggregate office equipment supplies per area |
| Space Management | BIM data + Occupancy data/ HVAC and lighting systems data | Analyze popular and unpopular areas in the building versus the space characteristics (windows, furniture, appliances, environmental conditions) |
| | | Plan building intervention during refurbishment and retrofit with less impact for the public |
| Safety | BIM data + Alarm systems data/ Occupancy data | Analyze suitable location of the security sensors, alarm devices and signs |
| | | Make sensor data visible in a 3d viewer to monitor critical thresholds |
| | | Safety evacuation simulations |
| Indoor conditions | BIM data + Occupancy data/ HVAC and lighting systems data | Calculate the IEQ level and make it visible in the 3D viewer to monitor critical thresholds |
| Cost management | BIM data + HVAC and lighting systems data | Perform lighting, daylighting and thermal simulations in the building |

Both the Atlas building and the ZEL can benefit from a more smart way to manage the maintenance records and plan interventions on the building. The last has even more advantages for the ZEL since it has special MEP systems to deal with combustive fluids studied there that require a better planned intervention (see Appendix). In addition, Aggregate maintenance records per space can disclose trends and possible cause of recurrent issues (Akcamete et al., 2019). Besides, it enables the optimization of work orders execution based on spatial closeness that has the potential to save time, since staff expend less time moving the equipment and supplies necessary for the work. This can be particularly interesting for the Atlas Building that is big both horizontally and vertically but has spaces with similar maintenance requirements. Similarly, the routes for suppling office equipment can be optimized. In addition, the indoor navigation capabilities of the BIM data associate with in loco information of asset tracking systems can decrease the time to find assets in both buildings.

Crossing BIM data with space management systems data support the investigation on how to balance the building occupancy, necessary to achieve compact campus targets for the Atlas building, or to validate transitional safety measures related to the TU/e policy for corona virus. Moreover, the flow of people can be used to better plan renovation and refurbishment procedures aiming at less impact on the building users' activities. Also, material and volume information from the BIM model support calculation for estimate demolition and renovation waste (Cheng & Ma, 2013).

Safety uses cases include analyzing the location of security devices based on buildings parameters (e.g. possible visual obstruction, proximity with building systems that can cause interference in the readings), and occupancy parameters. Both data sources are also useful to perform evacuation simulations (Nguyen et al., 2019), important for both buildings since they have complicated evacuation scenarios. The ZEL because of the toxic and explosion potential, and the Atlas because of the high number of people to evacuate.

Furthermore, The smart systems in both buildings generate data that can be used to calculate the indoor environment quality (IEQ) level or the oxygen level of the spaces, this assessment can be translated into a color scale and associated to the BIM data using a BIM viewer for better monitor these critical thresholds (Chang et al., 2018; Riaz et al., 2014). These same systems can be used for simulations aiming at planning long-term strategies for the optimization of resources (Chevallier et al., 2020; Patti et al., 2012).

### 2.2.3 Approaches for integration of BIM and FM data in the literature

This section gathers a few of the studies initiatives that have a practical approach to the integration of BIM data and FM data that applied semantic web technologies, web technologies, or a combination of both. Considering the scope of the thesis, this evaluation aimed to understand what kind of data sources are used to support decision making in the O&M stage, how they can be integrated with the proposed technologies, and what functionalities they can support. Besides, They were evaluated regarding the existence of features to support user interaction, namely a user interface, a 3D viewer, and if they considered features for mobile devices. The summary of this analysis can be found in Table 5.

*Table 5. Related studies*

| Articles | Data sources | Integration approach | UI[a] | 3D Viewer | Decision support approach | Mobile devices |
|---|---|---|---|---|---|---|
| Patti et al., 2012 | Energy, temperature, humidity, and light sensors. | Web-based database to store heterogeneous wireless sensors networks. | x | x | This database is made accessible through the web to support a BIM-based energy-intelligent control service. | x |
| Chang, et al., 2018 | Humidity, temperature, power supply sensors, and BIM data. | Sensor data included directly in the Revit environment. | (Revit) | (Revit) | Measurement values converted in a colour scale displayed in the BIM model. | x |
| Riaz, et al., 2014 | Oxygen level, temperature sensor, and BIM data. | Sensor data included directly in the Revit environment. | (Revit) | (Revit) | Measurements values are displayed in tables and charts format, as well as in a colour scale in the model. | App send alerts to mobile phones. |
| Rasys, et al., 2014 | Engineering data, BIM data, and documents. | Data sources are linked using semantic web technologies, stored in a web database and included | (Navis works) | (Navis works) | The data is displayed associated to the Building elements. | x |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | in the Navisworks environment. | | | | |
| Shen, et al., 2012 | BIM server, asset tracking system, and facility operation and management system (FMM). | Decentralized system integration using semantic web technologies, and web technologies. | yes | yes | The information from the other systems is displayed in a single point of access. | x |
| Hu, et al. 2018 | Personnel data, BIM data, BMS data for MEP, including maintenance and repair data. | Centralized web app with different databases for the BIM data, user data, and monitoring data. | yes | yes | The app support data enquire, analysis and statistics calculation to support facilitate repair tasks, routine patrol path, and aid to emergency response. | Optimi-zed UI for Mobile devices |
| Kim, et al., 2018 | BIM data, Cobie spread sheets, and maintenance work records. | Data integration with semantic web technologies. | x | x | The user should be able to retrieve all records for a building element using SPARQL queries. | x |
| Rasmussen, et al., 2017 | BIM model, and the information embedded in it. | Centralized web app connected with Triplestore and Autodesk Forge. | yes | yes | The user is able to query the building data with 3D visual support. | x |
| Chevallier, et al., 2020 | BIM data, and several sensors data (up to 90 sensors). | Sensor data and BIM data were stored in web databases and linked using semantic web technologies. | x | x | This linked data is made accessible through the web to be used for simulations. | x |
| McCaffrey, et al., 2015 | BIM data, and sensor data from a BEMS. | Sensor data were stored in a web database, and visually associated to the BIM model in the user interface. | yes | yes | The app included 3D visualization with several interaction possibilities, such as time series aggregation, data display in the model with a colour scale, anomalous values detection and data exporting in CSV. | x |

[a] User Interface

Several studies aimed at a central database, considering the part of the O&M data they were dealing with, and integration with BIM tools. Patti et al (2012) proposed a LinkSmart middleware capable of aggregating data, in a SQL database, from a network of wireless sensors with data for energy, temperature, relative humidity, and light monitoring, and make it accessible through the internet, to later connect it to a BIM model and other applications. Chang et al. (2018) also stored sensor data for humidity, temperature, and power supply and additionally used Firefly suite, Dynamo, and Python to include this data in a BIM model in Revit. The study used a color scheme to help the facility manager to visualize the differences in values in a spatial context. Riaz et al. (2014) proposed to include data from wireless oxygen and temperature sensors in confined spaces on the field directly to the Revit environment as an add-in, to allow better monitoring of the workers' safety during maintenance procedures. The data from the sensors are stored in a SQL server and the connection is made in the Revit API. The system was also integrated with mobile technology to send alerts when the sensor detects values beyond the defined thresholds. Although the results were positive, the focus group that evaluated the prototype stated that a simpler user interface outside Revit is better for this sector since ''the construction workforce is not very technology literate'' (Riaz, et al., 2014).

In a different approach, Rasy et al. (2014) created a centralized data storage for heterogeneous engineering information with the data requirements defined by an ontology called Class Library for the field of oil and gas. The resulting data is then converted to JSON format and stored in MongoDB, a NoSQL database, and is made available on the web through a REST web service. The idea is to provide a single secure access point to the project data in a format that can be easily exchanged through APIs. To test the visualization of this data, a plug-in in Navisworks manager was created where data is displayed in JSON. The researches point out that with heavy and complex models a better approach is to include 3D data in a web portal instead of relying on Navisworks (Rasys, et al., 2014). In their opinion, it could make content easier to load and manipulate, and accessible in portable devices (Rasys, et al., 2014).

Shen et al. (2012) expanded the universe of the data integrated by including three web-based applications: a BIM server (BIM Octoplus), an RFID tracking system application (AeroScout), and a Facility operations and maintenance management system, using an agent-based and service-oriented approach. In this context, the authors defined that making this integration in a loosely coupled way, using ontologies for data integration and allowing both reactive and proactive services and exchange of data fits better the defragmented nature of the industry. In the same line of research, Hu et al. (2018) created a platform for the monitoring and maintenance of the mechanic, electric, and plumbing system of a 100,000 $m^2$ total floor area office building. A web application was used to display data from the RFID tags, that were used to label all elements and store real-time data about them, the MEP-related documentation that was previously digitalized, and the monitoring data from the BMS that directly control these systems. Moreover, BIM data was included in the web application for 3D visualization and it was converted to GIS 2D maps in mobile devices with less render capacity. The platform was also a maintenance management system and the workers used it to input new work orders that are automatically associated with the building locations.

These studies supported the potential of BIM to aggregate O&M information in the building's elements level. However, the first made live sensor data available for other BIM-based applications but didn't go further to connect it with these applications. The other three bring the facility manager to the BIM environment by using Revit and Navisworks as the user interface to interact with this 3D database. Considering the commonly reported lack of familiarity of this department with BIM tools, a better approach is to bring the BIM data to the facility management environment by creating an independent user interface. In this context, the approach proposed in this thesis in more close to the works of Shen et al. (2012) and Hu et al. (2018) with a web application as a single point of reference for the current building status for the facility manager to quick locate the building element and the information associated to it. But differently from these two studies, the app proposed here don't intend to replace the existing IT tools for FM since they are considered better designed to deal with tasks they are built to support. Regarding a portability context, BIM models of highly detailed facilitates can represent a considerable drop in performance in mobile devices or are not compatible with the open-source 3D engines available for front-end development. That makes the simplification of the model and its conversion to other data formats relevant aspects to be considered in the implementation of 3D viewers.

Other studies addressed the lack of query capability of the IFC model. Kim et al. (2018) linked data from IFC, after conversion to RDF with an OWL-based ontology called FM ontology, with maintenance data. An application was developed to allow the query of the created database supporting search in historical maintenance works associated with the building elements in IFC. In the topic, Rasmussen et al. (2017) went a step further to combine the RDF file, converted from IFC, to the 3D visualization of the data in a visual query interface. It enabled the user to make queries in a web application that displayed the 3D model filtered by the content of the request. The study points out some limitations of the ifcOWL ontology, and because of that, it applied the building topology ontology (BOT) when generating the RDF file.

These studies address the problem of generating a web-compliant IFC that allows the user to query information from it. However, the user interaction with the data required knowledge in SPARQL queries. This thesis proposed a layout where the SPARQL language is used behind the scenes to query the BIM data, and is translated in buttons or simple commands in the user interface. Similarly, the data received from these queries need to be post-processed to a more simple format to facilitate the interpretation of the results by the end-user.

Regarding support for decision making, Chevallier et al. (2020) used semantic web technologies to create a digital twin of a building with data about the sensors networks and a BIM model stored in a triplestore on the web. They used a common naming convention to link the domain ontologies and be able to query information about the sensor associated with its structure representation in the BIM model. Other databases were used to deal with the actual data collected by the sensors, and they were integrated with the triplestore using a middleware which also deals with the communication with a simulation tool that uses data from this digital twin to perform analysis.

On a different approach to support decision making, McCaffrey et al. (2015), which already had sensor data organized in a SQL database and a BIM model available, focused on the development of a client-side application to display this information with as many interactive functionalities as possible. The end-user could interact with the model by selecting the predefined model zones and obtain information available about it. Additionally, it included both spatial navigations, with keyboard and mouse, and time-series navigation, with the association of a calendar to allow the user to choose the time frame of investigation. The application also can generate charts from the sensor data and search for abnormal values and has the possibility to download data in a CSV format.

This thesis approach combine elements for these two studies, on one hand, it creates a linkable BIM model in RDF to which other datasets can be connected and retrieved by the app in a table format, but on the other hand, the sensor data is retrieved directly from the BMS and post-processing to a chart. This makes the post-process of data to have less intermediate steps since these datasets are dealt with in different components in the application and are manipulated in a different way. Besides, the challenges of collecting and storing sensor data are handled by the BMS what makes the app infrastructure more simple.

## 2.3 Semantic web

The semantic web extends the current web by providing a standard structure for data representation and reasoning (Patel & Jain, 2019). The idea behind this is to enable machines to interpret the content of the files published and to process the knowledge they capture from it in formally defined ways (Dibley, 2011). A machine can find a document on the web, but it doesn't know what the file is about, what is the context of the information provided by it, and how it is connected to other files available in the web. With semantic web technologies, machines can't 'understand' in the same sense that humans do, but they offer the open standard, layered, distributed infrastructure that allows them to process on behalf of the users the knowledge available on the Web (Dibley, 2011). In this context, the Semantic Web supports a 'web of data' instead of a 'web of documents' that provide computers with the capacity to do more meaningful tasks, such as searching, combining, and mining data (Patel & Jain, 2019).

Semantic Web Technologies support the creation of data stores on the web and the definition of vocabularies and rules for handling its content (W3C, n.d.-b). They include different data interchange formats (e. g. Turtle, RDF/XML, N3), query languages (SPARQL, DL query), ontologies, and notations like Web Ontology Language (OWL) and the RDF schema (Patel & Jain, 2019). In the web context, ontology consists of taxonomy, which defines the relationship between classes, and inference rules, to describe real-world entities (Patel & Jain, 2019). Ontologies make it possible to combine deep domain knowledge with raw data, and bridge datasets across domains (Patel & Jain, 2019). OWL is the standard language to communicate ontologies in the semantic web developed by the World Wide Web Consortium (W3C) (Dibley, 2011).

The RDF standard is an interoperable format for data linking and sharing on the web. The RDF graph describes the data in subject, predicative and object structures (e.g. the building [subject] contains [predicative] the 8$^{th}$ Floor [object]), each of these elements are identified with a uniform resource identifier (URI) that are used to create links among these triple structures by matching on the URIs (Curry, et al., 2013). An example of the graphic power of RDF can be seen in Figure 3, where the nodes and arcs identified by URIs make it possible for both humans and machines to understand that the entity is a person, with name Eric Miller, that has a particular title and email. Moreover, because URIs can uniquely identify resources and relationships and are HTTP based, the web infrastructure can access the data distributed across domains (Curry, et al., 2013). RDF can be queried using SPARQL query language and associated protocols format  (W3C, n.d.-a). In the context of the semantic web, 'query' means the technologies and protocols that can retrieve data from the web of data.

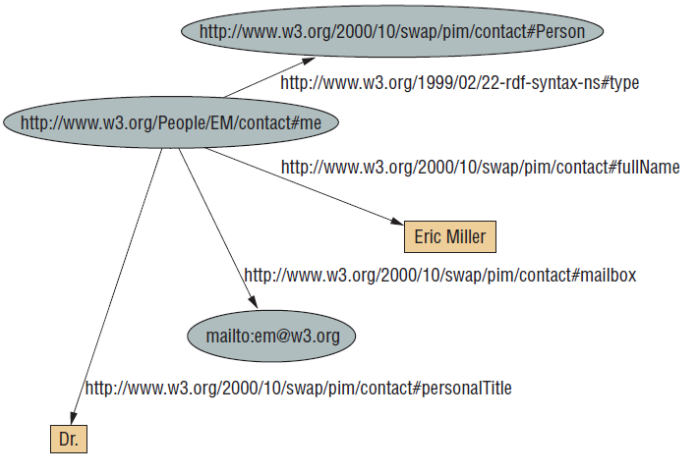



*Figure 3. An RDF graph (Shadbolt, Hall, & Berners-lee, 2006).*

An important concept associate with semantic web and enabled by it is linked data. The four principles of how linked data should be published on the web are (Berners-Lee, 2006):

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names on the web
3. When someone looks up a URI, provide useful information using the standards (RDF, SPARQL)
4. Include links to others URIs, so that they can discover more things

Furthermore, linked data is expressed in open, non-proprietary formats hence can be accessed by a variety of applications. Linked data is modular, since it can be combined with other pieces of linked data, and it is scalable, because it is easy to expand this network of data by connecting more and more linked data (Curry, et al., 2013). These concepts rely on ontologies as the backbone for structuring the linked data and define links within datasets and across datasets (Patel & Jain, 2019). This Linked Open Data[9] in the web form a cloud of interconnected RDF datasets that provide knowledge for several sectors of the society. With a similar approach for the building life-cycle data, the applications for this field would be able to theoretically access a larger and more diverse information source (Pauwels, 2014).

Moreover, Semantic Web technologies and linked data principles bring benefits for a decentralized integration approach for the AECO industry. It is created in a way that allows it to be easily linked at the information level (data), not in the infrastructure level (system) (Curry, et al., 2013). Hence, it complements the existing IT infrastructure as a technology for data sharing. The data is exposed within the existing system and only linked when the information needs to be shared (Curry, et al., 2013). Thus, to overcome the IFC limitations, semantic web technologies and linked data approach are getting increasing attention as a way of making building data web compliant allowing information exchange between different silos of knowledge, although some considerations can be made regarding usability, practical, and technical issues and the management and maintenance of the linked triples (Pauwels, 2014).

### 2.3.1 Semantic web for BIM data

As mentioned in section 2.2, the IFC format is the most consolidated standard to exchange BIM data across the building lifecycle but it has some limitations that prevent BIM to achieve its full potential.

Beetz et al. (2009) mention the lack of formal rigidness, limited reuse and interoperability, and lack of built-in distribution as a limitation to use STEP technologies in the context of the semantic web. OWL and other description logic-based ontology definition languages have a mathematically rigid theory behind them, which allows them to profit from existing 'intelligent' algorithms and technologies that IFC can't (Beetz, et al., 2009). Moreover, EXPRESS fit reuse and interoperability characteristics only in a few engineering domains, outside them its use among developers and the existence of affordable and free tools for it is limited (Beetz, et al., 2009). The

[9] https://lod-cloud.net/

lack of built-in distribution regards the structural limitations of mechanisms for STEP technologies to support multiple schemas and instances to be distributed, mapped, and merged among different resources (Beetz, et al., 2009).

They are complemented by Pauwels et al. (2017) that state as challenges for IFC to achieve interoperability binding issues, adaptability issues, and extensibility issues. The binding issues are originated by unintended geometric transformation and semantic errors caused by heterogeneous IFC translations and binding processes of the different BIM authoring tools (Pauwels, et al., 2017). Adaptability issues are related to the difficulty to quickly adapt the schema to meet the industry needs as it requires a lot of work to agree upon a change and generate the improved version (Pauwels, et al., 2017). Extensibility issues reflect the necessity of familiarity with EXPRESS language to be able to extend the schema (Pauwels, et al., 2017). Bonduel et al. (2018) also highlight as limitations of IFC the difficulties to apply generic reasoning and query methods to it due to the lack of methods to define formal semantics in EXPRESS and the large size and complexity of the IFC schema for software implementation.

Because of these limitations, several ontologies have been studied to convert IFC in RDF culminating in the ifcOWL[10] ontology, a standard supported by buildingSMART. IfcOWL is a connecting point between IFC and the semantic web technologies that maintain the use of the well-established IFC standard for building data and at the same time allow the exploration of the advantages of the semantic web technologies (Pauwels & Terkaj, 2016). The IfcOWL ontology allows building data to be represented in RDF, similarly as EXPRESS is a schema for IFC representation and XSD for XML (BuildingSMART, n.d.-a). BuildingSMART has recommendations for this conversion procedure and based on them several tools exist, namely ifcDoc[11] tool, IFC-to-RDF[12], and EXPRESS-to-OWL[13] converter.

However, although ifcOWL solves several IFC issues, it aims to mirror the IFC EXPRESS schema resulting in a still large and complex RDF file which makes it difficult to retrieve information from it, since query writing becomes longer and more intricate (Bonduel et al, 2018). As a result, simplified solutions were proposed to enable real industry applications.

Examples are simpleBIM, an approach to post-process RDF graphs compliant with ifcOWL to omit geometry and intermediate relationships instances between objects (Pauwels & Roxin, 2016). IfcWoD, an ontology that simplifies the relationships derived from ifcRelationship and the semantical representation of IfcPropertyAbstraction subtypes to improve query performance (Farias et al., 2015). Linked Building Data (LBD) by W3C and its dedicated group W3C LBD-CG including BOT[14], a simplified ontology to define the topology of a building (Rasmussen, et al., in press), PRODUCT[15] for classification of individual building objects, PROPS[16] to assign building-related properties, still in a conceptual design stage (Bonduel et al., 2018), and OPM[17], the Ontology for property management, that deals with properties that change as the building evolves (Rasmussen et al., 2018). More recently, the Building Product Ontology (BPO)[18] for assembly structures and the interconnections between its components were created to improve issues found in other ontologies including PRODUCT and PROPS (Wagner & Uwe, 2019). Other developments are the Building Element Ontology[19] (BE), to define building elements, and the Distribution Element Ontology[20] , for MEP systems.

In this thesis, the LBD ontologies were used through the adapted 0.1 version of the IFCtoLBD[21] converter branched from the 1.74 version of the original tool. This version also includes the Building Element Ontology and

[10] https://www.w3.org/OWL/
[11] https://github.com/buildingSMART/IfcDoc
[12] https://github.com/pipauwel/IFCtoRDF
[13] http://www.terkaj.com/tools/ExpressToOwl
[14]https://github.com/w3c-lbd-cg/bot
[15] https://github.com/w3c-lbd-cg/product
[16] https://github.com/maximelefrancois86/props
[17] https://w3c-lbd-cg.github.io/opm/#
[18] https://w3id.org/bpo
[19] https://pi.pauwel.be/voc/buildingelement/index-en.html#overv
[20] https://pi.pauwel.be/voc/distributionelement/index-en.html
[21] https://github.com/pipauwel/IFCtoLBD

the Distribution Element Ontology. BOT is the core ontology of LBD and its more mature report, it is lightweight and modular what allows it to be used in combination with other ontologies and provide simple means to link the described entities with their 3D model (Rasmussen, et al., in press). The building topology includes zones and sub-zones, to define entities with 3D spatial extent following a matryoshka doll principle (Figure 4), elements and sub-elements, for construction entities with technical function, form or position and the entities they host, and interfaces, to describe the relationship between zones, elements, or zones and elements, and that can be used to attach additional information about these areas (Figure 5) (Rasmussen, et al., in press).

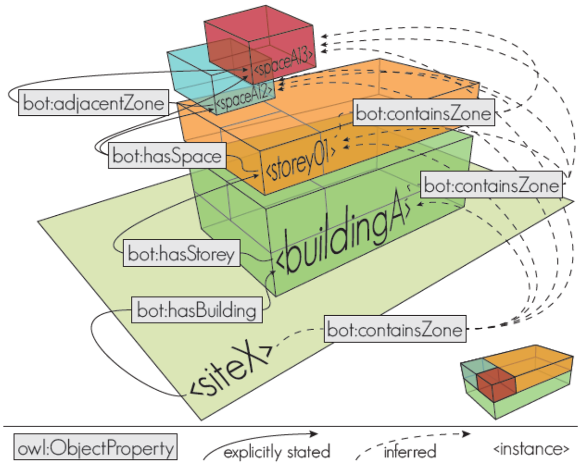



*Figure 4. Zones representation in BOT (Rasmussen, et al., 2017)*

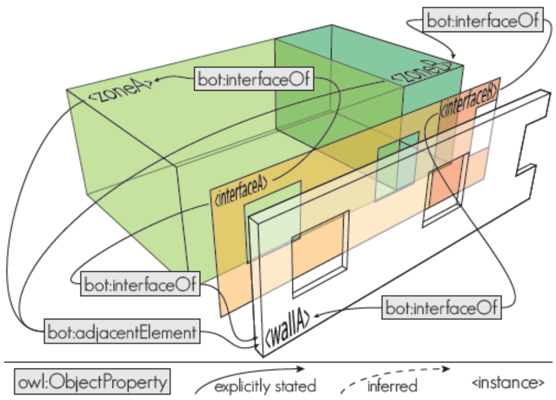



*Figure 5. Interfaces between two zones and a wall element (Rasmussen, et al., 2017)*

The IFCtoLBD tool is an open-source java-based project to handle the conversion of all current available IFC schemas to RDF using the LBD ontologies BOT, PROPS, and PRODUCT (Bonduel et al, 2018). It first uses internally the IFC-to-RDF converter based on ifcOWL to generate a file that is incrementally converted to one or multiple LBD based files according to the user settings. This process generate a more concise file in comparison with the RDF based in ifcOWL, with 83% fewer triples and 80% reduction in file size using all three modules exported to one file with PROPS level 3 (Bonduel et al, 2018).

## 2.4 Web-based futures

Information and communication technologies, especially internet and web-based technologies, evolved fast in the past 15 years for different application domains including engineering, construction, and facility management (Shen, et al., 2010). They provide a set of solutions to support collaborative creation, management, dissemination, and use of information (Shen, et al., 2010).

Web technology refers to the tools and techniques used to make machines communicate with each other in a network. They include web browsers (e.g. google chrome, Internet Explorer), web sites and web applications, and the programming languages and frameworks used to develop them, databases to manage the data required or received by them, and communication languages and protocols to effectively enable communication between devices and servers. The evolution from more informative web sites to an intense interactive web application represented a big step for these technologies (Conallen, 1999). Web applications are a web system in which input from the users changes the state of the business and hence this interaction is the main concern when implementing the business logic of the system (Conallen, 1999).

Web applications are praised for their flexibility in access, its virtually infinite storage capability, its control possibilities using login and passwords, and as a way to provide the most up-to-date information to all its users (Fox, 2018). But it also presents some challenges. As the web technologies scale-up and web content becomes available to users everywhere, tougher security measures will have to be taken to protect private web content from malicious cyber-attacks (Oppliger, 2003). Especially in mobile applications where logic inconsistencies are more frequent (Mendonza & Gu, 2018). Data privacy is another concern when applications rely on external servers and cloud services (Livraga , 2015). And, of course, the reliability on a good internet connection can be an issue for remote site work and inspection, what makes the possibility of offline work with synchronization options important features to be considered (Maes et al., 2016).

### 2.4.1 Web applications

A basic web application architecture is composed by browsers, a network, and a web server (Maes et al, 2016). The browsers get web pages from the server, which are a mix of content and formatting instructions expressed with HTML, and the user interacts with the content in these pages (Maes et al, 2016). Web servers can have behind them a database to handle the information exchanged in the web-based systems (Shen, et al., 2010).

This architecture is a client-server system. The client-side is also called front-end, which is what the user sees and interacts with, in other words, the user interface. The server side is the back-end, which englobes the background work and mechanisms that make the web page functional, handled by the application programming interface (API). The API is outside the user's computer, but it is what enables the application's visitor to access databases, create files, read and include other files, create content, send emails and connect to other servers and get data from them (Heilmann & Francis, 2007).

As mentioned, HTML, current in its 5$^{th}$ revision, is very important to the front-end development of web applications. It is a mark-up language used for structuring and presenting content on the internet, in combination with CSS and a series of JavaScript APIs, it allows the creation of complex applications that previously could only exist for desktop platforms (McCaffrey, et al., 2015). It is also suitable for cross-platform mobile applications as its features were designed taking into consideration low-powered devices such as tablets and smartphones (McCaffrey, et al., 2015).

On the back-end side, APIs of different applications and web services exchange data using their endpoints, where resources that can be accessed by third-parties are stored. The most common data formats exchanged between APIs are XML and JSON and this communication is possible due to several protocols, one of the most important is HTTP (Heilmann & Francis, 2007).

HTTP is an application-level protocol that allows machines to send and receive messages in a request-response way. Both types of messages have as a generic structure a start-line, zero or more header fields, and possibly a message-body (Network Working Group, 1999). The start-line in a request message specifies the HTTP method to be used in the resource specified and the identifier of this resource (Network Working Group, 1999). The identification of the resource is done by a URI. The basic HTTP methods and their application according to the architectural principals of a REST API are (Rodriguez, 2015):

- POST: to create this resource in the server
- GET: to read the resource specified in the message
- PUT: to update, replace or to change the state of that resource
- DELETE: to delete the resource

The start-line in a response message has the status code and textual phrase of the operation (e.g. 200 ok, 404 Not found) (Network Working Group, 1999). Moreover, the header fields pass general information about the message, such as the date and time it was sent, additional information from the client-side or the server-side, such as username and password, or metadata about the entity body or the resource identified, like accepted data types and content-length. The message-body is the body of the response or the request. The actual data requested from a database comes in the body entity of the response and, similarly, a POST request has in its body the data to be included in a database.

Furthermore, web applications are created using programming languages and frameworks, that give a basic structure to facilitate web development. Java, JavaScript, TypeScript, Python, Ruby, PHP, Swift are common programming languages. Node.js Express, Ruby on Rails, Laravel, Django and Flask are known frameworks more associated with back-end development and Angular, Vue, and React with front-end development. This study used Angular[22] to build the prototype web application, as it is a robust and complete framework that uses TypeScript (a superset of JavaScript) for the development of applications, supported by Node.js[23], an open-source JavaScript interpreter. It was used to build both front and back-end taking advantage of the available HTTP client module.

The angular architecture is component-based. This means that the user interface generated by it is an aggregation of components. Each one with its template in HTML commonly associated with a style sheet like CSS, that defines what the user sees on screen, and with typescript files that control the data, functionalities, and interactions of this component.

Angular has a special class called service. A component can delegate the task of fetching data from a server to a service to increase the modularity of the application and allow the reusability of this task by other components (Angular, n.d.-d). Angular also uses the concept of modules. A module is a block of code that groups components and services, among other code files, that are dedicated to an application domain, workflow, or a strictly related group of capabilities (Angular, n.d.-c). A project in Angular has at least one component and one module, the app component and the root module. Figure 6 brings these concepts in an illustrative way.

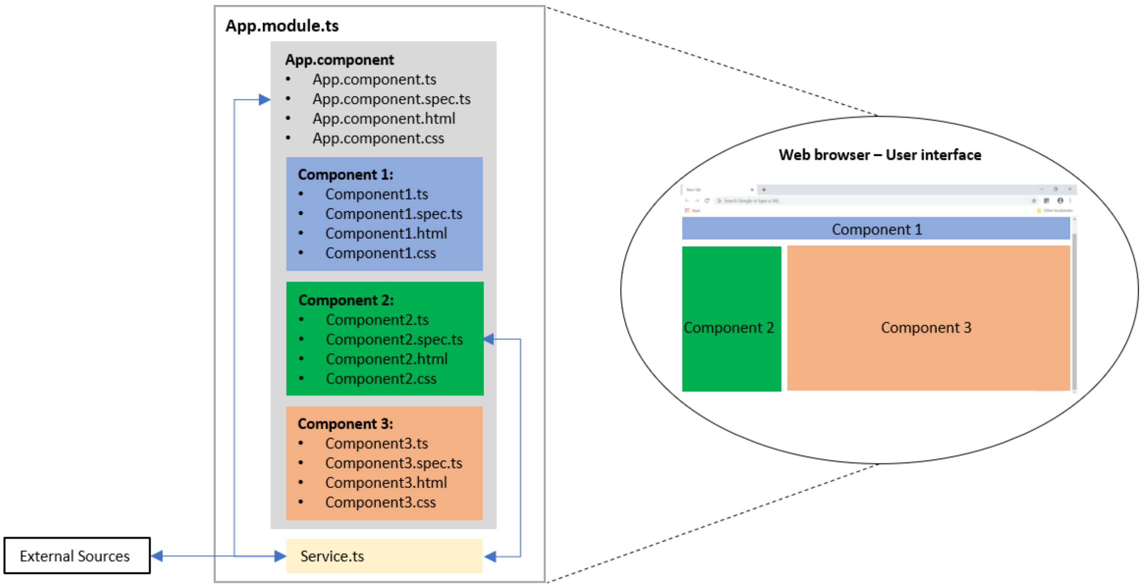



*Figure 6. Angular architecture*

### 2.4.2 Web Databases

A database management system refers to the collection of different tools and programs that enable the user to store, modify and retrieve data based in several parameters (Fatima & Wasnik, 2016). Based on how it manages

[22] https://angular.io/
[23] https://node.js.org/

the data, a database can be classified as a relational database, also called SQL based on the language used to query it, non-relational databases, or NoSQL, and in modern relational databases, or NewSQL (Fatima & Wasnik, 2016).

A SQL database is a simple data model to store and process data. It uses a structure and language consistent with first-order predicate logic to store data in form of tables (Fatima & Wasnik, 2016). However, its rigid data structure makes it harder to use when dealing with large data collections, especially when reading and writing real time data, and it can't handle the different data types that are emerging with the advance of the web technologies (Benymol & Abraham, 2020). In response to these challenges, a new type of relational database was developed that support new analytics capabilities and deals better with the scalability potential of the web, the NewSQL databases (Kepner, et al., 2016).

On the other hand, another solution for these issues was the NoSQL database, which is an umbrella term to databases that don't have a relational approach when dealing with data. Key-value stores, document stores, column family stores, and graph databases are sub-categories of NoSQL databases (Fatima & Wasnik, 2016). The general characteristic of NoSQL databases is to be schema-less, instead of the strict schema of relational databases (Chandra, 2015). Others common characteristics are to have a shared-nothing architecture, meaning each server has its local storage which gives the access the same speed of a local disk, and elasticity, meaning it can be dynamically expanded (Chandra, 2015). It is important to highlight that because of their differences each database has unique strengths that will make them suitable depending in the workload (Kepner, et al., 2016). For instance, SQL deals well with transactions tasks, NoSQL is good for internet searching, and NewSQL perform very well in data analysis (Kepner, et al., 2016).

In this study, a NoSQL graph database called Ontotext GraphDB [24] is used to store the RDF files generated by the IFCtoLBD tool. Graph databases store data as nodes and relationships with unique identifiers and hence they are good for mining meaning in heavy interconnected data (Fatima & Wasnik, 2016). Tripestores, like GraphDB, are a graph database specialized in the storage of RDF triples. It fitted in this thesis objectives because it has a free license option and uses the RDF4J[25] framework to enable SPARQL queries and API communication. This is necessary for the proof of concept application to be able to retrieve data from it to be displayed in the user interface.

## 2.5 Literature conclusion

This literature chapter proposed a review in the topics of facility management, BIM, semantic web, and web technologies, under the scope of the information management challenges in the O&M stage and the potential benefits for BIM data integration. The facility managers rely on very heterogeneous sources of data to make daily decisions and to base long-term sustainable strategies for the facility lifecycle. However, the coexistence of hard copies files, digital drawings and models, and different IT applications makes it very time consuming to find all the necessary information to support strategic decisions in this field. Moreover, some of these IT tools are unable to give spatial context to the information available and require manual input of data.

Considering this scenario, BIM data have potential benefits for this stage. The 3D visualization of the building brings possibilities for better plan interventions on the facilities and check safety measures, verify the maintainability of the building systems, or study strategies for deconstruction, disassembly, renovation, and retrofit. When the BIM model is combined with sensor data it could support simulations for optimization of resource usage. With the correct level of modeling detail and embedded parameters, it can support calculations for lifecycle assessment, carbon foot-printing, and recycling rate. When connected with asset tracking tags, it can give the actual position of the assets in the building, and with O&M data included in the model, it becomes a more organized database where the facility managers can fast obtain information about a building element.

The potential use cases for the application were obtained combining these BIM potentials for the operation and maintenance processes with the tasks described for the two study cases. In the maintenance field, a web-based application for BIM data and O&M data could become a better organized database for the building elements historical conditions and maintenance reports, and support maintainability assessments and work order

[24] https://www.ontotext.com/products/graphdb/
[25] https://rdf4j.org/documentation/reference/rest-api/

execution planning. Similarly, the logistics process could be improved with a BIM powered building indoor navigation to locate assets and better plan equipment replenishment. Moreover, the application can be a platform for spatial trends analysis and provide data for refurbishment and retrofit procedures. Other uses are to monitor safety and indoor quality thresholds, and support safety devices positioning, and simulations for emergency scenarios and building performance like lighting, daylight, and thermal studies.

Semantic web and web technologies can enable the potential of BIM For O&M. Semantic web technologies provide the structure to link different data sources to BIM data. Machines can understand those links and retrieve relevant data based on specific queries. Web technologies enable the construction of powerful web applications that are supported by web databases to store the constantly growing O&M related data. The front-end can be customized to support the client needs, with some examples from the related studies being display data using a color schema in the model, time series navigation of the datasets, sensor data showed in chart format, performing calculations, and sending alerts to inform values above defined thresholds. Web technologies also enable cross-platform development for mobile applications allowing the staff to bring data to the field to support decisions on the fly.

Based on the evaluation of other applications in the literature, is possible to say that the proposed system architecture should have an easy to use user interface, that can display the O&M data connected to the BIM data. In this user interface, the facility managers should be able to interact with the building database to access information about the element they desire, and the response needs to be easy to understand. The facility manager should not need deep knowledge outside his area of expertise to operate the app, hence technical language should only be used by the back-end of the application. Other desirables features are the implementation of a 3d viewer and support for portable devices. In addition, the BMS data has to allow data exchange with third-party applications and, for that, need to be reachable on the web.

The specific BIM conditions to support the O&M process depend on how it is intended to be used. But, in general lines, it can be said that it needs to be modelled with a compatible level of detail and geometrical accuracy, and include the important semantic relationships among elements. The information included in it has to be defined together with the facility management team and should result in the elimination of superfluous information for this stage and the inclusion of relevant O&M data. BIM data and the IT tools for facility management also have interoperability problems since the most used BIM standards are more compatible to exchange data inside the BIM tools environment. In the context of the developed application, the LBD ontologies and the Graph DB structure are used to enable the app to retrieve data from the BIM model.

# 3 Methodology

This thesis uses two study cases to extract facility management tasks that can be improved by the association of BIM data and O&M data. To complement the literature review in the topic, a proof of concept application was developed to build the basic infrastructure to enable this connection and analyses the potential and challenges of the proposed approach.

The first study case is the renovation of the Zero Emission Lab, under construction, in the TU/e campus. The design files are assessed based on the common problems associated with 3D modelling extracted from the literature research. This assessment was quantitative, as the issues found were analyzed according to the number of times they occur and the elements they affected. It aimed to suggest improvements to the final 3D model to be delivered in the closeout phase.

The second study case is the Atlas Living Lab, also in the TU/e campus. It is a laboratory where data about the operation of the lighting system in the Atlas building is captured live, using multifunctional sensors. This second study case is used to develop the proof of concept. The 3D model of the 8$^{th}$ and 9$^{th}$ floors of the Atlas building was made by the researcher incorporating the conclusions from the first study case. The development of the application followed typical software development stages depicted in Figure 7.

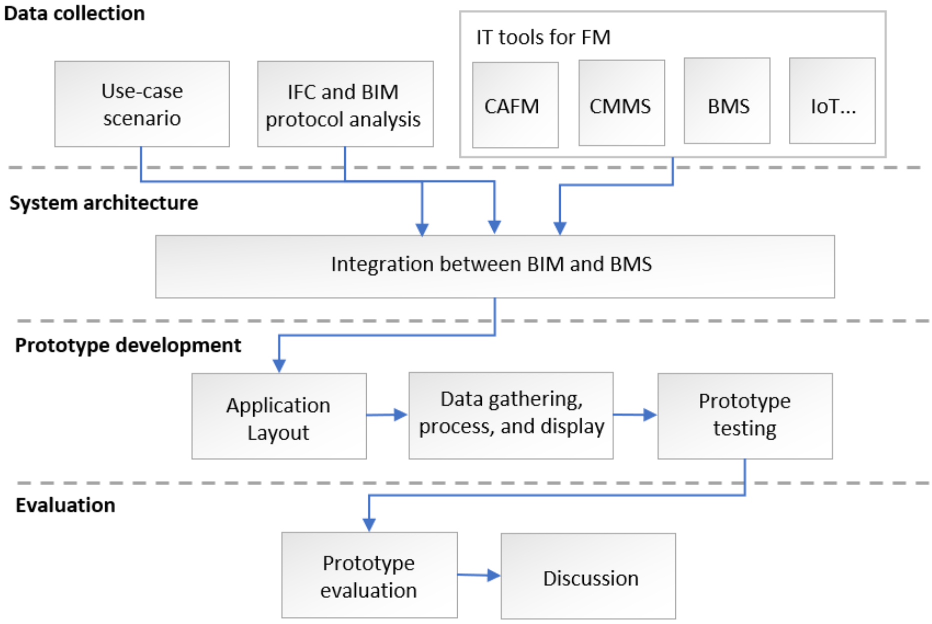



*Figure 7. Research model (adapted from (Arslan, et al., 2014))*

The data collection of the web app consists in the literature research included in the second chapter of this thesis, where the problem of integrating BIM data and O&M information is explored in-depth, and also in the conclusions from the evaluation of the IFC files of the ZEL study case. In the next stage, the system architecture is outlined based on the proposed use case and in the characteristics of the available data sets. The development stage implements the defined architecture until an alfa test phase. The final stage is the evaluation of the results obtained with the application regarding information missing, presence of errors, level of automation, and the general application behavior.

Thus, it is possible to say that this research used both quantitative and qualitative methods. The study cases and the software development stages are going to be further explored in chapters 4 and 5.

## 3.1 Methodological justification

The method applied in this thesis was a research-based design since it aimed to develop a prototype application to integrate the BIM data and O&M data. The main goal is to evaluate the potential and challenges of the proposed application for this stage and how to implement it using semantic web and web technologies. This goal is divided into three research sub-questions to disclose potential use cases for the application, define the system architecture and its requirements, and verify how BIM data can be reached by it.

The first step in the prototype development is data collection. It includes two case studies. The ZEL, to identify modelling issues for BIM data to be used in the application using Solibri Model Checker. And the Atlas Living Lab, to define the use case scenario to be implemented through document analysis of the data policy and the laboratory infrastructure description. Moreover, the common facility management tasks from both the ZEL and the Atlas building are used to build the potential use cases for the application. These tasks were extracted from an unstructured interview with a policy advisor from the TU/e real estate department and the adjunct program coordinator for the Gemini project. The study cases are complemented by a literature review in the common IT tools for O&M, the potentials and challenges of BIM for this stage, and the basic system requirements to integrate O&M data and BIM data and other features of interest.

The system architecture stage translates the data collection in functionalities to be implemented in the prototype. It also brings the system preconditions to make the application operational since the sensor data storage will be handled by the Atlas Living Lab infrastructure and the BIM data is assumed to be ready for use in GraphDB and reachable through the RDF4j framework. Based on the use case scenario, the integration approach used was to include the sensor ID in the BIM model so that the app API could identify when this property matches the location ID from the sensor metric report and display both data sets in the user interface. This dynamic needs to be performed every time the user selects a building element. Hence, the app needs to query the triplestore whenever a new element is selected and run this verification on the BMS sensor data.

The prototype development stage involves the implementation of the application layout, the client and server-side mechanisms, and the prototype test. The app layout is based on the user requirements defined from the literature review and developed using the Angular framework and other open-source JavaScript libraries. It aims at a responsive and intuitive design with space for future features such as a 3D viewer and other sensors data. The application mechanism focus in retrieve data from external sources, find ways to process it into more simple formats than JSON and XML, and show it in the user interface with features for the facility managers' interaction with the database like filters and buttons. The prototype was tested according to the proposed system architecture using the Chrome DevTools.

The prototype evaluation includes the critical analysis of the obtained results, including the data conversion from the system preconditions steps for the BIM data. In the discussion chapter, The prototype is evaluated against the potential use cases defined in the data collection stage to highlight which tasks it is better designed to support and what tasks would need other O&M data and system functionalities to be implemented.

## 3.2 Study Cases

A study case approach is used to explain, describe, or explore a complex issue in a real-life context (Crowe, et al., 2011). A study case is valuable to scientific development since it provides context-specific validation necessary to support generic concepts and hypotheses (Flyvbjerg, 2006).

This thesis has two study cases. The Zero Emission Lab was used to analyze a BIM model and documentation in the design stage to identify problematic points to be improved for the handover documentation. Moreover, the compact campus approach of the university results in state-of-the-art solutions for facility management. This includes the installation of several building management systems in the renovation of the Gemini Building on the TU/e campus which made it relevant to use it as a study case in this thesis. In addition, a combustive engine laboratory brings interesting use cases related to the monitoring of safety thresholds, evacuation simulations, and maintenance of special building systems.

Similarly, the Atlas building increments the use cases with opportunities for optimization of natural resources with light and thermal simulations, assessments of indoor environmental conditions and occupation trend. Moreover, the Atlas Living Lab supplied this research with a second data set necessary to develop a proof of concept that ultimately aims at showing different sources of information together on the web.

### 3.2.1 Zero Emission Lab

The Zero Emission Lab (ZEL) is the laboratory of the Mechanical Engineering department where research about internal combustion engines is conducted, it is part of the North Gemini building owned by the Technical University of Eindhoven (TU/e, 2020b). Due to the aging of the building, it is unable to comply with the modern research standards, especially regarding ventilation and gas detection, what motivated the renovation program targeting the installations, facades, the liquid-tight floor in the ground floor, the painting and the fire separations to be finished in the third quarter of 2020 (TU/e, 2020b).

The Renovation of the ZEL is part of the real estate strategy Campus 2030 which includes the renovation of the whole Gemini complex by 2025. The strategy aims at making the campus future-proof with state-of-the-art facilities that comply with the future demands in education and research and adapt the campus to accommodate the increase in the number of students and employees (TU/e, n.d.-c). This spirit of innovation made it a fitting study case for this thesis since the Gemini complex is expected to host smart operational systems, such as ATES[26], a heat and cold central storage system, Envision Manager[27], Philips lighting management system, and Planon[28], a space management system.

The University vision is to convert part of the building in a living lab and special attention is being given to the elaboration of a digital twin of the building to further explore smart ways to manage and operate the building complex, although its future use is still being defined in close collaboration with the participants of the project organization in plenary sections (TU/e, 2020a). Achieving this goal depends on the quality of the information in the BIM model delivered in the as-built phase. Hence, this study case evaluated the information embedded in a temporary BIM model and its documentation elaborated in the design phase aiming to investigate the improvements needed for the final digital twin and to get insight for the development of the proof of concept. The BIM model was evaluated in the IFC format since it is accepted by most of the BIM authoring tools, but a more detailed Revit file is also available.

The evaluated models were the architectural design (1839_arch_020, Figure 8) and the mechanical, electrical, and plumbing project (20181214_VS_DO_TUe_ZEL_INS, Figure 9). The documents that regulated the information to be included in these models and their modelling process were also analyzed. They are the BIM protocol for the design phase, the information delivery specification, and its annex, the BIM project information delivery specification (48038.00_BIM protocol Ontwerpfase, 48038.00_Informatie levering specificatie Ontwerpfase, Bijlage 2_BIM Project Informatieleveringsspecificatie).

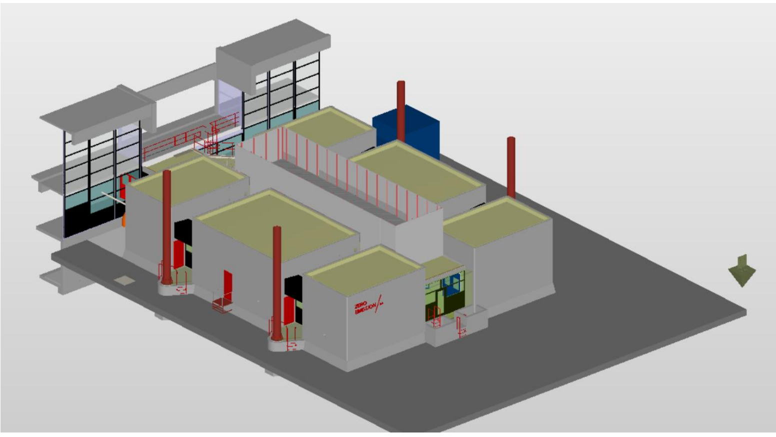


*Figure 8. Overview model 1839_arch_020 in Solibri Model Checker*

[26]https://www.tue.nl/en/our-university/about-the-university/sustainability/campus-and-operational-management/energy/heat-and-cold-storage-ates/
[27] https://www.dynalite.org/lighting-controls/software-apps/envisionmanager-system-manager
[28] https://planonsoftware.com/nl/

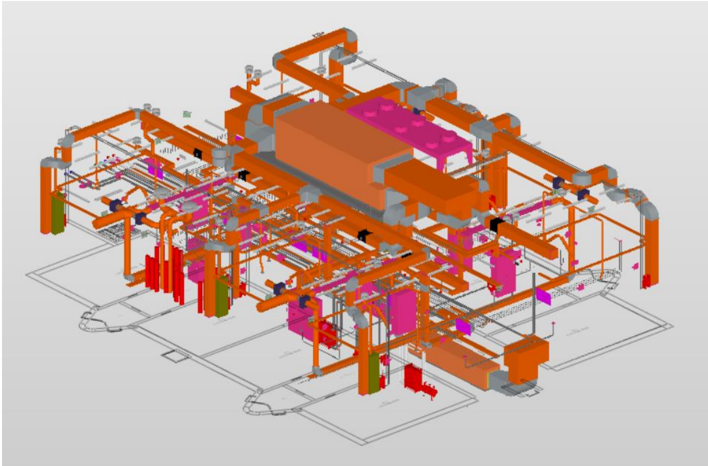


*Figure 9. Overview model 20181214-VS-DO-TUe_ZEL_INS in Solibri Model Checker*

Solibri Model Checker[29] was the main application used to assess the content of the IFC models. This software can generate a 3D visualization of the geometry described in the IFC file and displays the information associated with every element in the model in a table structure and the hierarchy relationship between the elements in a tree structure. Additionally, it is possible to upload several IFC files allowing the researcher to evaluate how they interact in 3D. This clash detection feature generates detailed reports about the elements intersecting and can be personalized to perform specific checks by changing the clash detection rules set. Another feature used in this evaluation was the quantity take-off to get the precise number of elements according to the characteristic analysed.

The evaluation of the models targeted the following topics:

- Describe the content of the models.
- Verify if all elements' generic properties and classification systems were present and displayed in the correct property set.
- Check the specific properties for spaces (number, area, unbounded height, and perimeter) and Lighting fixtures.
- Analyse if the relationships among the elements had de correct semantic meaning (aggregation and containment relations).
- Disclose possible intersections and duplicated elements.

The checks were performed in the light of what was agreed on in the protocols and specifications for the design phase between the project stakeholders.

### 3.2.2 Atlas Living Lab

The Atlas building renovation was part of the TU/e Campus 2018 targets, it was officially inaugurated on 21 March 2019. The whole process is aimed at the delivery of one of the most sustainable educational buildings in the world according to the BREEAM sustainability assessment (TU/e, n.d.-b). Smart building systems for temperature, light, and daylight control were installed in the facility to promote lower use of natural resources in the operational phase. Because of this available infrastructure and the nature of the project, the university also aimed from its conception to transform part of the building in a Living Lab (Cursor, 2019).

A living lab is a public or semi-public area where infrastructure is provided for scientific experiments regarding human subjects (Intelligent Lighting Institute, 2019a). The main difference from traditionally controlled experiments is that these areas are functional spaces used in real life (Intelligent Lighting Institute, 2019a). The Atlas Living Lab is an indoor living lab that aims to support long-term research in user behaviour patterns in energy use and its effects on the health, well-being, and the social interaction of its occupants (TU/e, n.d.-a). It collects data from the intelligent lighting infrastructure from the 4th until the 11th floor of the Atlas building work

[29] https://www.solibri.com/

environment (e.g. offices, study spaces, meeting rooms, and classrooms) except for toilets, technical spaces, and emergency stairs (TU/e, n.d.-a).

The Intelligent Office Lighting System available is provided by the Philips Dynalite features (Figure 10). The Head-end software of the Dynalite system is Envision Manager (EM). With it, multiple operators can control, monitor, and manage the whole lighting system (Koninklijke Philips N.V., 2014). It communicates with the hardware through a gateway that in turn communicates with the other parts of the system through Dynet protocol (Signify Holding, 2018a). Envision Manager retains the multipurpose sensors data which includes, among others, daylight-level detection, and occupancy detection and allows the facility manager, using a digitally addressable Lighting Interface (DALI), to group and regroup the lighting features in logical areas without the need for physical intervention (Signify Holding, 2018b). The users can interact with the lights using the physical user interface and with the Philips Dynalite Control mobile App.

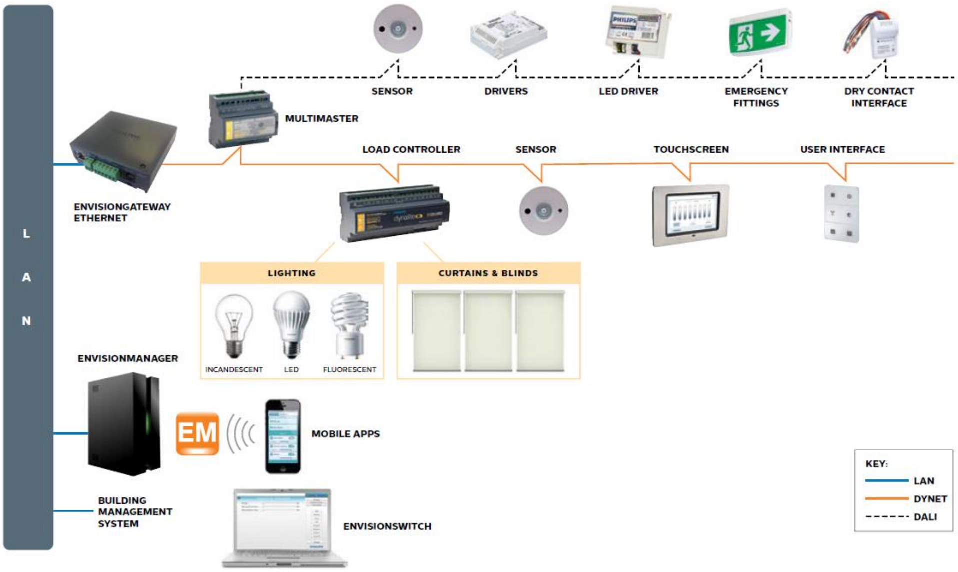



*Figure 10. Philips Dynalite system structure* (Koninklijke Philips N.V., 2014)

One of the resources provided by the Atlas Living Lab is an intermediate server with which the applications in development can connect to get the data extracted from the lighting system without interacting directly with its API (Figure 11). The complete infrastructure is named Research Environment Atlas Living Lab (REALL) and is a safe environment built-in virtual machines that provide to the researcher's applications controlled access to the lighting system API and the Atlas Living Lab database (Atlas Living Lab, 2019). This protective layer is necessary to safeguard the building from commands generated from the researcher's applications.

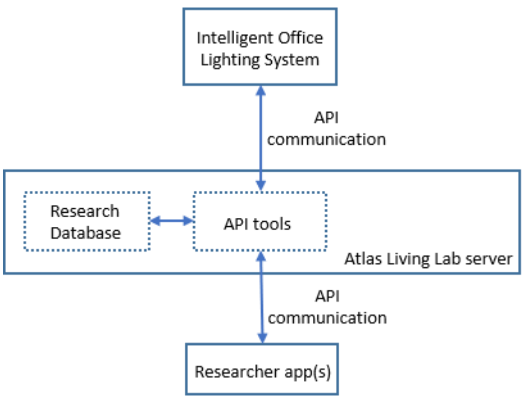



Figure 11. Atlas Living Lab research environment (adapted from (Atlas Living Lab, 2019))

Beyond this REALL research environment, the living lab infrastructure includes multifunctional sensors embedded in every two lighting fixtures, they have a measurement range for movement of 2m x 3m for desk level movements and 3m x 5m for intense movements (Intelligent Lighting Institute, 2019b). These devices can provide information about the energy consumption, aggregated per area and period in Watts, the temperature, inside the luminaires, and the percentage of occupancy based on the movement detection (Intelligent Lighting Institute, 2019b). This research used the energy metric report with hours as the time period for its proof of concept. It includes the maximum, minimum, and average energy consumed per hour throughout the day and the accumulated energy consumption in the beginning and at the end of the measurement (Intelligent Lighting Institute, 2019b). Figure 12 shows a sample energy metric report from the system virtual test environment which is provided as an XML format file.


```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:getMetricReportResponse xmlns:ns1="http://daa.ispf.philips.com">
      <metricsReportResponse xmlns:ns2="http://daa.ispf.philips.com/analytic">
        <locationID>00000000-0000-0001-0000-0000000007D5</locationID>
        <metricReportList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns4="http://daa.ispf.philips.com"
xsi:type="ns4:energyReport">
          <fromDateTime>2020-04-01T00:00:00Z</fromDateTime>
          <toDateTime>2020-04-01T00:59:59.999Z</toDateTime>
          <avgKW>0.0</avgKW>
          <endAccKWH>0.0</endAccKWH>
          <maxKW>0.0</maxKW>
          <minKW>0.0</minKW>
          <startAccKWH>0.0</startAccKWH>
        </metricReportList>
```


*Figure 12. Sample energy metric report*

Furthermore, a simplified model of the 8th and 9th floor of Atlas was made by the researcher in Revit 2018 to obtain a complete study case for the proof of concept, with BIM data in addition to the building management system data provided by the Atlas Living Lab.

## 3.3 Methodology conclusion

The thesis aims to evaluate the benefits and challenges of a web-based application that can connect BIM data with O&M data. Therefore, to implement and analyze this approach a research-based design was applied to develop a prototype application that could link the sensor data from a BMS lighting system and BIM data for the Atlas' 8 and 9th floors.

The first stage of prototype development is data collection. It consists of the literature review to disclose facility management tasks and supporting IT tools, potential and challenges of BIM to be used in this stage, system requirements for the app, and the semantic web and web technologies to enable their implementation. The literature review is complemented by the document analysis of the ZEL and Atlas Living Lab study cases. The ZEL BIM models and related protocols evaluation targeted the content of the models, the general and specific properties present in it, and the classification system used, the semantic relationships among building elements, and geometric inaccuracies. The Atlas Living Lab analysis aimed at the type of data obtained from the system virtual test environment and the REALL characteristics. Moreover, an unstructured interview was conducted to extract the facility management tasks for these two buildings.

The second stage is the system architecture where this data collection is translated into functional requirements for the application. They are implemented in the third stage, where the app layout and its front-end and back-end mechanisms are materialized. Furthermore, This stage is finalized with the application alfa test. The last stage was the prototype evaluation and discussion, where the obtained results are critically analyzed and the potential use cases and future challenges are discussed.

# 4 Proof of concept: data collection and system architecture

The development of the application followed four stages. This chapter focuses on describing the process of data collection and the development of the system architecture.

## 4.1 Data collection

As mentioned in the methodology chapter, the data collection is the initial phase in the development and implementation of the proposed prototype application. It includes the literature review, the analysis of the two study cases, and an unstructured interview with people involved in the Gemini renovation project and the TU/e real estate department.

The literature was based on the research questions, which in turn derivate from the thesis goals. It aimed at gathering data about the typical problems found in the BIM models delivered as part of the handover documentation, and the facility manager difficulties to generate knowledge when dealing with multiple sources of information. Furthermore, the interview aimed at defining facility management tasks that could be improved with the connection with BIM data. Other targets were the data structures incompatibilities that prevent a centralized approach to display building data and how a decentralized approach could facilitate this objective. This analysis included the features enabled by web applications and semantic web technologies that can be used to achieve this decentralized approach. Moreover, The ZEL BIM model and documentation were studied to complement the literature findings, and the lessons learned from this process were incorporated in the modelling of the Atlas model.

The Atlas Living Lab provided insight into the network structure of a building management system and the type of data generated by the sensors. Gathering information about the format, content, and structure of this data was necessary to understand what kind of transformations the system needed to perform behind the scenes to read and display in a user-friendly way the sensors' report in XML. Moreover, an appropriate understanding of how the available network functions is needed to comprehend how the prototype application can communicate with the database with the sensor readings.

The immediate results of the data collection are the scope of the use case scenario, which defines the system architecture, and the elaboration of the 3D model for the Atlas study case.

### 4.1.1 Use case scenario

The use case scenario for the proof of concept was elaborated considering the main goal of this master thesis and the available data from the Atlas Living Lab study case. It scopes the system functional requirements from the point of view of the end-user and brings the sequence of steps that will allow the user to achieve the main goal of the scenario.

This scenario was depicted using Unified Modeling Language (UML) use case standards that focus on describing the functional requirements of the system. In this standard, actors are not only people but everything that interacts with the proposed system. In this case, the facility managers are the primary actors in the application because they initiate events in the system. Moreover, the triplestore with the RDF triples generated from the IFC file and the Atlas Living Lab server are secondary actors of the system as they receive requests from it but don't initiate events themselves.

The description of the use case scenario is found in Table 6. It brings the definition of the main workflow of the application and its ramifications, as well as the preconditions for the correct functioning of the prototype.

*Table 6. Use case scenario description*

| | |
|---|---|
| **Scenario** | Access information from two datasets on the same web page. |
| **Actors** | Facility manager, triplestore, and Atlas Living Lab server. |
| **Precondition** | 1. The IFC file is converted to RDF<br>2. The RDF graph is imported to a repository in a triplestore. |

| | |
|---|---|
| | 3. The Atlas Living Lab server database is accessible to the application. |
| **Description** | 1. The facility manager accesses the application in a web browser.<br>2. The application requests the predicative and objects associated with the building entity to the triplestore.<br>3. The application requests the measurement data available for the sensors in the model from the Atlas Living Lab server.<br>4. The application renders a table with the content received from the triplestore.<br>5. The facility manager can interact with the table to get information about the elements inside the building. |
| **Extension** | 5a. When the facility manager selects an instance inside the element displayed (e.g. a storey inside the building element), the content of the table changes to show the information about this new element.<br>5b. The facility manager can select to go back to the previous element information displayed in the table.<br>5c. When the element selected is a sensor, the application displays the data from the Atlas Living Lab server in chart format. |
| **Exceptions** | When the element selected has no other datasets associated with it only the data from the triplestore is displayed in the application. |
| **Results** | The facility manager can visualize the information from the triplestore, in a table, and the data from the Atlas Living Lab, in a chart form, for a desired element inside the building. |

This scenario creates the basic infrastructure of the proposed approach to connect BIM data with BMS data, Figure 13 depicts it in a UML use case diagram. Considering the lab infrastructure, the expected sensor data are humidity and temperature measurements, occupancy status, and energy consumption readings. Some examples of information that could be retrieved from the BIM data are elements' reference, ID, location, geometric data like area, height, and position in relation to the ceiling or floor. Other examples are elements contained in the rooms or floors, and sub-parts of the building systems.

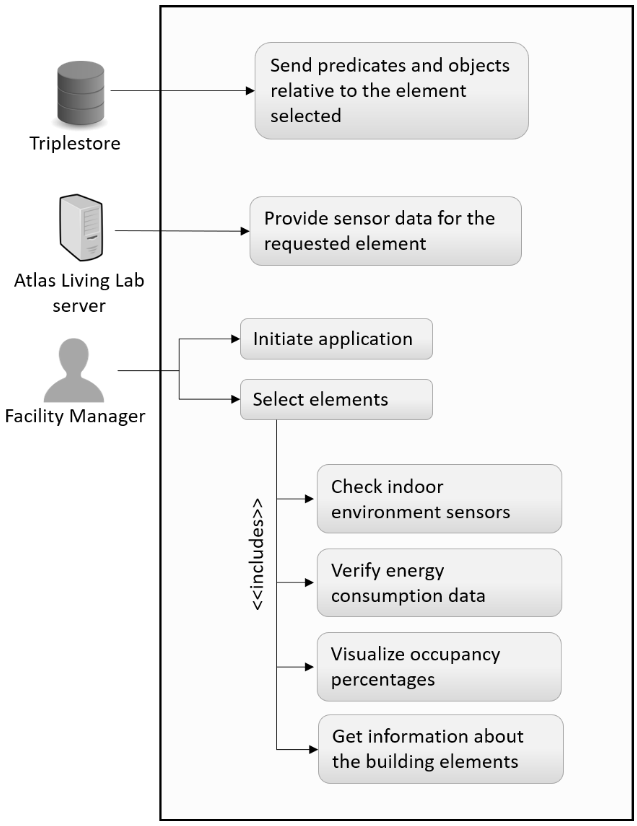



*Figure 13. Use case diagram*

### 4.1.2 3D model

The Atlas BIM model was developed by the researcher based on the scope of the use case scenario since this model was developed solely for the use in the proof of concept. The space identification used was the same used in the sample data from the BMS. The location name specified in the building navigation sample file was included in the ‘Name’ property of the ‘Identity Data’ property set of the spaces (Figure 14 and Figure 15). The actual function of the rooms according to the Atlas planning in March 2020 was included in the same property set in the ‘comments’ property.


```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Body>
  <ns1:getBuildingNavigationResponse xmlns:ns1="http://daa.ispf.philips.com">
   <buildingNavigation xmlns:ns2="http://daa.ispf.philips.com/analytic">
    <childLocations>
     <childLocations>
      <childLocations/>
      <description>
      </description>
      <location>
      </location>
      <name>Area_4001</name>
      <type>AREA</type>
      <id>00000000-0000-0001-0000-000000000FA1</id>
     </childLocations>
```


*Figure 14. Part of the Building Navigation sample report*

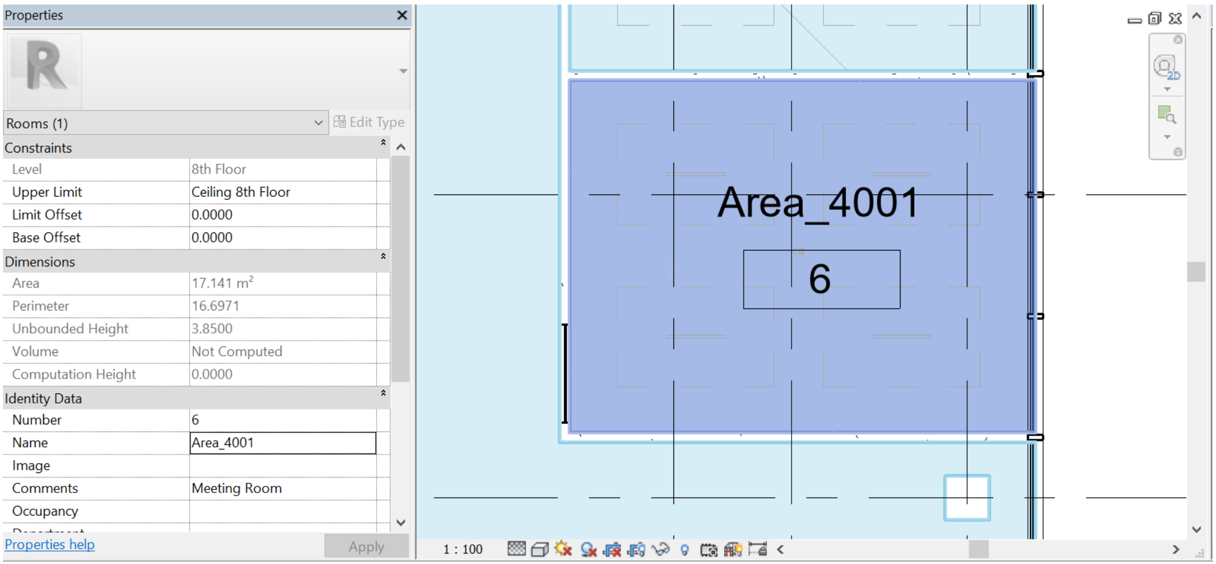



*Figure 15. ‘Identity data’ property set in a space entity*

The perception of a building commonly is building-> storey-> rooms -> rooms’ elements and the developed application aimed at using this logic to allow the user to navigate in the building. Hence, the Revit file was modelled to follow this structure with sensors and lighting fixtures contained inside the rooms they serve. Additionally, the absence of this relationship would not be ideal for the application since the sensor’s reports are aggregated per space.

Because of this structure, it was opted to add the ‘location ID’ from the metric report to the sensor entities in a property named ‘SensorID’ created by the researcher inside a property set called ‘data’ (Figure 16). It was possible to work like that because the ‘location ID’ is used as a reference to generate the sensor readings in the energy metric report. Hence, in a scenario where several BMS exist, each with their sensors, the ‘sensorID’ can be used to identify to each system that sensor belongs to. Similarly, the property ‘lightingID’ was added to the lighting fixtures in preparation to include the DALI reference code from the lighting BMS. The same approach could also be used to include the sensors in the ZEL Revit model.

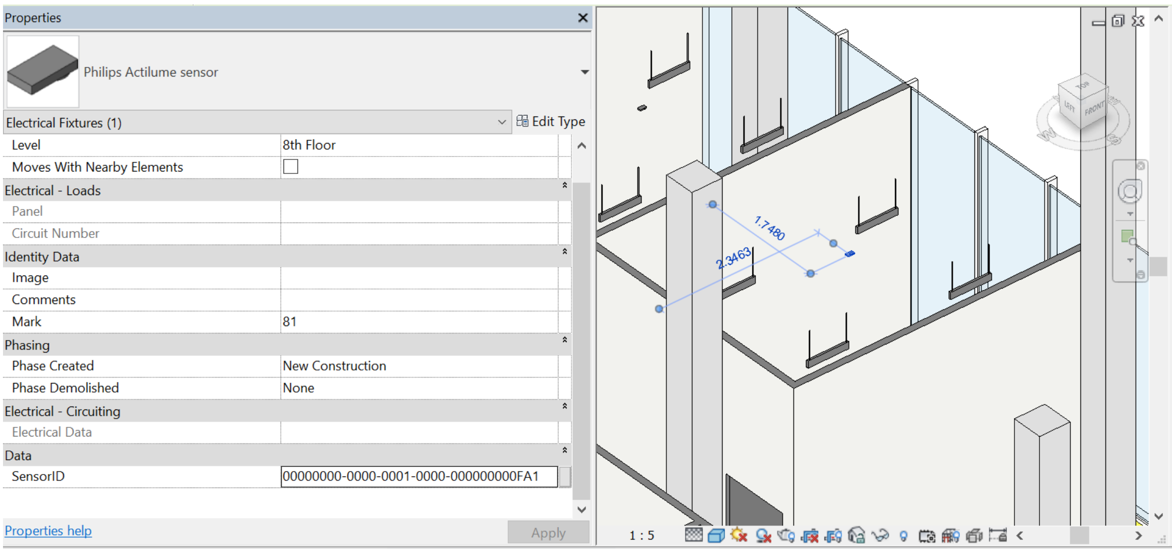



*Figure 16. 'SensorID' property in 'Data' property set created for the sensors*

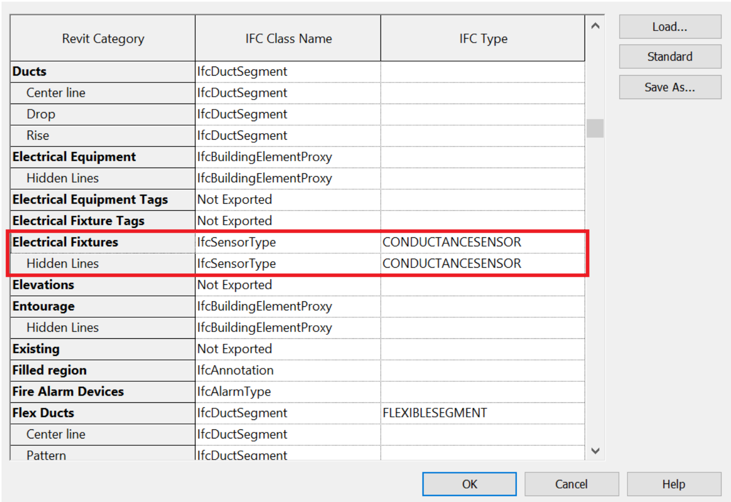
The Revit model was converted to IFC because it is a common data format to exchange building data in the AECO industry due to its compatibility with several BIM tools. In addition, the converter tool used requires input in this format. However, it is also possible to generate the RDF graph directly from the Revit file, for instance by using a tool like the revit-bot-exporter[30].

Some precautions were taken in the conversion process from Revit to IFC to keep the semantic logic of the building. They were the correct association in the floor level and the Revit-IFC mapping of the element's classes. The IFC version used was IFC4 supported by Revit 2018 IFC exporter, this version incorporates the lessons learned from the IFC 2x3 version bringing more consistency throughout its schema and reducing the footprint needed to populate the IFC dataset (BuildingSMART, 2018). The MVD used was the IFC4 Design Transfer View.

Regarding the association to the floor level, there are some differences between the modelling logic in Revit and the expected building configuration. Because the model represents the 8th and 9th floor of the Atlas building, when searching for an element, it is expected to find it by searching the content of one or another. However, working in an organized way in Revit requires several supporting levels (e.g. structural floor level, finished floor level, ceiling level) that can cause some confusion if exported to IFC and when using the application. The facility manager may guess correctly that the lighting fixtures are semantically associated with the ceiling level but a sensor, for instance, can be related to the ceiling level or to the floor level depending on whether it is a ceiling or wall-based instance. Thus, only the 9th floor and the 8th floor level were exported as building storeys in IFC.

The mapping from Revit categories to IFC classes can be done in the IFC options dialog box inside Revit. Verifying how the software is doing this mapping prevents elements with proper designated IFC classes to be exported as generic building element proxy that should only be used when IFC doesn't have a semantic definition for it yet. In this case, the mapping of the sensor was changed (Figure 17). Without a proper category in Revit, they were classified as electrical fixtures and mapped to the ifcSensorType class in IFC. This type has an enumeration of specific subtypes of sensors, the CONDUNCTANCESENSOR definition was used in this model as it was the closest related to the energy measurement feature used in the proof of concept. Additionally, the MVD was modified to include the Revit properties as property sets in the IFC, since they were used to input information in the model and are not part of the IFC common property sets.

[30] https://github.com/MadsHolten/revit-bot-exporter



| Revit Category | IFC Class Name | IFC Type |
|---|---|---|
| **Ducts** | IfcDuctSegment | |
| Center line | IfcDuctSegment | |
| Drop | IfcDuctSegment | |
| Rise | IfcDuctSegment | |
| **Electrical Equipment** | IfcBuildingElementProxy | |
| Hidden Lines | IfcBuildingElementProxy | |
| **Electrical Equipment Tags** | Not Exported | |
| **Electrical Fixture Tags** | Not Exported | |
| **Electrical Fixtures** | IfcSensorType | CONDUCTANCESENSOR |
| Hidden Lines | IfcSensorType | CONDUCTANCESENSOR |
| **Elevations** | Not Exported | |
| **Entourage** | IfcBuildingElementProxy | |
| Hidden Lines | IfcBuildingElementProxy | |
| **Existing** | Not Exported | |
| **Filled region** | IfcAnnotation | |
| **Fire Alarm Devices** | IfcAlarmType | |
| **Flex Ducts** | IfcDuctSegment | FLEXIBLESEGMENT |
| Center line | IfcDuctSegment | |
| Pattern | IfcDuctSegment | |



*Figure 17. Revit categories to IFC classes*

## 4.2 System architecture

The system architecture is the conceptual model of the prototype application. It describes the system organization and its expected behaviour. Moreover, it translates the use case scenario in requirements to be implemented in the system development.

### 4.2.1 System preconditions

For this proof of concept, three preconditions must exist. First, the IFC file needs to be converted to RDF. As discussed in the literature review, this study uses the IFCtoLBD[31] conversion tool which follows the LBD ontologies to perform this action. The second prerequisite is that this RDF file is stored in a web database, that allows the application API to communicate with it and make queries in SPARQL. As also mentioned in the literature review, Ontotext GraphDB was used as a graph database to store the RDF triples. Finally, the Atlas Living Lab server infrastructure needs to be collecting data from the building and be reachable through the web. This network is being developed by the Atlas Living Lab. The system prerequisites workflow is described in Figure 18.

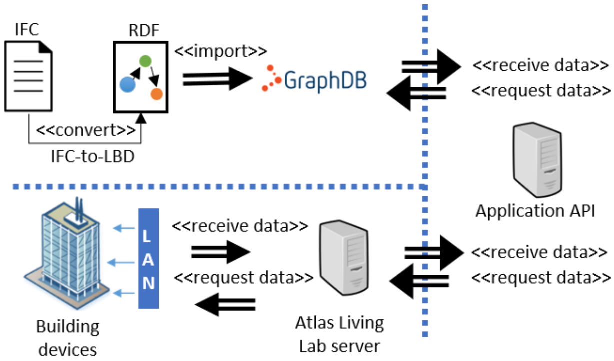



*Figure 18. System preconditions*

### 4.2.2 Application API

The overall system architecture UML activity diagram is depicted in Figure 19. The initial action is performed by the end-user by connecting to the application in a web browser (e.g. Google Chrome, Internet Explorer). This triggers the communication with GraphDB and the Atlas Living Lab server through an HTTP request of the method 'GET'. For the triplestore, the request URL (the address of the resources accessible over the internet or intranet)

[31] https://github.com/pipauwel/IFCtoLBD

includes a select query to return the objects and predicates associated with the building elements. For the lab server, it accesses the data stored in the server database by passing a username and password in the header of the HTTP request.

The data from the triplestore is displayed in a table format. The first set up of the application shows the information about the building element, extracted from the IFC file. When the user selects a different element in the table, the app sends a new HTTP request to the triplestore to get the data about it and update the user interface.

The sensor data is not displayed by default but only when a sensor is chosen. Hence, the component that handles the RDF data passes the information of what is being displayed at the moment to the root component and it sends this information to the component dealing with the sensor data. This component then evaluates if there is energy consumption data about the element selected, and in a positive scenario, builds the chart representation of the energy metric report on the user interface. In a negative scenario, no chart is created, or the previously created chart is removed from the screen. Additionally, the table component should allow the navigation back and forward inside the building hierarchy.

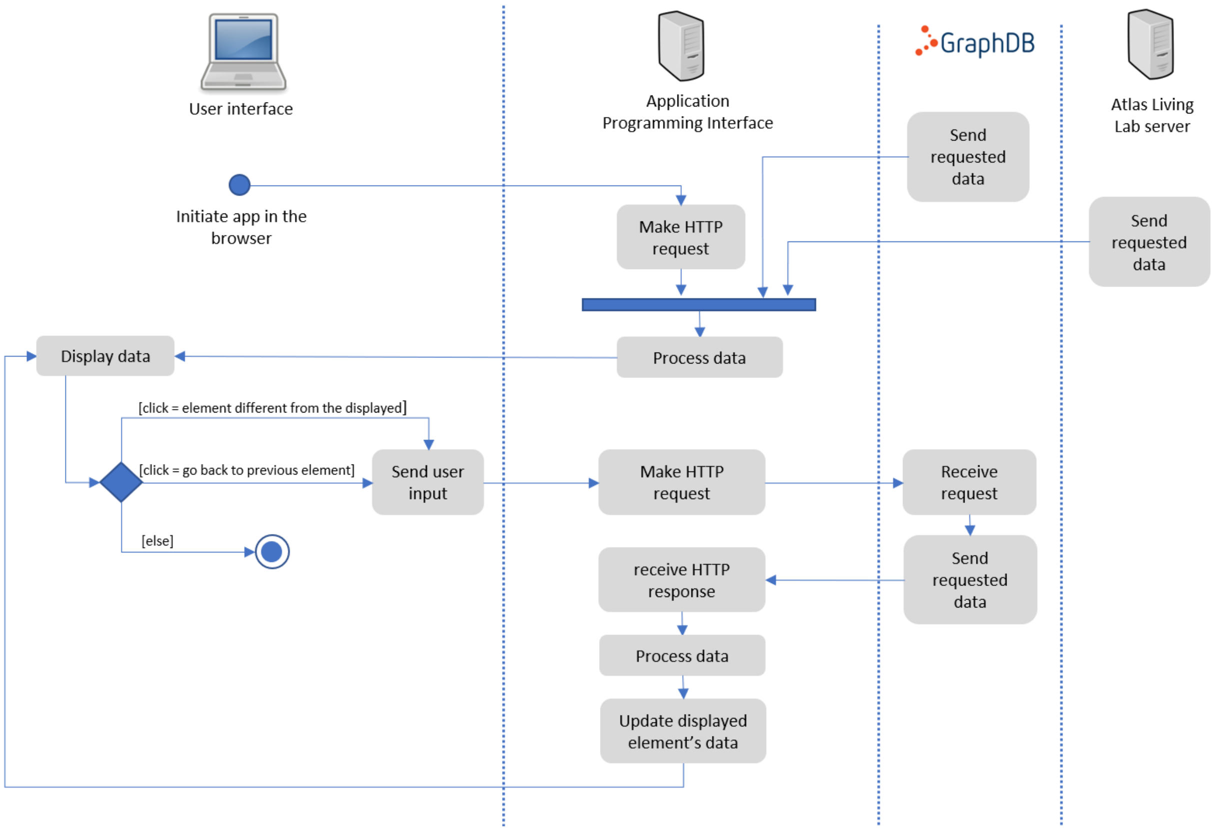



*Figure 19. System activity diagram*

## 4.3 Data collection and system architecture conclusion

This chapter focus on the two initial stages of prototype development. The data collection process resulted in the use case scenario for the application. It considered the luminaire based multi-sensors in Atlas and the BIM modelled for this study case. Thus, the app layout has to optimize the display of data from these two data sources and synchronize their appearance in the user interface. with that, the user should be able to monitor indoor environment sensors, rooms occupancy, and energy consumption associated with the sensors included in the model. In addition, get information about the building elements present in the triplestore. To connect these two data sources the approach was to model the sensor in Revit and include its identification from the BMS as an entity property.

The system architecture brings the functional requirements to enable the use case scenario and comply with the application requirements described in the literature. The app must not rely on the users' knowledge of SPARQL or display raw JSON and XML responses. Also, it has to allow its user to navigate the building data both forward and backward. Hence, the architecture defined supports intuitive queries by using HTTP requests to the triplestore every time a building element is selected by the user. Moreover, it processes the response received by both actors in formats like simple tables and charts, and has mechanisms to refresh the content displayed according to the user input.

## 5 Proof of concept: prototype development and evaluation

This chapter describes the steps taken to develop the prototype and how it was evaluated. The development stage includes the definition of the application layout, how the application retrieves data from the external sources, and how this data is transmitted within the application components, the main points in the data post-processing, and the mechanisms that allow the user interaction with the app. The complete application code can be found in https://github.com/Angelalima/Building-OM-center.

### 5.1 Prototype development

The prototype development implements the system architecture. This stage includes the creation of the user interface and the API that are the two parts that compose the full application. It also describes how it communicates with the endpoints of the external sources' APIs, namely the GraphDB database and the Atlas Living Lab server, and how the system API processes the data received to be displayed in the user interface.

#### 5.1.1 Application Layout

The first step in the implementation of the system architecture was to define the user interface layout. Although only two data sets were used in this proof of concept, the idea was to define it considering a fully developed application with placeholders for a future 3D viewer and spaces for other sensor data such as the temperature and occupancy status that are also monitored by the Atlas Living Lab server.

The app layout requirements are based on the conclusion from the data collection and system architecture, namely:

- Simple interface independent from the BIM authoring tool
- Display BIM data together with sensor data in a format simple to understand
- No technical knowledge needed to query the database
- 3D viewer and responsive design are desirable features

Thus, the final layout considered the information from the IFC file to be displayed in a table format besides a future 3D viewer canvas, and the sensor data in a chart format to be placed in the bottom part of the application to open space for other sensor readings. Each of these pieces was associated with a component to code its appearance and functionality considering the Angular framework, which was also used to communicate with the external sources APIs (Figure 20). Moreover, the user interaction with the app components happens through buttons to select elements or go back to the previous showing data, and a filter field to aid the localization of resources, all SPARQL queries are performed by the app API.

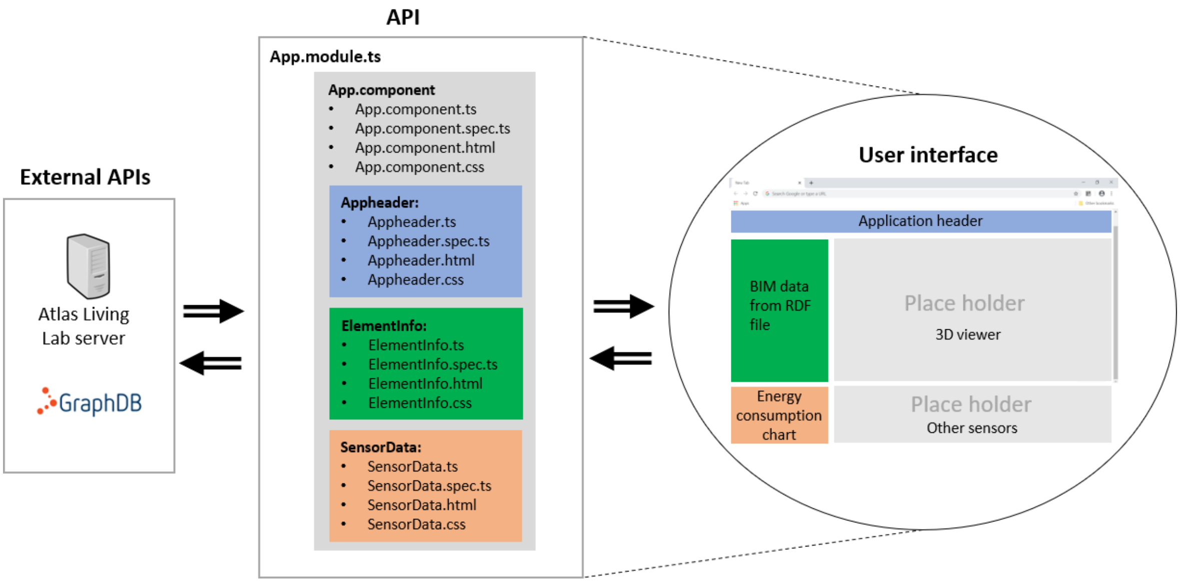



*Figure 20. User interface in the angular framework*

Additionally, the layout is responsive to smaller screens aiming to make the app easy to use on mobile devices. The modification in the layout for these devices was done coding a different set of CSS configurations when the screen width is around 40em or less (approximately 600px considering the standard layout).

### 5.1.2 Communication with external sources and internal components

The system API connects the client-side to the endpoints of the external sources APIs since these endpoints are where these servers store resources that can be accessed by third-party apps. This communication occurs through the HTTP protocol, a request-response system in which the prototype API sends the requests from the user to the databases and receives the response from the database to deliver it to the client. Angular provides a simplified HTTP API called *HttpClient*, this service class is part of the *HttpClientModule* that can be imported from the library *@angula/common/http* (Angular, n.d.-a).

Besides this module, Angular works with *observables* to wrap the response obtained from an HTTP request. The components of the application must *subscribe* to this *observable* to trigger the HTTP request, the observable then emits the data requested when a response is received to the ones that subscribed to receive it (Angular, n.d.-a). Observables and observables' operators can be imported from the *rxjs* and *rxjs/operators* libraries. They make it possible to handle asynchronous requests, when the action completes at a later cycle then it started or never completes. For instance, when an API requests data from sensors it will not get it immediately but will rather receive a stream of data as they collect building information throughout the time unit of collection. The *observable* wraps this stream of data and keeps listening and transmitting the information.

HTTP requests were implemented using the Angular infrastructure to allow the application to communicate with both external data sources. However, there were some variations between the two cases which are discussed in the topics below.

#### 5.1.2.1 Communication with graphDB

The *ElementInfo* component is responsible for handling the table in the user interface that displays the triples from GraphDB. However, this component needs to interact with the *sensorData* component since the use case scenario defines that the data displayed in the table triggers actions in it. Therefore, the app needed to set up the communication not only with the external database to populate the table but also needed to inform the other parts of the application what is being displayed from this database.

The interaction of components in Angular commonly happens through *services* or *input/output* decorators. The latter is applicable when there is a parent/child relationship between the components, meaning that one component (child) is contained in another (parent). On the other hand, *HttpClient* used to interact with the external API's endpoints can be injected in services or imported in components, once that *HttpClientModule* is imported to the root *appModule*.

In this thesis, the approach used was to handle the communication with graphDB in the ElementInfo component (Figure 21). When the application is first compiled, the component makes the HTTP request to query the database for the building element, which is a higher hierarchy in the building navigation and the initial value in the user interface. This request is triggered when it subscribes to this observable, and when the data arrives, it is stored in a global variable named *displayedData*. When the user changes the table, the information is passed to the parent component using the *@output* decorator. Since all the components are children of the AppComponent, it can send this output from the table as an input to the *sensorData* component.

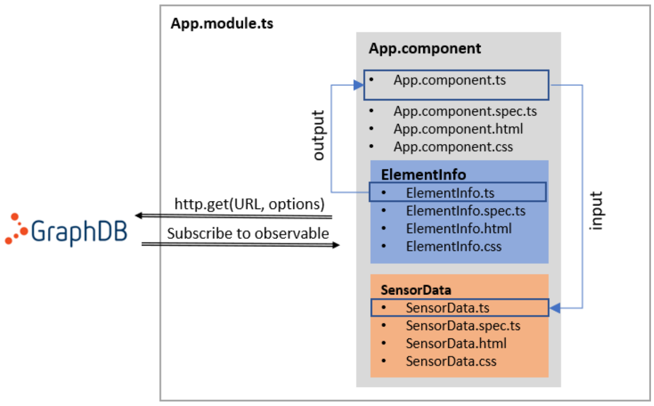



*Figure 21. Communication flow between GraphDB and the ElementInfo component*

The HTTP request setup in the component starts by importing the elements from the *@angular/common/http* library. Then, *HttpClient* is injected as a dependency type in the constructor. Constructor is a method called when a class is instantiated, and a dependency is an external service or object that a class calls to perform its functions (Angular, n.d.-b).

The first HTTP request to get the default data on display is implemented in the *ngOnInit()* lifecycle hook that is called when the application is initiated in the browser. The other queries the user might make while navigating in the app are implemented by the *updateData()* function (Figure 22). In both cases, the HTTP method used is GET, the one adequate to read data from the data source. GET only takes two arguments: the URL of the resource requested and an options object, since the *http.get()* method has no body. The *options* argument passes information about the request configuration. In the application, the *options* object was used to pass the HttpHeaders and the HttpParams.

```
updateData (element) {
  const httpParams = new HttpParams ({
      fromObject:{
        query:'select ?o ?p where {<' + element.o.value + '> ?p ?o}'
      }
    });

 this._http.get('http://localhost:7200/repositories/05', {params: httpParams,

  headers: new HttpHeaders()
        .set('Content-Type', 'application/x-www-form-urlencoded')
        .append('Access-Control-Allow-Methods', 'GET')
        .append('Access-Control-Allow-Origin', '*')
        .append('Access-Control-Allow-Headers', "Access-Control-Allow-
Headers, Access-Control-Allow-Origin, Access-Control-Request-Method"),
      responseType: 'text'
})
.subscribe((data) => {
  this.displayedData = (JSON.parse(data)).results.bindings;
  this.outputoParent.emit(this.displayedData);
});

this.arr.push(this.displayedData)
}
```

*Figure 22. UpdateData() function*

When the user selects an element in the table, the value selected is included as the subject of the SPARQL query that is used as a parameter in the resource URL. An example of a query generated by this method is:

http://localhost:7200/repositories/05?query=select ?o ?p where {<http://linkedbuildingdata.net/ifc/resources20200408_172328/storey_147> ?p ?o}

the *http://localhost:7200/repositories/05* refers to the location of the repository and the repository ID (05). The '?' marks the start of a parameter. In this case, the *query* parameter has as a value a SPARQL query. The select clause identifies the variables to appear in the results, in this case, *object* (o?) and *predicate* (?p). the *where* clause defines the pattern to look for in the data, here triples that have as subject *storey_147*. Thus, the expected result from this HTTP request is all the objects and predicates that have as subject the 8th floor storey.

The last part of the code shows when the component subscribes to the observable to get the data from GraphDB and the response is parsed to JSON using the JSON.parse() method (Figure 23). JSON data works with *name: value* pairs, hence it is possible to access specific parts of the content using its name. In this case, *bindings* inside *results* is the name of the arrays of properties to be displayed to the end-user.


```
{
"head": {"vars": [ "o", "p"]},
"results": {"bindings": [
  {
"p": {"type": "uri", "value": "http://www.w3.org/1999/02/22-rdf-syntax-ns#type"}, "o": {"type": "uri", "value": "https://w3id.org/bot#Space"}
  },
    ]
      }
```


*Figure 23. Part of GraphDB's API response*

When the user changes the content displayed in the element information table. The *@output* decorator is used to emit an event for the *ElementInfo* component to the *app.component* with the new values to be stored in a variable that is later transmitted to the *sensorData* component using the @input decorator.

#### 5.1.2.2 Communication with Atlas Living Lab server

The lab server was not operational during the time this application was developed, but the structure for this connection was coded with the URL of the living lab server endpoint replaced by the asset folder path where the sample file of the sensor data was placed. The function created to implement this connection was *loadXML()*.

The response from the lab server is in XML, thus the request needs to accept this data type. The response type was set to text and later on parsed using the DOMparser() method that deals with XML and HTML as Document Object Models (DOM). The HTTP method is also GET (Figure 24).

```
  loadXML() {
    this._http.get('/assets/response_POST_getMetricReport_Energy_Hour.xml',
  {
    headers: new HttpHeaders()
          .set('Content-Type', 'text/xml')
          .append('Access-Control-Allow-Methods', 'GET')
          .append('Access-Control-Allow-Origin', '*')
          .append('Access-Control-Allow-Headers',        "Access-Control-Allow-
Headers, Access-Control-Allow-Origin, Access-Control-Request-Method"),
        responseType: 'text'
  })
  .subscribe((data) => {
    var parser = new DOMParser();
```

```
    this.xmlItems=parser.parseFromString (data, 'text/xml');
    this.getSensorID(this.xmlItems);
    this.buildArray(this.xmlItems);
    });
};
```

*Figure 24. HTTP request and subscription for the sensor data*

As mentioned, the *displayedData* value is passed to the *sensordata* from the *app.component* with the *@input* decorator. Every time there is an update in its value, the code checks to verify if the 'sensorID' property in the BIM data match the 'LocationID' from the BMS, in positive cases, it builds the sensor data chart. That is enabled by the *ngOnChange()* lifecycle hook method (Figure 25).

```
ngOnChanges() {
 if (this.displayedData !== undefined) {
  for (let i=0; i<this.displayedData.length; i++) {
    if (this.displayedData[i].p.value =='https://w3id.org/props#sensorID')
    {this.sensorID = this.displayedData[i].o.value;
     if (this.sensorID == this.locationID)
   {this.makeGraph(this.arr_fromDateTime,
    this.arr_toDateTime, this.arr_avgKW, this.arr_endAccKWH,this.arr_maxKW,
    this.arr_minKW, this.arr_startAccKWH)}}
    else if (this.chart !== undefined) {this.cleanChart()}
   }}
}
```

*Figure 25. SensorID evaluation*

### 5.1.3 Data Processing

The data retrieved from external sources are in JSON and XML. These are the most common data formats exchanged by APIs but using them in raw format is not very human friendly. To achieve the proposed application layout, this data needs to be post-processed. Different approaches were used for each file format.

#### 5.1.3.1 Table component: JSON

The angular material table was used to make the basic structure of this component. Angular material[32] is a library of pre-formatted components that are customizable for a fast user interface design. In this table structure, the first column was named *Property* and was associated with the predicate part of the RDF triple, the second one was named *Value* and displays the object part of the triple (Figure 26).

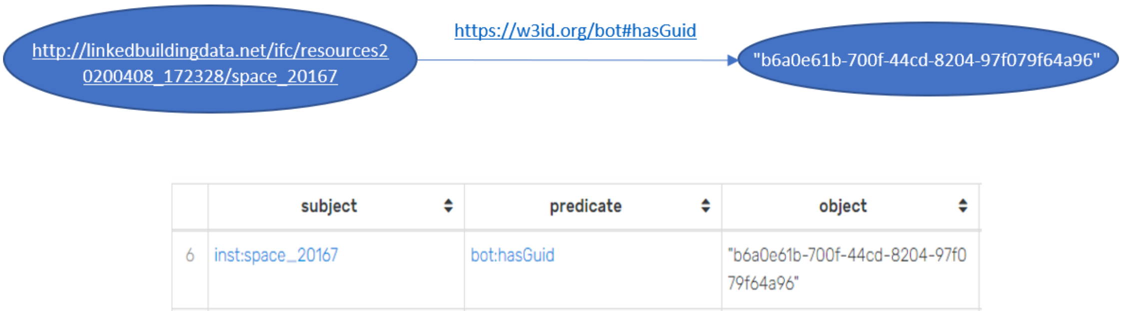



*Figure 26. Adapted data representation in GraphDB*

[32] https://material.angular.io/

Figure 23 shows how the response from graphDB arrives in the prototype application. The objects and predicates content can be accessed using *p.value* and *o.value* due to it being in JSON syntax and the split() and pop() methods were used to separate the prefix, with the URI that identifies the resource, from the resource itself, what made it more human-readable (Figure 27). For the example above, the application property column would show hasGuid and the value column "b6a0e61b-700f-44cd-8204-97f079f64a96".

```
<table mat-table [dataSource]="displayedData | triplefilter">
<!--Property Column -->
<ng-container matColumnDef="property">
<th mat-header-cell *matHeaderCellDef> Property </th>
<td matcell*matCellDef="let element">{{element.p.value.split("#").pop()}}</td>
</ng-container>
<!-- Value Column-->
<ng-container matColumnDef="value">
<th mat-header-cell *matHeaderCellDef> Value </th>
<td mat-cell *matCellDef="let element">{{element.o.value.split("/").pop()}}
</td>
</ng-container>
<tr mat-header-row *matHeaderRowDef="displayedColumns; sticky: true"></tr>
<tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr>
</table>
```

*Figure 27. Table component first part coded in HTML*

Moreover, this table component was divided into two parts to facilitate the navigation between the building elements. The first part of the table shows the information about the component selected, and the second part has the elements contained in the selected element. To enable this dynamic, the first challenge was to separate the relevant properties for each part of the table from the whole set of properties associated with the building element. For that, the chosen approach was to use customized *pipes*.

The angular framework has *pipes* (coded as |) that can be customized and used to transform data. The *Tripefilter* and *containfilter* were classes created with the *@pipe* decorator. They have inside them a function that takes as parameter the *displayedData p.value* and verifies if the value is among the ones to be included in the table or not using logical comparisons.

In the *tripefilter*, the resources included came from the IFC common property sets and the Revit property sets that should not appear on the table because no value was attributed to them in the design phase or because the information in them is related to the Revit modelling environment. Hence, the logic was to compare the predicate values with the defined resource using the operator !== (not equal and not the same type) and && (and), therefore only when different from all the resources identified it was included in the first part of the table.

The *containfilter* applied this reversed logic, the predicates were compared with constainsElement, hasStorey, hasSpace, and hasSubElement using the == (equal to) and the || (or) operators, hence if the predicate had any of the containment relationships, it was included in the second part of the table (Figure 28).

```
import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'containfilter'
})
export class ContainfilterPipe implements PipeTransform {

  transform(datab: any): any {
    const term = [
```

```
      "https://w3id.org/bot#containsElement",
      "https://w3id.org/bot#hasStorey",
      "https://w3id.org/bot#hasSpace",
       "https://w3id.org/bot#hasSubElement",
     ];
return datab.filter(function(f) {
return (f.p.value == term[1] || f.p.value == term[0] || f.p.value == term[2]
|| f.p.value == term[3]);
    });
  }
}
```

*Figure 28. ContainfilterPipe Class*

5.1.3.2 Sensor data component: XML

The layout of this application aimed at displaying the sensor data in a graphic form to be more easily readable for the facility manager than the raw XML report. Chart.js[33] is the open-source JavaScript library that was used in this prototype. It enables the creation of several charts' types in the HTML5 <canvas> tag. The XML data needed to be simplified in an array of simple numbers and dates to be used as the x and y axis of a line chart.

This extraction of simple values happened in several steps after parsing the file as DOM. First, a specific variable for each dataset in the report was created to store the XML tags using the *getElementByTagName()* method. Then a *for* loop was used to look into these tags and get the value between them. The values that represented a date were changed to this data type using the *Date()* method. The last step was to push these values to an array, values that represented numbers were converted to this data type using the + sign in front of it. The date types were further formatted to have a two-digits hours and minutes since the original date value in the XML report is the complete description of the date (e.g. 2020-04-01T06:00:00Z). This process is described in Figure 29. A similar data process is used to extract the 'locationID' property from the sensor report (Figure 30).

```
  buildArray(x) {

    let fromDateTimes = x.getElementsByTagName('fromDateTime');
    let toDateTimes = x.getElementsByTagName('toDateTime');
    let avgKWs = x.getElementsByTagName('avgKW');
    let endAccKWHs = x.getElementsByTagName('endAccKWH');
    let maxKWs = x.getElementsByTagName('maxKW');
    let minKWs = x.getElementsByTagName('minKW');
    let startAccKWHs = x.getElementsByTagName('startAccKWH');

    for (let i=0; i<fromDateTimes.length; i++) {
      var fromDateTime = new Date (fromDateTimes[i].innerHTML);
      var toDateTime = new Date(toDateTimes[i].innerHTML);
      var avgKW = avgKWs[i].innerHTML;
      var endAccKWH = endAccKWHs[i].innerHTML;
      var maxKW = maxKWs[i].innerHTML;
      var minKW = minKWs[i].innerHTML;
      var startAccKWH = startAccKWHs[i].innerHTML;

     this.arr_fromDateTime.push(
```

[33] https://www.chartjs.org/

```
        fromDateTime.toLocaleTimeString('nl-NL', {hour: '2-digit', minute:'2-
digit'})
      );
      this.arr_toDateTime.push(
        toDateTime.toLocaleTimeString('nl-NL', {hour: '2-digit', minute:'2-
digit'})
      );
     this.arr_avgKW.push(
        +avgKW
      );
    this.arr_endAccKWH.push(
        +endAccKWH
      );
     this.arr_maxKW.push(
        +maxKW
      );
     this.arr_minKW.push(+minKW);
     this.arr_startAccKWH.push(
        +startAccKWH
      );
    }
```

*Figure 29. Function buildArray() that format the XML file in a usable dataset for Chart.js*

```
  getSensorID(x) {
    let LocationID = x.getElementsByTagName('locationID');
    for (let i=0; i<LocationID.length; i++) {
       this.locationID = LocationID[i].innerHTML;
  }
};
```

*Figure 30. Function getSensorID()*

### 5.1.4 Data display and user interaction

Figure 31 brings the resulting table component in the user interface. The first part of the elementInfo component has the type property of the element from the RDF file formatted to appear as the title of what is currently in the display. Besides it, there is the 'go back' button. When selected, it renders the element with a higher hierarchy then the element in display. For example, if a lighting fixture is showing and the user selects to go back, it will show the room where this element is located, if the back button is selected again, then the floor where this room is located appears. Hence, it allows back navigation in the building.

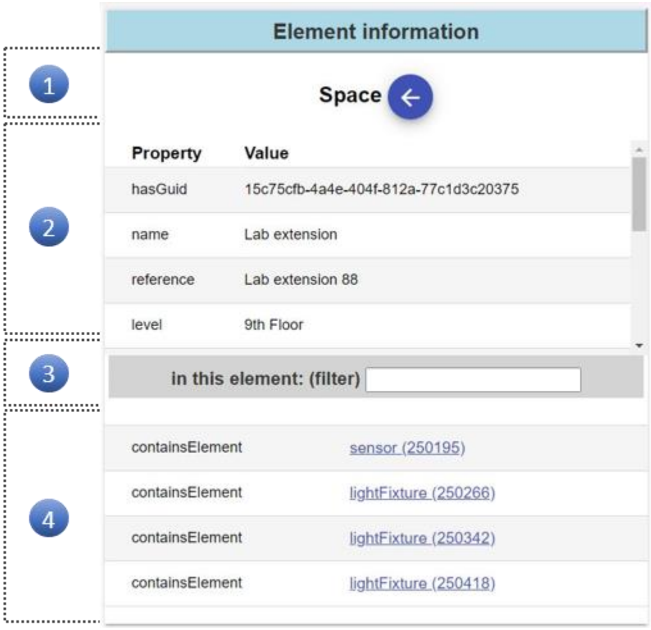



*Figure 31. Table component*

Behind the scenes, the user's click activates the *goBack()* function (Figure 32). In Figure 22, the last line of the code indicates that the data gathered from the database is pushed to the *arr* variable. This is an array of objects that is used to allow the back navigation. When the user selects to go back, the element on display becomes the penultimate registered. Then the last value in this array is eliminated using the splice() method, to allow the user to go back several times. Finally, this update is outputted to the *app.component* and transmitted to the other parts of the application.

```
goBack (displayedData) {
  this.displayedData = this.arr[(this.arr.length)-1];
  this.arr.splice((this.arr.length)-1,1);
  this.outputoParent.emit(this.displayedData);
}
```

*Figure 32. Goback() function*

The second part of the component brings the formatted properties of the elements except for the ones filtered by the triplefilter, as described in the data process topic. The third-part of the table is a filter pipe created to make navigation easier to the facility manager when dealing with building parts with a lot of elements contained in it, like building floors. The filter takes the text inputted by the user and compares it to the value column to include only the elements that have at least a part of what was written in their description. To prevent errors the pipe converts both what was written in the filter field and the contained elements names to lowercase before comparing them (Figure 33).

```
import { Pipe, PipeTransform } from '@angular/core';
@Pipe({
  name: 'filterpipe'
})
export class FilterpipePipe implements PipeTransform {
  transform(displayedData: any, term: any): unknown {
    //check if search is undefined
    if(term === undefined) return displayedData;
    //return updated data
    return displayedData.filter(function(f){
```

```
      return f.o.value.toLowerCase().includes(term.toLowerCase());
    })
  }}
```

*Figure 33. Filter pipe code*

In the bottom part of the table, the elements contained in the element on display are separated from the other properties using the *containFilter* pipe. The data displayed in the value column were wrapped as a button and their appearance aimed at signalling to the user that they are clickable elements. When the user selects one of those elements, behind the scenes the application uses the *updateData()* function as described in the 5.1.2.1 .

This change triggers other actions through the application. As mentioned, the output mechanism is activated and sends the new element data back to the *app.component* which in turn inputs it in the sensor data component and activates its *ngOnChange()* lifecycle hook. For the sensor (250195), the *sensorData* component creates the chart in Figure 34. In this chart, the user can pass the mouse at a point associated with a time of the day and get a better visualization of the energy values in that period. When the data displayed has no sensor data associated with it, the component activates the *cleanChart()* function that uses the *destroy()* method to clean the canvas to a new graph.

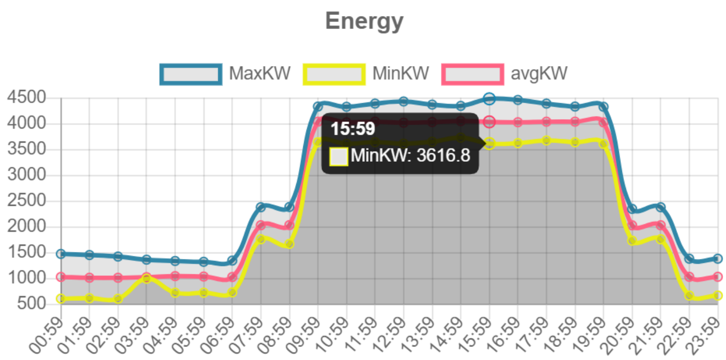



*Figure 34. Energy metric report in graphic representation*

## 5.2 Evaluation

The prototype was evaluated regarding its ability to function as designed through an alfa testing phase. In other words, the researcher assessed whether she could retrieve the correct information for a specific element, both from the information embedded in the IFC and the sample data from the energy metric report. This includes verifying if the setup mechanisms for user interaction worked as programmed and if the RDF and XML files were formatted and filtered as specified in the code.

Moreover, the datasets used in the application were evaluated regarding information loss and semantic inconsistencies when going from one data type to another. The output RDF from the IFCtoLBD converter is compared with the IFC file, and the IFC file is compared with the original Revit file.

Finally, the application is assessed regarding its limitations when applying it to different buildings and its scalability possibilities to include other datasets in the app layout.

## 5.3 Prototype development and evaluation conclusion

This chapter describes how the prototype was developed and tested. The app layout targets the requirements defined in the system architecture based on the data collection. It resulted in a simple interface to display BIM data in an interactive table and the sensor data in chart format associated with the sensor element in the model. All technical knowledge is handled in by the code of each component in the Angular framework. In addition, the layout considered components for the future inclusion of a 3D viewer and other sensor measurements, and responsive design.

The *ElementInfo* component handles the communication with GraphDB, processes the JSON response using customizable pipe filters, and displays it in the user interface using an Angular Material table. The database is queried using a dynamic HTTP parameter with the selected element and the displayed data is transferred to the *sensoData* component using input/output decorators. The sensorData component gets and parse response in XML to a simple array of values. They are used to build the sensor chart using the Chart.js library. The connection between sensor data and sensor element in the model is made by evaluating if the sensorID property exists and matches the locationID in the BMS data.

The mechanisms coded were critically analyzed regarding its ability to retrieve, process, and display data from the two data sources and its usability for other use case scenarios. Moreover, the data transformation of the BIM data was evaluated to disclose data loss and semantic problems.

# 6 Study case results

This chapter concludes the research methodology by showing and evaluating the results obtained in both study cases.

## 6.1 ZEL study case evaluation

The analysis of the ZEL study case is fully described in the **Appendix**. In this topic, the main results are presented and discussed. The evaluation starts with the documentation that guided the modelling process and determined the design deliverables. It includes the BIM protocol and its information delivery specification that were developed specifically for the design phase.

The first relevant aspect is the existence of written requirements, obligations, and liabilities for the stakeholders in the process since the lack of clear and documented requirements for BIM modelling is still an issue found in the literature. These documents reflect the standard scope of work of the design firm. For instance, general and specific properties were defined for the building elements represented in the architectural design, but no similar agreement is in place for the MEP systems yet. Hence, further discussion is needed to define the client requirements based on the facility managers and contractors' needs for the model delivered in the handover stage.

This as-built model needs to reflect the maintenance and operational procedures it is going to support. Some examples of O&M requirements are more detailed information about parts of an equipment or system, the inclusion of classification systems typically used by the facility managers, or the addition of zones and spaces in MEP models to aggregate data about the systems heat and cooling loads. Moreover, an internal combustion engine laboratory has systems like fuel pipes, complex ventilation systems, liquid-tight floors, and special fire protection systems that will have specific modelling and information requirements for this stage of the building lifecycle.

The second part of the evaluation focused on the BIM models to identify points to be improved for the final model. They presented some of the modelling problems described in the literature. Regarding the data embedded in the building elements, problems with property missing or extra information added beyond agreed were found, considering what was described in the documentation. Some examples are the classification present in both models with values referring to the Uniformat classification standard when the documentation states that the classification system used was the Dutch STABU code, both information exist in several building elements. Properties missing were the most recurrent issue, assembly code, description, and material properties were with blank values in several elements although they are declared to be common properties to all building components. Moreover, the numbering system of the rooms in the model have no systematic approach and are not unique per element.

Another modelling problem is the intersection of building elements geometry and in one case a duplicated element (Figure 35). Beyond the geometric representation inaccuracy, this can cause errors when extracting quantities from the model, for instance, to calculate the material necessary to execute a maintenance procedure. However, in this study case, they are not enough to create a big discrepancy. The modelling problems found in the ZEL study case are probably related to human error and the reuse of Revit families that already had some information embedded in it. The random numbering system of the spaces in the model might be related to the lack of a definition from the client about how their classification should be in this initial phase, considering there is no mention to that in the documentation. The lack of a consistent system to name and number spaces/rooms makes it difficult for the facility manager to fast identify where the room is in the building. Moreover, building management systems generally have a classification system and because of that without a coherent classification in the BIM data, it is harder to correlate the data from one to the other.

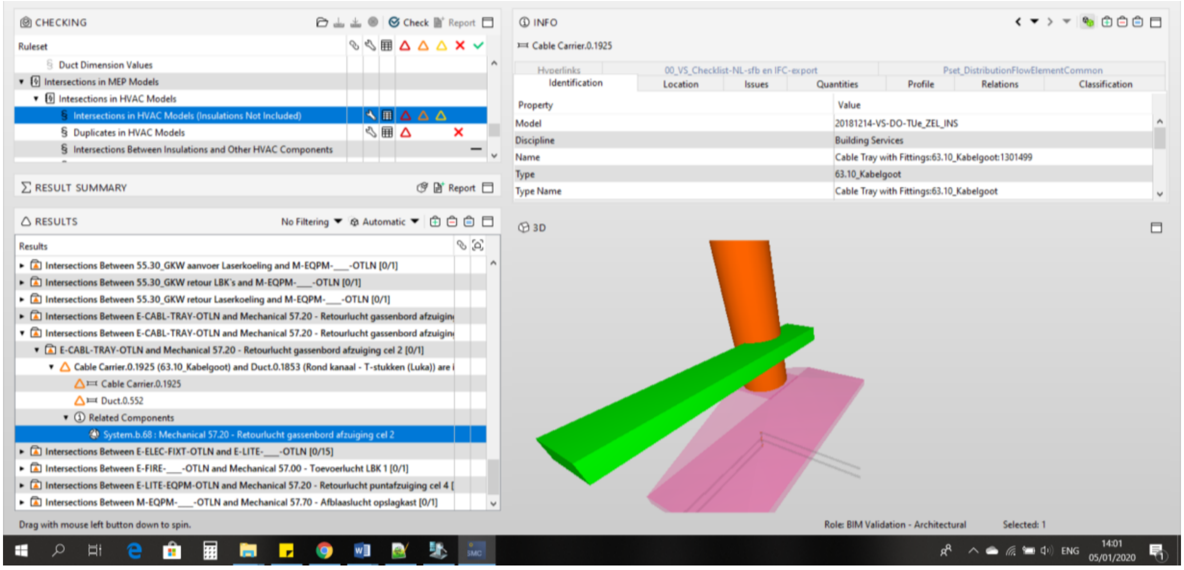


*Figure 35. Intersection between pipe and cable carrier*

Furthermore, the BIM models presented some semantic issues that need to be resolved before using these models in the prototype application. There are several relationship issues in the models. The main one is the fact that the spaces entities were only modelled in the architecture model, thus there is no association between the MEP systems and the rooms where they belong to. Additionally, there are elements with a semantic context that were misrepresented in the model. For instance, Figure 36 shows a wall associated with the -1 fundering floor and Figure 37 a door hosted by this wall associated with the 00 begane grond floor. Another example is that most of the stairs in the architectural model are composed of independent stairsteps with no relationships between them. Moreover, elements that are common in both models (site, building, and building floors) are different entities with different GUID and the top two floors are even named differently, 01 laag dak and 02 hoog dak in the architecture model, and 01 eerste verdieping and 02 tweede verdieping in the MEP model. In the MEP model, the mechanic and plumbing components are organized as actual systems, but the electrical elements are partly grouped and part individual elements with no semantic relationship between them.

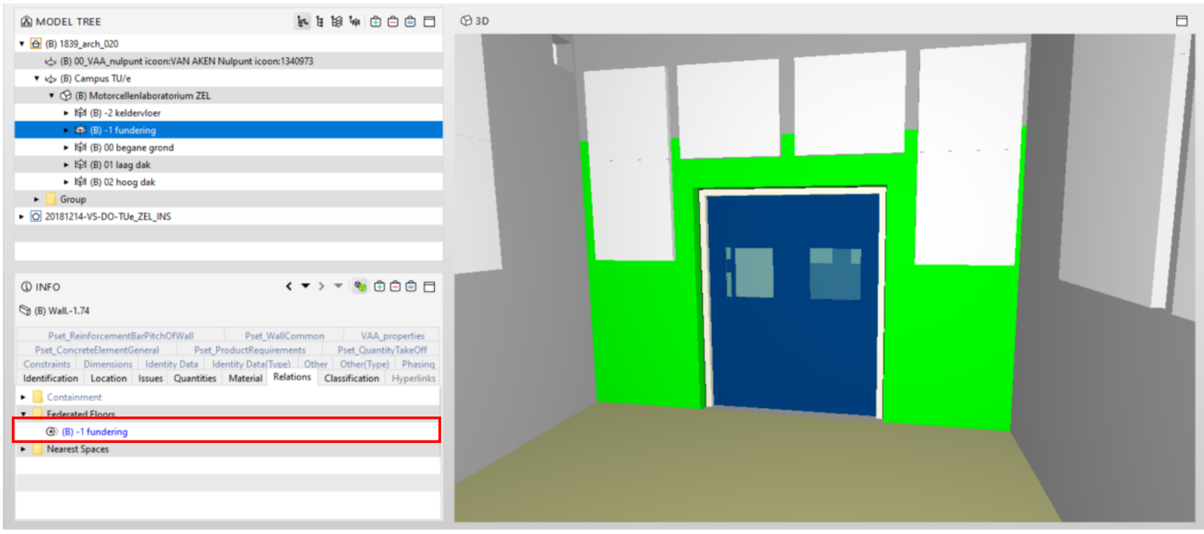


*Figure 36. Element relationship to the -1 fundering floor*

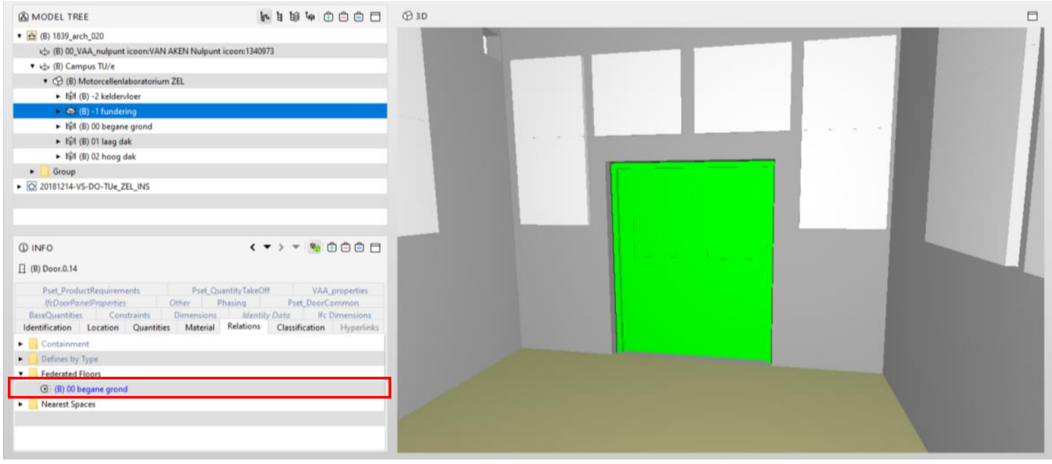


Figure 37. Element relationship to the 00 begane grond

If included in the prototype app, these issues would generate a navigation problem. In the first case, all MEP elements would be associated with the building floor hence the user would not be able to select a room to get the information, for instance, about the lighting and sensors on it. In the second case, the information given by the *elementInfo* table would state wrongly that the wall and door are not on the same floor, and in the third case it would generate a lot of different elements contained in the floor level instead of one stair entity. In addition, if the common entities are not correlated to each other when the IFC is converted to RDF, the app will display the information about the elements as if they came from two different buildings, and relationships between architecture elements and MEP systems parts would not appear.

The problems found in this analysis corroborate the findings in the literature research like elements missing, incorrect semantic definition and spatial associations, and inconsistent naming conventions and classification systems. On the other hand, they can be easily solved with adjustment in the model. Considering the scope of the use case scenario, two main lessons from the first study case were to integrate the spaces naming and numbering system with the requirements from the BMS and to match the semantic relationships from the building when modelling the BIM model.

## 6.2 Application evaluation

The final user interface can be seen in Figure 38 and Figure 39. The placeholders allowed the creation of a CSS grid layout that took into consideration the fully developed application. For smaller screens, the horizontal grid changes automatically to a vertical layout (Figure 40). This change was developed using the Chrome DevTools[34] Device Mode that enables to simulate the user experience when using the browser in different devices.


Figure 38. Application user interface top part

[34] https://developers.google.com/web/tools/chrome-devtools

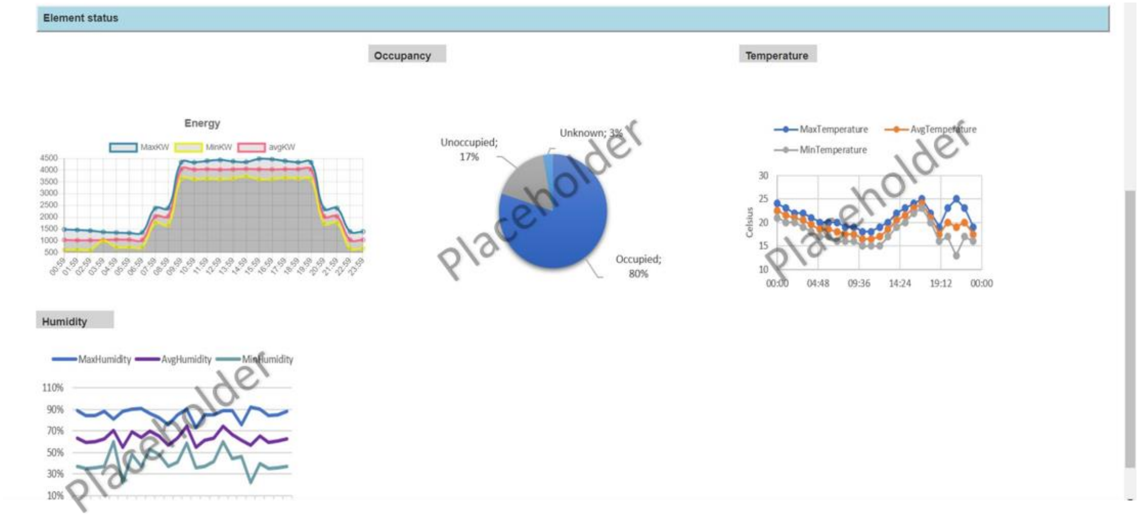



*Figure 39. Application user interface bottom part*


*Figure 40. Responsive layout for portable devices*

The application behaviour and data display worked as programmed when used by the researcher in a Google Chrome browser with no errors register in the console considering that the application is running locally. Furthermore, the table component was evaluated comparing the data displayed with the data available in the database taking into consideration the filters and the data processing coded in the app and it worked as designed. Figure 41 shows a partial example of a room element. The type property was filtered out of the first part of the table and included as a title. The *hasGuid* was included in the element information table. The *containsElements* predicates were removed from this first part and added to the second part due to the combination of the *triplefilter* and the *containfilter* pipes, and the lighting and power properties that don't have a relevant value for this case were filtered out and are not displayed. Moreover, the data transformation worked to make it more readable and the filter field functioned to retrieve all elements that contained a part of what was typed in it.

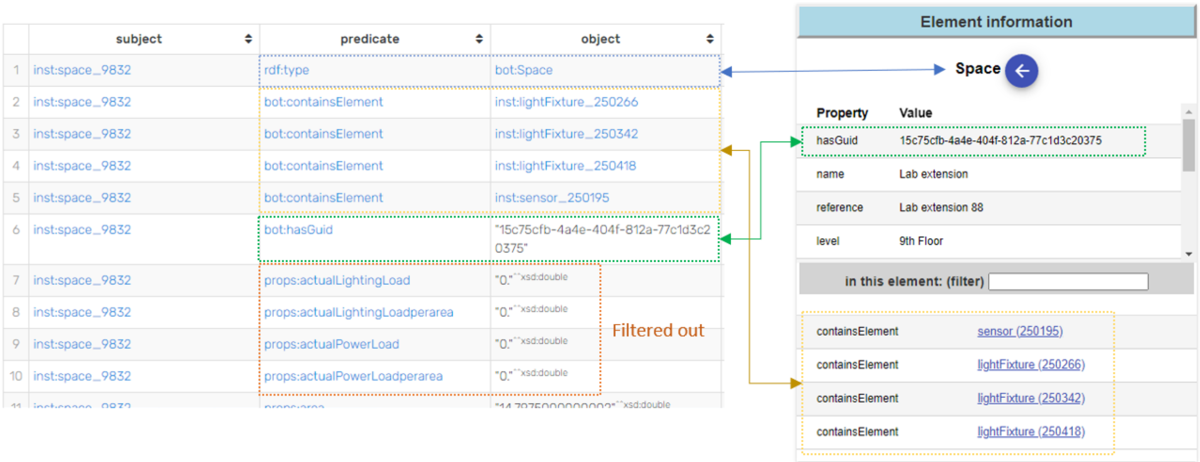



*Figure 41. Data processing of RDF graph*

This evaluation was done for the building, the 8th and 9th building storeys, space (9832), and the sensor and lighting fixtures in it. This scope was related to the sample data available from the energy metric report. Furthermore, even though the table component worked as coded there is a potential issue with the approach taken. Because the app uses the predicate values to determine what to be displayed and where, object values that are different but share the same predicate will be included or excluded in the same way. It happened for the sensor and the lighting fixtures, for instance, the *rdf:type* predicate is shared by the *bot:element* and the *mep:Sensor__CONDUCTANCESENSOR* objects, and both are filtered out from the table and only the first is used as the data title (Figure 42). For this study case the information erased, *mep:Sensor__CONDUCTANCESENSOR* and *mep:LightFixture*, can be found in other properties displayed except for the sensor subtype information (conductance sensor) that was lost in this process. Alternatively, this property could be included twice one as the table title and another in the property/value table, this would avoid this minor data loss.

| | subject | predicate | object |
|---|---|---|---|
| 1 | inst:sensor_250195 | rdf:type | mep:Sensor__CONDUCTANCESENSOR |
| 2 | inst:sensor_250195 | rdf:type | bot:Element |

*Figure 42. Objects sharing the same predicate*

The chart component was analysed regarding the create and destroy routines coded, the content displayed, if it matched the XML sample data file, and the data processing results. Figure 43 illustrates this evaluation and shows the link made between the sensor data and the BIM data by including the sensor reference as a property in the BIM model. The chart helps the user to better visualize the data from the report generated by the BMS and is a close enough approximation of the reality. The sensor measurement is per interval of one hour, and the chart associates average, maximum, and minimum energy consumption to one point in this interval, the final timestamp of the measurement.

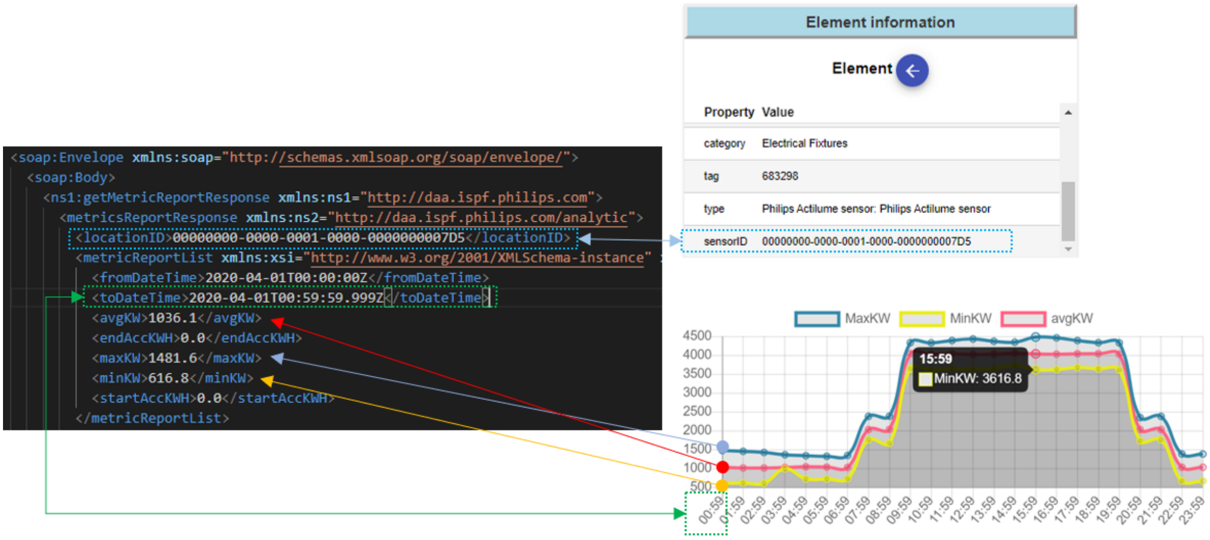



*Figure 43. Data processing of XML file*

Overall, the application fulfils the use case scenario requirements established for the application and was useful to study how to display different datasets in a web application in a user-friendly way. In addition, the developed layout is reusable, and new components can be added to the available CSS grid, which can also be expanded, without affecting the existing parts of the application.

Some small adjustments need to be made to change the building on display. The URL in the HTTP request needs to be changed to the GraphDB location in the computer of the application user. Moreover, for the first HTTP request, the subject part of the triple in the select request needs to be changed to the new building to be explored. For the sensor data, options for automation need to be investigated when the Atlas living lab starts collecting live data again. Right now, the sensorID predicate is being used to identify the sensor and link it to its BMS data. This property needs to be added to the BIM model or directly in the triplestore. Finally, the *triplefilter* pipe will work as well with a different building that uses the same ontologies applied in this study to generate the RDF file, but it should be re-evaluated to reflect the user needs since it was tailored to this use case scenario.

## 6.3 Data conversion evaluation

The resulting conversion from Revit to IFC has two main remarks. The first regards the mapping process between the Revit categories and the IFC classes. Revit doesn't have a specific category for sensor devices. Therefore, in this study, they were modelled as electrical fixtures. IFC, on the other hand, has specific classes to deal with sensors, namely, IfcSensor, IfcSensorType, and IfcSensorTypeEnum. The last two are used to specify the subtype of the sensor, for instance, if it is a $CO^2$ detection device or a fire detection device. Hence, the electrical fixture category was mapped to the IfcSensor class with a type conductance sensor.

For this use case, this didn't have any further implications since the sensors were the only electrical fixtures included in the model. For a bigger scope, this would generate a semantic conflict since all electrical fixtures would be interpreted as sensors by the IFC converter. There is no perfect way to correct this issue. One approach could be to map the electrical fixtures to the IfcDistributionControlElement class, a more generic class that keeps the semantic correct since the IfcSensor class is a subtype of it. In this case, all electrical fixtures would be considered distribution control elements of type undefined, since declaring a type would generate the same semantic inaccuracy of the beginning.

The second is related to the information from Revit that was exported to IFC. The exporting settings were defined to include both property sets from Revit and the common IFC property sets (Figure 44). This was necessary to include the *Data* property set from Revit that has the *SensorID* and the *LightID* properties. However, this added a lot of unused information to the IFC file and some duplication of information, for instance, the same instance reference information is present in several property sets from Revit and IFC. An alternative approach could be to make use of the 'Export user defined property set' option in the exporting settings and create a specific Revit property set just for what needs to be exported. This approach can take some time to determine what are the information from Revit that is useful and what is not per instance or type. Moreover, This conversion step can also be eliminated by converting the Revit file directly to IFC, but this requires different system preconditions that will need to be adapted when using other BIM authoring tools.

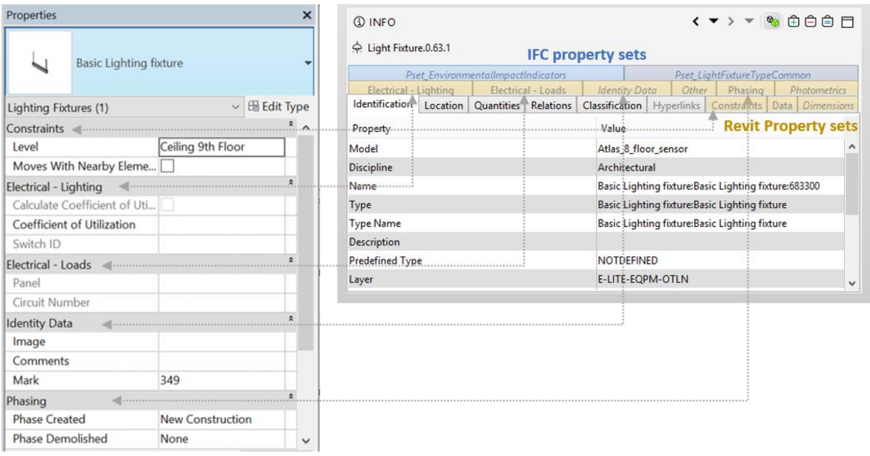



*Figure 44. IFC and Revit property sets in the IFC model*

The next evaluation was between the RDF graph generated in comparison with the original IFC file. This evaluation focused on elements from the use case scope, namely the building, the building floors, the rooms, and the sensors and lighting fixtures. The first observation is that the numbers that are displayed beside the element name in the RDF come from the identification of this element in the IFC file. For instance, if the line in the IFC file that identify a space entity is '#9832 = IFCSPACE ('0LnrpxIav0Ju4gTy7JmWDr', #42, '88', $, $, #9802, #9829, 'Lab extension', .ELEMENT., .SPACE., $)', this '#9832' is used every time a reference for this element is made, similarly, in the RDF, this elements is identified as 'inst:space_9832'. After the data processing in the application, this element is displayed like 'space (9832)' hence the user uses this number to differentiate between all spaces in the building, and more information about the entity is showed after it is selected. Ideally, it should be replaced by the classification system currently used by the facility managers so that they can easily identify the resource they want.

A second observation that also applies to all elements in the RDF is that the GUID information is in a decompressed state and hence it is how they appear to the end-user. In the example above, the GUID for Space_9832 is presented as 'bot:hasGuid "15c75cfb-4a4e-404f-812a-77c1d3c20375" '. This will only be an issue if the user needs this information to find an element in another BIM tool that displays this ID since most of them utilize the compressed version, but that seems an unlikely scenario.

Another difference regards the information present in the IFC file that is filtered out in the RDF graph. In the RDF, the site, building, and building storeys only contain their classification according to the BOT ontology, their GUID, these entities' names identified as a label property, the relationship they have with the other elements, and any comment that might be associated with them. This means that for these elements the ontologies used cleaned the file content and excluded the property sets from Revit and IFC (Figure 45).

For this use case, it is a good approach since the information excluded was not defined by the researcher (e.g. for *Pset_BuildingStoreyCommon* property *AboveGround* value *UNKNOWN*) or is irrelevant data that is related to the Revit environment (e.g. for *Identification*, property *Discipline*, value *Architectural*; for *Constrains*, property *Elevation Base*, value *Project Base Point*). This corrected the excess of information generated by the exporting setting in Revit and simplified the data displayed for the end-user. Thus, this process generates a simple building data representation that can be linked to O&M data.

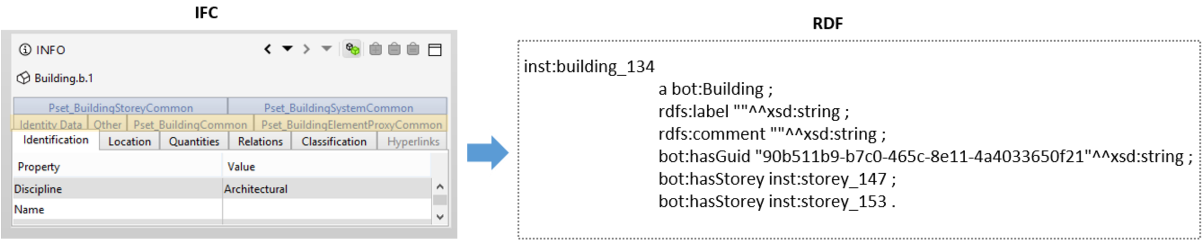



*Figure 45. Data present in the building instance in IFC x RDF*

For the building elements, the PROPS ontology captured the information from the property sets in IFC. To analyse this aspect, the spaces, sensors, and lighting fixtures elements were evaluated in more depth. In this case, all properties single value entities in the IFC were included in the RDF. Hence, the properties with non-defined values or that are not relevant for the use case were also transferred. These properties were filtered out in the data process stage in the application development.

Moreover, some minor issues were found. The properties with dimension values are captured from the *IFCpropertysingle* values which bring the name and value of this property, but it doesn't have its measurement unit since this was defined as IFC SI unit for the whole project in the IFC (Figure 46). Also, the numbers are showed with high precision, probably it would be worth to post-process the data in the application to display only two or three decimal places.

<u>In IFC:</u>
#9890= IFCPROPERTYSET('0LnrpxIav0Ju4gSSFJmWDr',#42,'Dimensions',$,(#312,**#9878**,#9879,#9880));
#9878= IFCPROPERTYSINGLEVALUE(**'Area'**,$,IFCAREAMEASURE(**14.7975000000002**),$);
#44= IFCSIUNIT(*,.AREAUNIT.,$,.**SQUARE_METRE**.);

<u>In RDF:</u>
props:**area** "**14.7975000000002**"^^xsd:double ;

*Figure 46. Comparison between dimension display in IFC and RDF*

Furthermore, the PROPS ontology deals with the IFC property sets but not with the IFC quantity sets, hence the data included in this category was filtered out of the application. That was observed only in the spaces entities, for which the *Qto_SpaceBaseQuantities* set includes the information about the gross floor area, the gross perimeter, the gross volume, the height, and the net floor area of the room. In the RDF graph, there is only the area, the perimeter, and the unbounded height from the *Dimensions* property set.

However, these are minor debugging issues that can be solved by small changes in the conversion tool which is currently under development and open to improvements. In addition, It is also possible to edit, add, and remove RDF resources directly in the triplestore. Moreover, one advancement for the application development would be to add the conversion tool used directly in the app code, and also let the API handles the creation of the repository and the graph inclusion in it using the GraphDB and RDF4J infrastructure. This would eliminate part of the system preconditions and create new system requirements to be considered in the system architecture.

## 6.4 Results conclusion

This chapter brings the main results from the evaluation of the ZEL BIM models and documentation and the prototype app. The ZEL models were made for the design stage, hence they were used to extract requirements for the handover model. The main improvement points are to include the facility management data requirements in the BIM protocol, and to present the building systems with enough detail for this stage and with the correct semantic relationship between the elements. Some examples are:

- Create a balance between the level of detail and model complexity, for instance, model all relevant parts of the buildings's systems but with the correct level of geometrical details to avoid unnecessary heavy models

- Attach properties to these elements that the facility managers commonly consult like the equipment reference, the manufacturer, the material reference, or the system characteristics and eliminate the ones that are not related to this stage.

- Include a classification system that makes sense for the facility management team for them to fast locate the building elements and areas.

- Add spaces and zones entities in the model to allowed sensor data to be associated with these entities and enable building navigation to the room level.

Moreover, the application responsive layout and interaction mechanisms complied with the design requirements. They enabled the app user to have access to a digital database with the building elements information from the BIM model. In addition, they could access in the same platform the building indoor conditions provided by the sensors in the facility. This connection between the building elements and the sensor data was tested with sample data and achieved by entering the sensor identification inside the BIM sensor elements. However, more tests should be done to define app performance with live sensor data. Moreover, the data display mechanisms are based on the predicate values from the LBD ontologies used to define the building RDF graph. Hence, it only requires small adjustments to be applied to other buildings and use case scenarios that apply these same ontologies.

Furthermore, the data conversion process of the BIM data, part of the system preconditions, was also evaluated. IFC is already a consolidated standard to exchange information between BIM software. Thus, it was opted to work with this file format to create the RDF graph rather than use the vendor dependent Revit file. This creates the need to verify the mapping between Revit categories and IFC classes, and define exporting settings that are compatible with the information needed in the model. In this study case, the sensor devices were modelled as electrical fixtures and mapped to IfcSensor, and all Revit properties were exported and pos-processed in the application, but other approaches are available. The RDF graph only had minor issues that can be solved by small changes in the conversion tool or by editing the triple store. Moreover, an improvement for the application would be to include the IFC conversion and storage in the triplestore directly in the app code eliminating this system precondition, but that will generate new system requirements.

# 7 Discussion and conclusion

This study targets a decentralized approach for the integration of BIM data and O&M data through web technologies and semantic web technologies. It aims at evaluating what benefits this approach could bring to support facility management tasks and the challenges for its implementation. To achieve these goals, two study cases were proposed to extract conclusions regarding how BIM and BMS data can be retrieved by a web application, what functionalities this system has to support to post-process and display this information in a useful way and what potential use cases are expected for the facility managers. Based on this scope the main research question of this thesis is:

- **What are the challenges and potentialities of implementing a web-based application with building data from a BIM model and a BMS to support tasks in the O&M phase?**

To be able to answer this question, first, typical facility management tasks were extracted from the operation and management procedures in the Atlas building and the Zero Emission Lab in the TU/e campus. Next, the current IT tools were evaluated as a way to support these tasks. Some opportunities for improvement were found namely, a faster way to access different data sources, a decrease in the time expend inputting manual data in these applications, and improvement in their geometrical visualization capabilities.

Following this evaluation, a literature review in the potential of BIM for the O&M stage was performed to analyse how BIM data can enable these opportunities for improvement and disclose other possibilities for BIM in this field. This literature review also covered reported challenges for the implementation of BIM in this stage. Furthermore, the findings from this review were crossed with the facility management tasks to identify potential use cases for the application.

Complementing this investigation, semantic web technologies and web technologies concepts were explored to create a proof of concept application that can connect BIM data with sample energy consumption data from a BMS sensor. The literature review and the development of the proof of concept were used to answer the research sub-questions.

## 7.1 Potential use cases and O&M data

The first research sub-question is defined as:

- **What are the potential use cases for the application and what kind of O&M data is needed to enable them?**

The facility managers need to access relevant and up to date data about the facility to improve the building daily operations, along with historical data, to define long term strategies to decrease resource consumption and keep the user satisfaction high. However, due to the heterogeneous data sources and formats, they waste a lot of time to find what they need and to extract knowledge from big collections of data. Considering the facility management tasks from the two study cases and the proposed connection approach with BIM data, potential uses were identified in the following areas:

1) Maintenance

Maintenance procedures need information from several data sources. Some of them are common delivered in paper or digital spreadsheets like equipment lists, product data sheets, lists of spare parts, warranties, and schedules of preventive maintenance (Wong et al., 2018). Building data available in the BIM model is also necessary to calculate material quantities, identify what equipment is needed for the action, and anticipate difficulties to access the building systems (Liu & Issa, 2016). Moreover, the maintenance history records with the correct level of detail are important to perform assessments of recurrent or high-risk problems like case-and-effect evaluations, root cause, and failure modes analyses, and risk matrices to give a strategic approach to preventive maintenance actions (Arunraj & Maiti, 2007). In addition, Modern BMS systems have predictive maintenance capability that supplies the facility manager with information about future demands for maintenance, potentially decreasing disruptions time (Bumblauskas et al., 2017).

In the proposed application, the hardcopy files and spreadsheets should be digitalized and connected to the BIM data in the triplestore using the linked data principles. In this scenario, the facility manager would be able to retrieve all the technical information available to a specific building element in the table component. Moreover, the maintenance records could be retrieved from the CMMS and associated with the building elements by the app API. Interactions options like filters, time series aggregation, and quantities take-off can be included to support analysis in these records. Furthermore, predictive maintenance data can be retrieved for the BMS sensors and be displayed in the element status part of the user interface. With the implementation of the 3D viewer, the investigation on the site conditions can be performed in the application. In addition, considering the portability features of web technologies, the app could be used during site inspections in the smartphones and tablets web browsers to have information about the building elements on the fly.

2) Logistics

The main data sources of the building logistics operations are the building assets, its location, and its demand for supplies replenishment. In the educational context, this also includes special supplies to support researches and study activities. The logistic stream from placing the order until receiving the product need to be in close synchronization with the building demand to don't have equipment out of service or delayed projects.

Assets track systems use tags or barcodes to store information about the asset location and status (Wong et al., 2018). Retrieve this information from this system and combining it with BIM data has the potential to increase the time to find an asset in the building and the status data can be used to generate alerts for intervention in the user interface. Moreover, in the 3D viewer, the locations that require actions can be grouped to determine areas with a higher or lower priority of intervention. However, an important remark regards the fieldwork to update the equipment status. BMS systems have sensors to automatically update the supply status of its parts that can be retrieved by the application. But individual devices like printers and snack machines that are monitored by asset tracking systems commonly need the manual input of the asset status in the tag or barcode used to track them. In these cases, the application use will depend on the reliability of this status data.

3) Space Management

The data sources to manage space are the building and its users. Mainly the number of occupants in the building, what activities they perform, the current space available, and its physical conditions. Space management tasks involve allocating people (Pishdad-Bozorgi et al., 2018), identify when areas in the building don't comply with the quality or user standards, and need to be renovated or refurbished (Volk et al., 2014), and provide data to support acquisition or disposal of real estate (Lavy, 2008).

The IT tools for O&M can measure the building occupancy in the percentage of occupied areas using motion sensors or more precisely in the numbers of building occupants using people tracking systems. This data can be associated with the spaces element in the BIM data that represent the rooms in the building to evaluate trends and detect opportunities for more compact buildings. In addition, BIM data with a correct level of detail provides useful information to manage renovation and refurbishment procedures. Some related examples are to calculate the volume of waste and other sustainability assessments, plan the dissemble of components, verify the number of users impacted by the construction activities and find options for relocation, find storage places for construction materials, and define safety areas based in the flow of people in the area and the risk involved in the construction.

4) Safety

Keep people and assets protected are processes that depend on the activities performed in the building, the type of assets and the flow of people in it, and its spatial configuration. Considering the two studies cases the main risks are fires, unauthorized people access, and robbery. The ZEL environment also brings the concern of explosions and contact with toxic substances, and Atlas needs evacuation strategies for large groups of people. The data needed to support tasks in this field include the number and identification of people in the building, the location of valuable assets, measurements of several safety sensors (e.g. smoke, heat, oxygen level), data from monitoring systems (e.g. CCTV), and data about the building to plan evacuation scenarios.

In the context of the proposed application, the data from the sensors can be displayed in the user interface associated with the rooms for monitoring of the conditions of the spaces, especially during the execution of procedures with risk. In addition, alerts associated with safety thresholds can be implemented. Moreover, the 3D viewer could be used to perform blind spots assessment for security signs, alarm sensors, and security cameras, when these devices are included in the BIM data. However, the display of computer simulations for evacuation studies in the 3D viewer requires further investigation in these algorithms.

5) Indoor conditions

The quality of the indoor environment is monitored according to health and safety work environment regulations and have repercussion for user satisfaction and performance (Kallio et al., 2020). The IEQ level depends on several measurements like indoor air quality, daylight, artificial light, acoustics, thermal, humidity level, and survey with building users. Bring these sensors data inside the application associated with the BIM spaces can make it easier for the facility managers to find information about a room in the building. This data can also be used as input for IEQ calculations that can be performed by the application and this indicator can be implemented in the viewer for an overview of the building status.

6) Cost management

In this thesis, cost management was focused on the optimization of resource usage. The daylight, lighting, energy consumption, and thermal data from the lighting and HVAC BMSs can be associated with the BIM data to evaluate the building performance (Hosseini, et al., 2019; Yan et al., 2013). This data can also be implemented in the 3D viewer using a colour scale in the space elements to identify opportunities to decrease resource usages like the placement of skylights, windows, and other openings to depend less on artificial lighting during part of the day and the identification of potential heat loss areas. However, similar to the evacuation simulations, a more precise evaluation of the problematic points in the facility relies on the implementation of simulators that need to be further analysed.

In this thesis, the proof of concept generated a layout adapted to show BIM data and energy consumption data, with room for the future inclusion of a 3D viewer and other sensor readings. Thus, its current configuration supports best the complementation of BIM data with indoor conditions sensors and resource consumption monitoring.

## 7.2 System architecture

After defining the potential use cases, the next question regards how the approach can be implemented to support the O&M use cases. Thus, the second research sub-questions is defined as:

- **What functionalities the system architecture should enable to support O&M tasks and how they can be achieved?**

From related studies in this topic, it was concluded that the information needed to be presented in a clear and intuitive way, preferably without relying on the technological expertise of the user (Riaz et al., 2014). The effort to make data more user friendly was present in all studies that created user interfaces with colour schemes (Chang et al., 2018; Riaz et al., 2014), graphs (Riaz et al., 2014), reports (Hu et al., 2018), interactive options (Hu et al., 2018; Mccaffrey et al., 2015; Rasmussen et al., 2017), and pop-up alerts being used to improve the user experience (Hu et al., 2018; Riaz et al., 2014). Moreover, the use of databases to store historical and live data appeared as a necessity to organize the big volume of information generated throughout the building life cycle or to make BIM and BMS data reachable on the web. Some used relational databases (Patti et al., 2012; Riaz et al., 2014), and some used documents (Rasys et al., 2014) and graphs databases (Chevallier et al., 2020; Kim et al., 2018; Rasmussen et al., 2017), depending on the scope of the research, but all provided some sort of search engine to make the exploration of the data possible. The use of 2D and 3D geometry for building navigation was also highlighted as an important feature for facility management and was implemented from BIM or GIS data models generated from the BIM model (Hu et al., 2018; Mccaffrey et al., 2015; Shen et al., 2012).

These concepts were incorporated in the development of the proof of concept. It aimed at decreasing the time necessary to find information about a specific building element by using BIM data to allow building navigation, based on its embedded semantic relationships, and to associate data in an object-oriented approach. Moreover, the app post-processes the information to improve the readability of the dataset and tailor it to this stage of the building lifecycle. Furthermore, the app layout was made to the future inclusion of a 3D viewer that will give spatial context to user interaction with the database. The idea is to have intuitive model filters using buttons and model navigation features.

To enable these functionalities, the app needed to handle API communication, data process, and display, and this can be achieved with web technologies. In this project, the Angular design framework features were used to simulate the integration of BIM data and sensor data in a simplified user interface. This approach provides a basic platform for the visualization of data from different sources to give the facility manager an overview of the building conditions without having to change the underlying system architecture of the IT tools used to support O&M. A web-based application also brings other advantages like portability, which is an important feature for the AECO industry since it involves site inspections where nowadays smartphones and tablets are more suitable than computers or paper-based building plans and checklists. Another advantage is scalability, to allow the app to include new datasets and historical data as the building ages. Moreover, semantic web and linked data principles can be used to connect BIM data with data from other domains that will be displayed together in the element information table. This has the potential to decrease the time expend to find all the necessary information about an asset, device, or system in the building.

Thus, the prototype developed has the basic infrastructure to integrate BMS data and BIM data to which other sensors data can be added in the user interface and other datasets can be linked to the building elements. However, the implementation of the application to support the described use cases relied on different conditions. Regarding API communication, some level of data manipulation such as the digitalization of files, changes in data format, or inclusion in web databases is common to all use cases to enable hard copy or digital files to be reachable by the application. Furthermore, the facility management IT tools needs to enable communication with third-part applications with endpoints where information about the system operation and status can be reached.

Regarding data process and display, the data exchanged by APIs is commonly in XML or JSON format which makes it easier to manipulate and transform using DOM or JSON parsers. The post-process challenges will depend on which functionalities will be included in the user interface. For instance, to achieve the chart display of the sensor data in the prototype, the measurement values were aggregated like points in a timescale by extracting the data in the XML tags into simple arrays. In addition, the arrays of values needed to be converted in a format accepted as an axis dataset, in this case, number and dates. Moreover, some functionalities depend on the implementation of the 3D viewer like the visualization of the building geometry.

Other implementation challenges that appear in the literature are the necessity of finding lighter solutions to display 3D geometry in a format more compatible with the web technologies (Hu et al., 2018; Riaz et al., 2014), the need of more domain-specific ontologies for the O&M stage (Kim et al., 2018), and concerns with web safety and data privacy (Livraga, 2015; Oppliger, 2003), and performance in lower-powered devices (Mendoza & Gu, 2018).

## 7.3 BIM data

The last research question regards the barrier of BIM data to be used in this O&M stage:

- **What challenges BIM data need to overcome to support these use cases?**

From the literature research, it is possible to say that there are a lot of possibilities for BIM in the O&M field because of its characteristics of model visualization and information aggregation at the object level. However, its potential as a source of knowledge for O&M is intrinsically related to the accuracy level of the model (Leite et al., 2011), and the information embedded in it (Ashworth et al., 2019). Moreover, its usability in application outside the BIM modelling environment raises several interoperability issues (Pishdad-Bozorgi et al., 2018).

Both levels of information and modelling problems were found in the first study case. The model had missing property values and extra property sets beyond the agreed in the BIM protocol. In addition, modelling problems such as elements intersections, duplications, and semantic incongruences were found. The documentation and model evaluated were developed for the design phase and hence don't include yet requirements that reflect how this data is going to be used in the O&M phase like the level detail of the systems representation and the information from the construction and design phase to be included in the model. In the ZEL case, the university already has a maintenance and management department for the whole campus and because of that the evaluation of the design model was the first step to define requirements for the handover model. However, that is not a typical situation since commonly the firm responsible for the O&M process is not defined in the design phase.

Furthermore, the BIM models are delivered commonly in the IFC format since it is the most consolidated vendor-neutral standard for BIM tools. But this format has several issues reported in the literature to be used in a linked data approach, namely problems to adapt or extend the schema to include other domains information (Pauwels et al., 2017), and difficulty to apply query methods and lack of web compliance (Beetz et al., 2009). For the Atlas study case, the LBD ontologies were used to convert IFC to RDF, a format that can address these shortcomings.

The simplified BOT ontology cleaned the extra information from IFC for the site, building, and building storeys, and the PROPS ontology transferred all properties associated with the building elements from IFC to RDF enabling them to be read by the web application. These ontologies bring the advantage of being a simpler data model to represent the building topology what makes the file lighter and query writing more compact (Bonduel et al., 2018). At the same time, it enables the future linking of other information to the building elements in a scalable and decentralized way when following the linked data principles.

Hence, Although fully interoperability is not achievable and some data loss is always going to occur, semantic web technologies proved to be a viable way to enable the retrieval of BIM data by a web application to support facility management tasks.

## 7.4 Conclusion

This research main goal was to analyse the challenges of using a web application for a decentralized integration between BIM data and BMS data and how it could be used to support facility management tasks in the O&M stage. This included a literature review in the topics of facility management and the IT tools to support O&M procedures and potentials and barriers for implementing BIM data in this stage. After this evaluation, it was possible to disclose potential use cases for the integration of O&M data with BIM data. Moreover, Studies in this topic were reviewed to identify common features considered relevant to support FM tasks and discover challenges for this integration. Furthermore, semantic web and web technologies concepts were explored to contextualize the approach used to develop the proof of concept app.

To achieve the research goal, this thesis used the Zero Emission Lab and the Atlas building as study cases to define facility management tasks that can be improved by a single source of visualization for BIM data and FM data. Moreover, the ZEL study case complemented the literature review by analysing a real BIM model and its related documentation and modelling agreements. Also, the conditions BIM and BMS data needed to meet to be retrieved by the app and how it could transform them to be displayed in a more user-friendly way were assessed during the process of a prototype app development. The use case scenario and the datasets used in the app were provided by the Atlas Living lab study case and the BIM model was developed by the researcher. Due to the interruption of the data collection of the Atlas Living Lab in March 2020 to comply with the Covid-19 preventive measures, its infrastructure was used theoretically to build the use case scenario and pre-code a future connection with the lab server. In addition, a sample data generated by the BMS virtual test system was included in the app to enable the evaluation of the data process and display stage.

In conclusion, the app can potentially benefit maintenance, logistics, space management, safety, indoor conditions, and cost management tasks. These potentials derivates from the possibility to aggregate O&M information in the building element level, to have access to different sensors readings in the same user interface, and to extract geometrical information for the BIM data to future enable a 3D viewer in the application. To enable these functionalities, the basic tasks the system architecture has to deal with are the retrieval, process, and display of data. Considering use cases and the literature review, interesting features to be implemented are sensor data in charts formats, interactive options for data navigation, the materialization of sensor data in the building with colour schemas, alerts for defined thresholds, generation of reports for determined periods of time, and a 3D viewer to take advantage of the geometric information in of BIM data outside the BIM authoring tools.

These functionalities can be achieved with front-end development frameworks such as Angular and JavaScript libraries. Furthermore, web technologies have portability and scalability characteristics that provide mobility and long-term use potential, since virtually web applications can be accessed anywhere and in any device with an internet connection and the web database systems behind it can have very high storage capacity.

Regarding the BIM data, it has some incompatibilities to be retrieved by web applications. To solve these challenges, Semantic web technologies were used to convert the BIM model into RDF according to the LBD ontologies. The resulting file was included in a graph web database, where the unique identified resources that represent the building entities and their relationships and properties can be retrieved by the application. Furthermore,  this simplified model to represent the building data can be linked to other O&M data to enable the aggregation of FM data to the object level.

Therefore, the developed proof of concept was a first effort to build the basic infrastructure for a user interface that can retrieve BIM data and sensor data on the web to analyse potential use cases and disclose implementation challenges. The next research development steps include investigating the creation of a linked database for the O&M data with BIM in its core, the retrieval of live sensor data, and the implementation of more functionalities for user interaction with the data in the front-end. Beyond that, further web development steps are improvements in the application code and the implementation of web safety measures like the creation of an independent back-end.

## 7.5 Scientific and Societal relevance

Considering the scope of the research, this thesis aimed at giving some contribution to the scientific development in the topics of BIM data to support O&M tasks. The investigation in the topic included a literature review in the potentials of BIM for this stage, the challenges derived from the BIM modelling process and data format, and how semantic web and web technologies could be applied to overcome some of these barriers. This investigation was complemented by two study cases to disclose potential use cases for BIM data in this stage based on real facilitates and develop of a proof of concept application for the proposed integration approach.

The societal relevance of improving facility management regards a more intelligent use of natural resources and better spatial and environmental conditions which influences the performance and wellbeing of the building users. Develop better ways for facility managers to perform its tasks, as tools for improved data access, brings benefits for the management of building and people resources. This can result result in more sustainable solutions for enterprise growth and an increase in user satisfaction.

## 7.6 Recommendation for prototype improvement

This topic brings a critical evaluation of the approach adopted and highlights opportunities for improvement and alternative approaches.

**Data conversion**

Regarding the data conversion, it was done in two steps, Revit to IFC and IFC to RDF, outside the application. A better approach would be to let the application API perform the conversion and inclusion of the RDF triples in GraphDB. The user would only need to upload the IFC file on the client-side decreasing the system preconditions. Another approach would be to eliminate the number of steps needed converting the Revit file directly to RDF to decrease some data loss and conflict with the IFC schema. But being Revit a commercial application this will bring limitations to apply the app to buildings that were not modelled in it.

**Data retrieval and post-process**

The application back-end and front-end processes are handled in the Angular framework. A safer approach would be to create an independent back-end to deal with the external API communications and a front-end just for the client-side. Also, this separation could open opportunities to simplify the flows of data exchange between the front-end components. Moreover, a good approach is to have some mechanism in place to detect errors in the data collection. For instance, with live sensor data, the physical sensor devices or the network might suffer disturbances generating errors in the measurements. One approach to detect these error values is to use the *rxjs* library with methods like *pipe*, *tap,* and *catchError* that can be combined to detect errors in the stream of data. Furthermore, the data processing has a lot of room for improvement investigation considering the new datasets to be included to support the potential use cases.

**Integration approach**

The application handles the integration of BIM data with sensor data in the API. In this way, the BMS deals with the storage of live sensor data and the app should just retrieve data from it. Another approach is to link the sensor data directly to the building elements in the triplestore. For instance, The SOSA/SSN ontologies can be used to describe the wireless sensors and their observations and link this data to the building data (Chevallier et al., 2020).

**3D viewer implementation**

A component for the visualization of the 3D representation of the building was included in the app layout but not implemented. Several factors need to be considered for this feature. For instance, how it is going to be the interaction of this component with the table component since what is displayed in both need to be synchronized. Another is what JavaScript 3D engine will be used to implement it, and what user navigation tools could be useful, especially considering facilities with systems with high complexity. An example of this implementation is the 3D viewer developed by Mccaffrey et al. (2015) that used a COLLADA file in a WebGL framework to enable the visualization of the building geometry in a web application.

## References


Akcamete, A., Akinci, B., & Garrett, J. H. (2019). Potential utilization of building information models for planning maintenance activities. *EG-ICE 2010 - 17th International Workshop on Intelligent Computing in Engineering.* http://www.engineering.nottingham.ac.uk/icccbe/proceedings/html/76.htm

Alileche, L. (2018). *Use of BIM for optimal management of existing buildings* [Universite of Lille, Ecole Doctorale Sciences pour I'ngenieur]. http://ori.univ-lille1.fr/notice/view/univ-lille1-ori-467961

Angular. (n.d.-a). *Communicating with backend services using HTTP*. Retrieved May 27, 2020, from https://angular.io/guide/http

Angular. (n.d.-b). *Dependency injection in Angular*. Retrieved May 27, 2020, from https://angular.io/guide/dependency-injection#create-an-injectable-service-class

Angular. (n.d.-c). *Indroduction to modules*. Retrieved June 2020, 10, from https://angular.io/guide/architecture-modules.

Angular. (n.d.-d). *Introduction to services and dependency injection*. Retrieved June 10, 2020, from https://angular.io/guide/architecture-services

Arslan, M., Riaz, Z., Kiani, A. K., & Azhar, S. (2014). Real-time environmental monitoring, visualization, and notification system for construction H&S management. *Journal of Information Technology in Construction*. http:/itcon.org/2014/4

Arunraj, N. S., & Maiti, J. (2007). Risk-based maintenance-Techniques and applications. *Journal of Hazardous Materials*, 142(3), 653–661. https://doi.org/10.1016/j.jhazmat.2006.06.069

Ashworth, S., Tucker, M., & Druhmann, C. K. (2019). Critical success factors for facility management employer's information requirements (EIR) for BIM. *Facilities*, 37(1–2), 103–118. https://doi.org/10.1108/F-02-2018-0027

Atlas Living Lab. (2019). *Atlas Living Lab Data Management Plan*.

Aziz, N. D., Nawawi, A. H., & Ariff, N. R. M. (2016). ICT Evolution in Facilities Management (FM): Building Information Modelling (BIM) as the Latest Technology. *Procedia - Social and Behavioral Sciences*, 234, 363–371. https://doi.org/10.1016/j.sbspro.2016.10.253

Beetz, J., Van Leeuwen, J., & De Vries, B. (2009). IfcOWL: A case of transforming EXPRESS schemas into ontologies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 23(1), 89–101. https://doi.org/10.1017/S0890060409000122

Benymol, J., & Abraham, S. (2020). Perfomance analysis of NoSQL and relational databases with MongoDB and MySQL. *Materials Today: Proceedings*. https://doi.org/https://doi.org/10.1016/j.matpr.2020.03.634

Berners-Lee, T. (2006). *Linked Data*. https://www.w3.org/DesignIssues/LinkedData

Bonduel, M., Oraskari, J., Pauwels, P., Vergauwen, M., & Klein, R. (2018). The IFC to Linked Building Data Converter - Current Status. In M. Poveda-Villalón, P. Pauwels, & A. Roxin (Eds.), *Proceedings of the 6th Linked Data in Architecture and Construction Workshop* (pp. 34–43). http://ceur-ws.org/Vol-2159/

Borrmann, A., Konig, M., Koch, C., & Beetz, J. (2015). Building Information Modeling: Why? What? How? In A. Borrmann, M. konig, C. Koch, & J. Beetz (Eds.), *Building Information Modeling: Technology Foundations and Industry Practice*. https://doi.org/https://doi.org/10.1007/978-3-319-92862-3

BuildingSMART. (n.d.-a). *ifcOWL*. Retrieved July 10, 2020, from https://technical.buildingsmart.org/standards/ifc/ifc-formats/ifcowl/

BuildingSMART. (n.d.-b). *Industry Foundation Classes (IFC) - An Introduction*. Retrieved June 25, 2020, from https://technical.buildingsmart.org/standards/ifc/

BuildingSMART. (2018). *Industry Foundation Classes 4.1.0.0 – F.1 IFC2X3 to IFC4*. https://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML/

Bumblauskas, D., Gemmill, D., Igou, A., & Anzengruber, J. (2017). Smart Maintenance Decision Support Systems (SMDSS) based on corporate big data analytics. *Expert Systems with Applications*, 90, 303–317. https://doi.org/10.1016/j.eswa.2017.08.025

Carbonari, G. (2016). Inefficiency in FM , can BIM help ? In P. A. Nielsen, S. B., & Jensen (Ed.), *15th EuroFM Research Symposium* (pp. 229–239). https://gala.gre.ac.uk/id/eprint/19184/

Chandra, D. G. (2015). BASE analysis of NoSQL database. *Future Generation Computer Systems*, 52, 13–21. https://doi.org/10.1016/j.future.2015.05.003

Chang, K. M., Dzeng, R. J., & Wu, Y. J. (2018). An automated IoT visualization BIM platform for decision support in facilities management. *Applied Sciences* , 8(7). https://doi.org/10.3390/app8071086

Cheng, J. C. P., & Ma, L. Y. H. (2013). A BIM-based system for demolition and renovation waste estimation and planning. *Waste Management*, 33(6), 1539–1551. https://doi.org/10.1016/j.wasman.2013.01.001

Chevallier, Z., Finance, B., & Boulakia, B. C. (2020). A Reference Architecture for Smart Building Digital Twin. *Proceddings of the International Workshop on Semantic Digital Twins Co-Located with the 17th Extended Semantic Web Conference*. http://ceur-ws.org/Vol-2615/

Codinhoto, R., & Kiviniemi, A. (2014). BIM for FM: A Case Support for Business Life Cycle. *11th IFIP International Conference on Product Lifecycle Management (PLM)*, 63–74. https://doi.org/10.1007/978-3-662-45937-9_7ï

Conallen, J. (1999). Modeling web application architecture with UML. *Communications of the ACM*. https://doi.org/https://doi.org/10.1145/317665.317677

Crowe, S., Cresswell, K., Robertson, A., Huby, G., Avery, A., & Sheikh, A. (2011). The case study approach. *BMC Medical Research Methodology*, *11*. https://doi.org/10.1186/1471-2288-11-100

Curry, E., Donnell, J. O., Corry, E., Hasan, S., Keane, M., & Riain, S. O. (2013). Advanced Engineering Informatics Linking building data in the cloud : Integrating cross-domain building data using linked data. *Advanced Engineering Informatics*, 27(2), 206–219. https://doi.org/10.1016/j.aei.2012.10.003

Cursor. (2019). *Atlas mustn't become an experimental maze*. https://www.cursor.tue.nl/en/background/2019/maart/week-2/atlas-mustnt-become-an-experimental-maze/

Dibley, M. J. (2011). *An Intelligent System for Facility Management* (Issue October) [Cardiff University]. http://orca.cf.ac.uk/id/eprint/23277

Di Santo, K. G., Di Santo, S. G., Monaro, R. M., & Saidel, M. A. (2018). Active demand side management for households in smart grids using optimization and artificial intelligence. *Measurement: Journal of the International Measurement Confederation*, 115(September 2017), 152–161. https://doi.org/10.1016/j.measurement.2017.10.010

Edirisinghe, R., London, K. A., Kalutara, P., & Aranda-Mena, G. (2017). Building information modelling for facility management: *Are we there yet? Engineering, Construction and Architectural Management*, 24(6), 1119–1154. https://doi.org/10.1108/ECAM-06-2016-0139

Durier, F. (2017). Smart HVAC system in buildings and energy savings. Retrieved August 6, 2020, from https://www.buildup.eu/en/news/overview-smart-hvac-systems-buildings-and-energy-savings-0

Epstein, E. (2012). *Implementing Successful Building Information Modeling*. Artech House. https://us-artechhouse-com.dianus.libr.tue.nl/CloudPublish/book.aspx?isbn=9781608071401

Farias, T. M. De, Roxin, A.-M., & Nicolle, C. (2015). IfcWoD , Semantically Adapting IFC Model Relations into OWL Properties. *Proc. of the 32nd CIB W78 Conference*. https://arxiv.org/abs/1511.03897

Fatima, H., & Wasnik, K. (2016). Comparison of SQL, NOSQL and New SQL Databases in light of internet of things - A Survey. *International Journal of Advances in Electronics and Computer Science*, 3, 2393–2835. https://doi.org/10.1109/IBSS.2016.7940198

Flyvbjerg, B. (2006). Five misunderstandings about case-study research. *Sage Publications*, 12(2), 219–245. https://doi.org/10.1177/1077800405284363

FMLink. (2003). *IFMA redefines facilities management*. https://fmlink.com/articles/ifma-redefines-facilities-management/

Fox, L. (2018). *Advantages and Disadvantages - Web Apps*. https://objectiveit.com/blog/the-advantages-and-disadvantages-of-web-apps/

Ghaffarianhoseini, A., Tookey, J., Ghaffarianhoseini, A., Naismith, N., Azhar, S., Efimova, O., & Raahemifar, K. (2017). Building Information Modelling (BIM) uptake: Clear benefits, understanding its implementation, risks and challenges. *Renewable and Sustainable Energy Reviews*, 75, 1046–1053. https://doi.org/10.1016/j.rser.2016.11.083

Heilmann, C., & Francis, M. N. (2007). *Web development solutions : Ajax, APIs, libraries, and hosted services made easy*. https://doi-org.dianus.libr.tue.nl/10.1007/978-1-4302-0393-3

Hosseini, S. M., Mohammadi, M., & Guerra-santin, O. (2019). Interactive kinetic façade : Improving visual comfort based on dynamic daylight and occupant 's positions by 2D and 3D shape changes. *Building and Environment*, 165(June), 106396. https://doi.org/10.1016/j.buildenv.2019.106396

Hu, Z. Z., Tian, P. L., Li, S. W., & Zhang, J. P. (2018). BIM-based integrated delivery technologies for intelligent MEP management in the operation and maintenance phase. *Advances in Engineering Software*, 115, 1–16. https://doi.org/10.1016/j.advengsoft.2017.08.007

IFMA. (n.d.). *What is Facility Management?.* Retrieved June 18, 2020, from https://www.ifma.org/about/what-is-facility-management

Intelligent Lighting Institute. (2019a). *Guiding principles TU/e Atlas Living Lab*. Retrieved June 9, 2020, from https://assets.tue.nl/fileadmin/content/onderzoek/ILI/Infrastructure/Guiding principles Atlas Living Lab-v1.0.pdf

Intelligent Lighting Institute. (2019b). *TU/e Atlas Living Lab Data Use Policy*. Retrieved June 9, 2020, from https://assets.tue.nl/fileadmin/content/onderzoek/ILI/Infrastructure/Atlas Living Lab Data Use Policy v1.0.pdf

Jensen, P. A. (2010). The facilities management value map: A conceptual framework. *Facilities*, 28(3–4), 175–188. https://doi.org/10.1108/02632771011023131

Jensen, P. A., van der Voordt, T., Coenen, C., von Felten, D., Lindholm, A. L., Nielsen, S. B., Riratanaphong, C., & Pfenninger, M. (2012). In search for the added value of FM: What we know and what we need to learn. *Facilities*, 30(5–6), 199–217. https://doi.org/10.1108/02632771211208486

Kallio, J., Vildjiounaite, E., Koivusaari, J., Räsänen, P., Similä, H., Kyllönen, V., Muuraiskangas, S., Ronkainen, J., Rehu, J., & Vehmas, K. (2020). Assessment of perceived indoor environmental quality, stress and productivity based on environmental sensor data and personality categorization. *Building and Environment*, 175. https://doi.org/10.1016/j.buildenv.2020.106787

Kelly, G., Serginson, M., Lockley, S., Dawood, N., & Kassem, M. (2013). BIM for facility management : a review and a case study investigating the value and challenges. *Proceedings of the 13th International Conference on Construction Applications of Virtual Reality*, January. https://www.semanticscholar.org/paper/BIM-for-facility-management%3A-a-review-and-a-case-Kelly-Serginson/ec6b1170e0baada36235fd29fce5dbc25218f917

Kepner, J., Gadepally, V., Hutchison, D., Jananthan, H., Mattson, T., Samsi, S., & Reuther, A. (2016). Associative array model of SQL, NoSQL, and NewSQL Databases. *IEEE High Performance Extreme Computing (HPEC) Conference 2016*. https://doi.org/10.1109/HPEC.2016.7761647

Khodeir, L. M. (2008). *Innovative Technologies for Integrating Facility Management Information Systems*. https://www.researchgate.net/publication/278300958

Kim, K., Kim, H., Kim, W., Kim, C., Kim, J., & Yu, J. (2018). Integration of IFC objects and facility management work information using Semantic Web. *Automation in Construction*, 87, 173–187. https://doi.org/10.1016/j.autcon.2017.12.019

Koninklijke Philips N.V. (2014). *Philips dynalite Controls: Technology Overview* (p. 32). https://www.lighting.philips.com/main/products/lighting-controls/dynalite/resources

Lavy, S. (2008). Facility management practices in higher education buildings: A case study. *Journal of Facilities Management*, 6(4), 303–315. https://doi.org/10.1108/14725960810908163

Leite, F., Akcamete, A., Akinci, B., Atasoy, G., & Kiziltas, S. (2011). Analysis of modeling effort and impact of different levels of detail in building information models. *Automation in Construction, 20(5),* 601–609. https://doi.org/10.1016/j.autcon.2010.11.027

Lindgren, J., & Widén, K. (2018). Diffusing building information management–knowledge integration, mechanisms and knowledge development. *Architectural Engineering and Design Management*, 14(5), 347–362. https://doi.org/10.1080/17452007.2017.1394260

Liu, R., & Issa, R. R. A. (2016). Survey: Common knowledge in BIM for facility maintenance. *Journal of Performance of Constructed Facilities, 30(3).* https://doi.org/10.1061/(ASCE)CF.1943-5509.0000778

Livraga, G. (2015). *Protecting privacy in data release*. Springer. https://doi.org/https://doi.org/10.1007/978-3-319-16109-9

Maes, K., Reynders, E., Rezayat, A., Roeck, G. De, & Lombaert, G. (2016). Offline synchronization of data acquisition systems using system identification. *Journal of Sound and Vibration, 381*, 264–272. https://doi.org/10.1016/j.jsv.2016.06.015

Matarneh, S. T., Danso-Amoako, M., Al-Bizri, S., Gaterell, M., & Matarneh, R. (2019). Building information modeling for facilities management: A literature review and future research directions. *Journal of Building Engineering, 24*. https://doi.org/10.1016/j.jobe.2019.100755

Mccaffrey, R., Coakley, D., Keane, M., & Melvin, H. (2015). Development of a Web-based BMS Data Visualisation Platform Using Building Information Models. *CIBSE Technical Symposium*. https://doi.org/10.13140/RG.2.1.4169.5840

Mendoza, A., & Gu, G. (2018). Mobile Application Web API Reconnaissance: Web-to-Mobile Inconsistencies & Vulnerabilities. *Proceedings - IEEE Symposium on Security and Privacy*, 2018-May, 756–769. https://doi.org/10.1109/SP.2018.00039

Mohanta, A., & Das, S. (2015). ICT-Based Facilities Management Tools for Buildings. *Proceedings of International Conference on ICT for Sustainable Development, ICT4SD*, 125–134. https://doi.org/DOI 10.1007/978-981-10-0129-1

Network Working Group. (1999). *Hypertext Transfer Protocol -- HTTP/1.* https://www.w3.org/Protocols/rfc2616/rfc2616.html

Nicał, A. K., & Wodyński, W. (2016). Enhancing Facility Management through BIM 6D. *Procedia Engineering, 164*, 299–306. https://doi.org/10.1016/j.proeng.2016.11.623

Nguyen, V. Q., Nguyen, H. D., Huynh, Q. T., Venkatasubramanian, N., & Kim, K. (2019). A scalable approach for dynamic evacuation routing in large smart buildings. *Proceedings - 2019 IEEE International Conference on Smart Computing*, SMARTCOMP 2019, 292–300. https://doi.org/10.1109/SMARTCOMP.2019.00065

Ock, J., Issa, R. R. A., & Flood, I. (2016). Smart building energy management systems (BEMS) simulation conceptual framework. *Proceedings - Winter Simulation Conference, 0*, 3237–3245. https://doi.org/10.1109/WSC.2016.7822355

Oppliger, R. (2003). *Security technologies for the world wide web*. ARTECH HOUSE, INC. https://us-artechhouse-com.dianus.libr.tue.nl/CloudPublish/book.aspx?isbn=9781580535854

Pärn, E. A., Edwards, D. J., & Sing, M. C. P. (2017). The building information modelling trajectory in facilities management: A review. *Automation in Construction*, 75, 45–55. https://doi.org/10.1016/j.autcon.2016.12.003

Parsanezhad, P. (2019). *Towards a BIM-enabled Facility Management : Promises, Obstacles and Requirements* [KTH Royal Institute of Technology]. http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-194878

Patel, A., & Jain, S. (2019). Present and future of semantic web technologies: a research statement. *International Journal of Computers and Applications*. https://doi.org/10.1080/1206212X.2019.1570666

Patti, E., Acquaviva, A., Abate, F., Osello, A., Cocuccio, A., Jahn, M., Jentsch, M., & Macii, E. (2012). *Middleware Services for Network Interoperability in Smart Energy Efficient Buildings. 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE),* 338–339. https://doi.org/10.1109/DATE.2012.6176491

Pauwels, P. (2014). Supporting Decision-Making in the Building Life-Cycle Using Linked Building Data. *Buildings*, 3, 549–579. https://doi.org/10.3390/buildings4030549

Pauwels, P., & Roxin, A. (2016). SimpleBIM : From full ifcOWL graphs to simplified building graphs. In S. Christodoulou & R. Scherer (Eds.), *Ework and ebusiness in architecture, engineering and construction* (Issue November, pp. 11–18). http://hdl.handle.net/1854/LU-8041826

Pauwels, P., & Terkaj, W. (2016). EXPRESS to OWL for construction industry : Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*, *63*, 100–133. https://doi.org/10.1016/j.autcon.2015.12.003

Pauwels, P., Zhang, S., & Lee, Y. C. (2017). Semantic web technologies in AEC industry: A literature overview. *Automation in Construction*, 73, 145–165. https://doi.org/10.1016/j.autcon.2016.10.003

Peng, Y., Lin, J. R., Zhang, J. P., & Hu, Z. Z. (2017). A hybrid data mining approach on BIM-based building operation and maintenance. *Building and Environment*, 126, 483–495. https://doi.org/10.1016/j.buildenv.2017.09.030

Pishdad-bozorgi, P., Gao, X., Eastman, C., & Patrick, A. (2018). Planning and developing facility management-enabled building information model (FM-enabled BIM). *Automation in Construction*, 87(February 2017), 22–38. https://doi.org/10.1016/j.autcon.2017.12.004

Rasmussen, M. H., Pauwels, P. , Lefrancois, M., Schneider, G. F., Hviid, C. A., & Karlshøj, J. (2017). Recent changes in the Building Topology Ontology. *LDAC2017 – 5th Linked Data in Architecture and Construction Workshop*. https://backend.orbit.dtu.dk/ws/portalfiles/portal/140019287/Untitled.pdf

Rasmussen, M. H., Hviid, C. A., & Karlshøj, J. (2017). Web-based topology queries on a BIM model. *5th Linked Data in Architecture and Construction Workshop*. https://doi.org/10.13140/RG.2.2.22298.95685

Rasmussen, M. H., Lefrançois, M., Bonduel, M., Hviid, C. A., & Karlshøj, J. (2018). OPM: An ontology for describing properties that evolve over time. *Proceedings of the 6th Linked Data in Architecture and Construction Workshop*. http://ceur-ws.org/Vol-2159/

Rasmussen, Mads Holten, Lefrançois, M., Schneider, G. F., & Pauwels, P. (in press). BOT : the Building Topology Ontology of the W3C Linked Building Data Group. *Semantic Web – Interoperability, Usability, Applicability*. http://www.semantic-web-journal.net/content/bot-building-topology-ontology-w3c-linked-building-data-group-0

Rasys, E., Hodds, M., Dawood, N., & Kassem, M. (2014). A Web3D Enabled Information Integration Framework for Facility Management. *Australasian Journal of Construction Economics and Building - Conference Series, 2*(1), 1. https://doi.org/10.5130/ajceb-cs.v2i1.3763

Riaz, Z., Arslan, M., Kiani, A. K., & Azhar, S. (2014). CoSMoS : A BIM and wireless sensor based integrated solution for worker safety in confined spaces. *Automation in Construction*, 45, 96–106. https://doi.org/10.1016/j.autcon.2014.05.010

Rodriguez, A. (2015). *RESTful Web services: Learn the basic principles of REST Web services*. IBM. https://developer.ibm.com/articles/ws-restful/

Rondeua, E. P., Brown, R. K., & Lapides, P. D. (2006). *Facility Management* (2nd ed.). Jonh Wiley & Sons, Inc.

Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). Introduction. In BIM handbook: A guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers (pp. 1–31). Wiley Online Library. https://doi.org/10.1002/9781119287568.ch1

Sattenini, A., Azhar, S., & Thurston, J. (2011). Preparing a building Information model for facility maintenance and management. *2011 Proceedings of the 28th ISARC*, (pp. 150-155). https://doi.org/10.22260/ISARC2011/0024

Shadbolt, N., Hall, W., & Berners-Lee, T. (2006). The Semantic Web Revisited. *IEEE INTELLIGENT SYSTEMS*. https://doi.org/10.1109/MIS.2006.62

Shen, W., Hao, Q., Mak, H., Neelamkavil, J., Xie, H., Dickinson, J., Thomas, R., Pardasani, A., & Xue, H. (2010). Systems integration and collaboration in architecture, engineering, construction, and facilities management: A review. *Advanced Engineering Informatics, 24(*2), 196–207. https://doi.org/10.1016/j.aei.2009.09.001

Shen, W., Hao, Q., & Xue, Y. (2012). A loosely coupled system integration approach for decision support in facility management and maintenance. *Automation in Construction*, 25, 41–48. https://doi.org/10.1016/j.autcon.2012.04.003

Signify Holding. (2018a). *Philips dynalite: The Dynalite System explained* (p. 32). https://www.lighting.philips.com/main/products/lighting-controls/dynalite/resources

Signify Holding. (2018b). *Philips dynalite Networked Controls: Envision Manager software*. https://www.lighting.philips.com/main/products/lighting-controls/dynalite/resources

Smith, S. (2011). Information management. In G. de Best, Rick; Langston, Craig; Valence (Ed.), *Workplace Strategies and Facilities Management: Building in Value*. Spon Press. https://doi-org.dianus.libr.tue.nl/10.4324/9780080521299

Sooriyarachchi, M. P., & Karunasena, G. (2010). *Information Technology usage in Building Operation & Maintenance Management*. https://www.researchgate.net/publication/324497171

Talamo, C., & Bonanomi, M. (2016). *Knowledge management and information tools for building maintenance and facility management*. Springer International Publishing. https://doi.org/10.1007/978-3-319-23959-0

TU/e. (n.d.-a). *About Atlas Living Lab*. Retrieved June 9, 2020, from https://www.tue.nl/en/research/research-institutes/top-research-groups/intelligent-lighting-institute/infrastructure/about-atlas-living-lab/

TU/e. (n.d.-b). *Atlas building: BREEAM hails Atlas the world's most sustainable education building*. Retrieved June 11, 2020, from https://intranet.tue.nl/en/university/services/real-estate-management/tue-campus/atlas-building/

TU/e. (n.d.-c). *Real Estate Strategy Campus 2030*. Retrieved June 5, 2020, from https://www.tue.nl/en/tue-campus/discover-your-campus/building-and-development/real-estate-strategy-campus-2030/

TU/e. (2020a). *Plenary sessions project organization*. Retrieved June 9, 2020, from https://intranet.tue.nl/universiteit/diensten/dienst-huisvesting/tue-campus/renovatie-gemini/plenaire-sessies-projectorganisatie/

TU/e. (2020b). *Zero Emission Laboratory (ZEL).* Retrieved June 9, 2020, from https://intranet.tue.nl/en/university/services/real-estate-management/tue-campus/renovation-gemini/zero-emission-laboratory-zel/

Volk, R., Stengel, J., & Schultmann, F. (2014). Building Information Modeling (BIM) for existing buildings - Literature review and future needs. *Automation in Construction*, *38*, 109–127. https://doi.org/10.1016/j.autcon.2013.10.023

W3C. (n.d.-a). *Query*. Retrieved July 10, 2020, from https://www.w3.org/standards/semanticweb/query.html

W3C. (n.d.-b). *Semantic Web*. Retrieved July 9, 2020, from https://www.w3.org/standards/semanticweb/

Wagner, A., & Uwe, R. (2019). BPO : The Building Product Ontology for Assembled Products. In M. Poveda-Villalón, P. Pauwels, R. de Klerk, & A.-M. Roxin (Eds.), *Proceedings of the 7th Linked Data in Architecture and Construction Workshop - LDAC2019* (pp. 106–119). http://ceur-ws.org/Vol-2389

Wong, J. K. W., Ge, J., & He, S. X. (2018). Digitisation in facilities management: A literature review and future research directions*. Automation in Construction*, 92, 312–326. https://doi.org/10.1016/j.autcon.2018.04.006

Yan, W., Clayton, M., Haberl, J., Jeong, W., Kim, J. B., Kota, S., Alcocer, J. L. B., & Dixit, M. (2013). Interfacing BIM with building thermal and daylighting modeling. 13th Conference of International Building

Performance Simulation Association, Porceedings of BS2013, 3521; 3528. http://www.ibpsa.org/proceedings/BS2013/p_1374.pdf
Zhang, Y., Kasahara, S., Shen, Y., Jiang, X., & Wan, J. (2019). Smart contract-based access control for the internet of things. IEEE Internet of Things Journal, 6(2), 1594–1605. https://doi.org/10.1109/JIOT.2018.2847705

## Appendix

# IFC evaluation - Zero Emission Lab

## Project information

Name Zero emission Lab (ZEL)
Location Gemini-Noord Building
Owner Eindhoven University of Technology

**Evaluated models**
1839_arch_020
20181214-VS-DO-TUe_ZEL_INS

**Referece documents**
48038.00_BIM protocol Ontwerpfase
48038.00_Informatie levering specificatie Ontwerpfase
Bijlage 2_BIM Project Informatieleveringsspecificatie

**Software used**
Solibri model checker
BIM Vision 2.19
Notepad++

**1. Content Description**

This topic provides a short description of the referenced documents and the content of the evaluated models.

**48038.00_BIM protocol Ontwerpfase**

This document is an agreement between TU/e and the company Valstar Simonis B. V. regarding how the BIM model is going to be developed based on the *Nationaal Model BIM Protocol.* It contains obligations and liabilities for both stakeholders, intellectual property rights, what will be delivered per project phase, and how these deliverables are going to be assessed.

**48038.00_Informatie levering specificatie Ontwerpfase**

This document is part of the BIM protocol, it deals with general and specific matters of the BIM model. Namely, Project information, coordinates system, file formats, name conventions, levels definition, applicable standards, deadlines for the delivery of the models, and clash control performance.

**Bijlage 2_BIM Project Informatieleveringsspecificatie**

This document is the annex of the *Informatie levering specificatie*. It is divided into two parts. The first details the property sets that are going to be included in the elements. These property sets include the property's name, short description, unit (when applicable), and the phase they appear in the model. The second part brings the definition of the level of development (LOD) and defines the LOD of all elements in the model organized per discipline and per phase.

**Models**

The IFC model 1839_arch_020 contains the architectural elements of the ZEL and its structure, at an architectural level, including its connection with the Gemini-Noord Building (Figure 1). The model 20181214-VS-DO-TUe_ZEL_INS comprises the mechanical, electrical (except cables) and plumbing installations of the laboratory, including emergency lighting fixtures and fire-related installations (Figure 2).

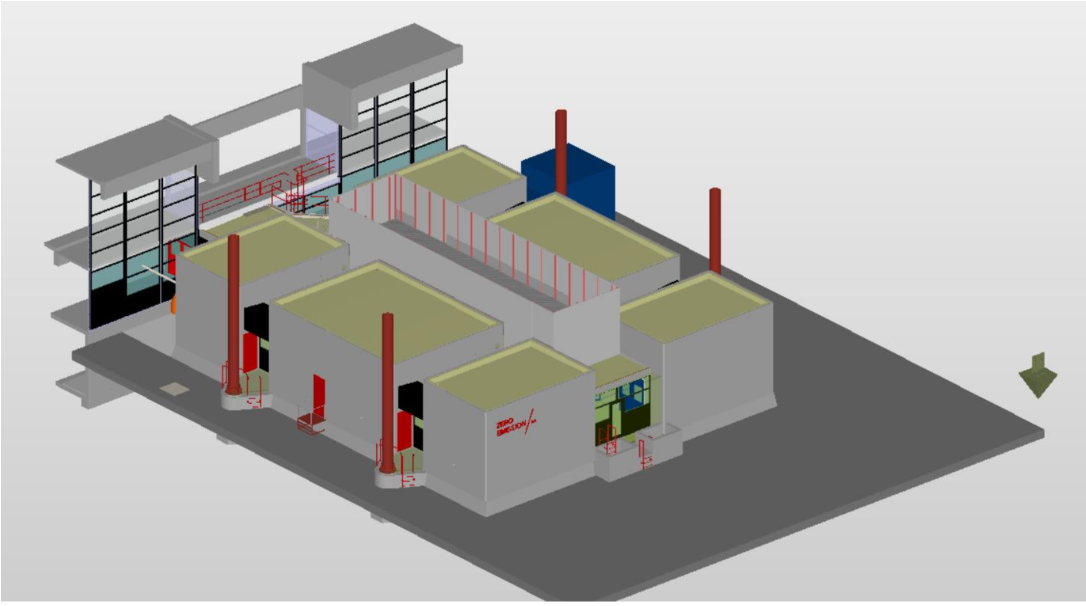


Figure 1. Overview model 1839_arch_020

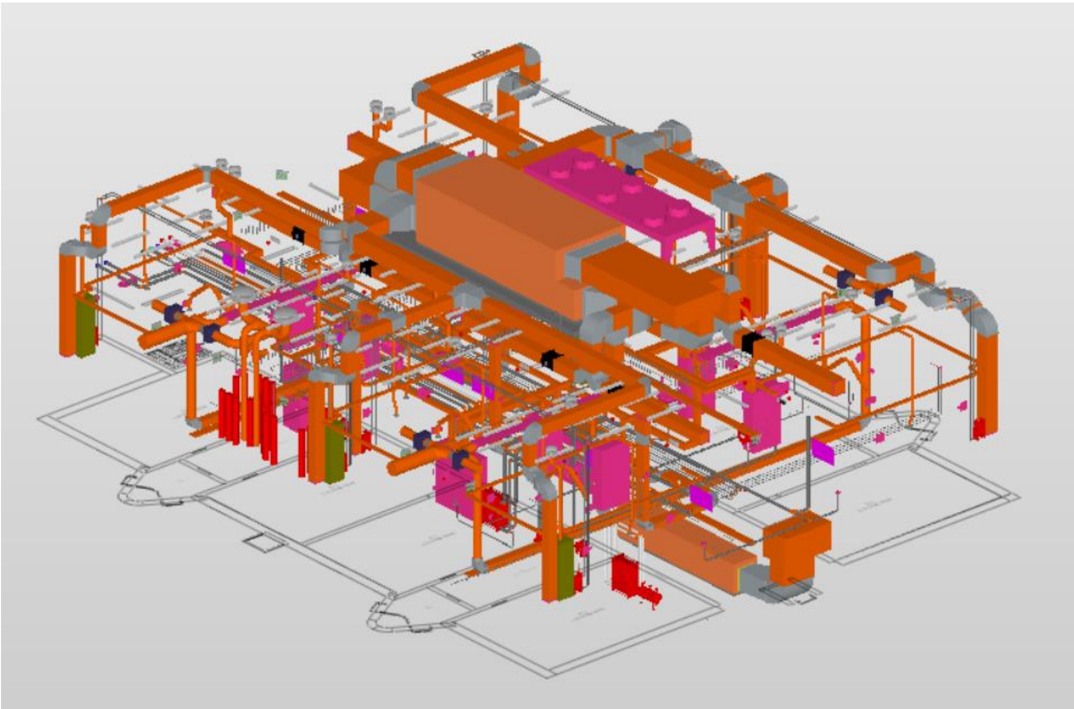


Figure 2. Overview model 20181214-VS-DO-TUe_ZEL_INS

The Building is spatially divided into 5 levels represented as building stories. These levels are -2 Keldervloer (Figure 3), -1 fundering (Figure 4), 00 begane ground (Figure 5), 01 laag dak (Figure 6), 02 hoog dak (Figure 7). Only in the architectural model, these building stories are subdivided into spaces. The project's coordinates are physically represented in the model by a generic object with the VAA logo, which is an ifcSite entity.

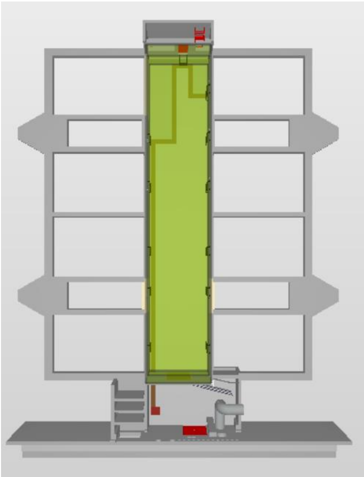


Figure 3. Level -2


Figure 4. Level -1

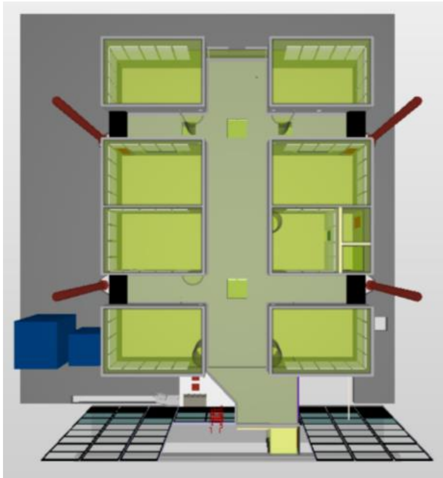


Figure 5. Level 00

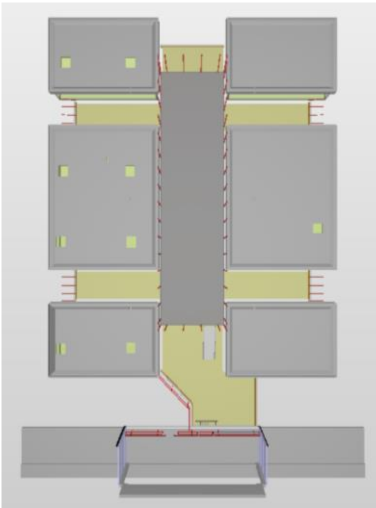


Figure 6. Level 1

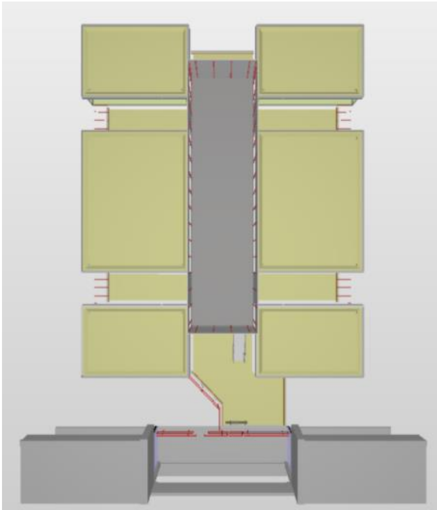


Figure 7. Level 2

These models can be combined to show the installations in their actual place. Figures 8, 9, 10 represent both models together in the basement, ground and roof level.

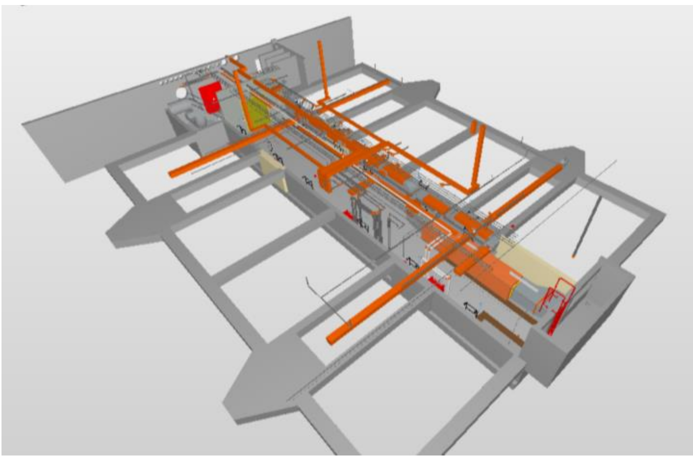


Figure 8. Installations in -2 level

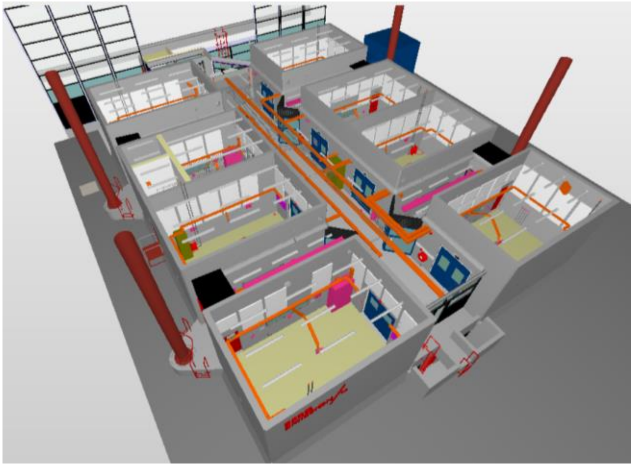


Figure 9. Installations in level 00

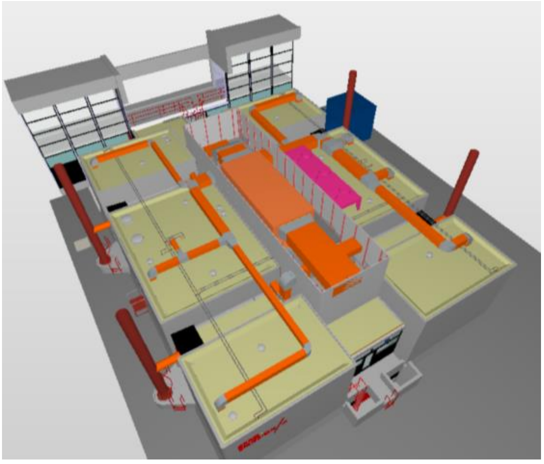


Figure 10. Installations in level 2

The MEP model is divided in 72 mechanical and plumbing systems. Namely, Aanzuiglucht LBK (2), Toevoerlucht LBK (2), retourlucht LBK(1), Afblaaslucht LBK (1), Argon (1), Acetylene (1), Methaan (1), Waterstof (1), Syngas H2/CO (1), Perslucht (2), Drinkwater (3), aardgas (1), Luchtbehandeling retour (4), Retourlucht puntafzuiging (8), Retourlucht gassenbord afzuiging (2), vuilwatervoer (1), Afblaaslucht opslagkast (1), Afblaaslucht gashokken (1), Afblaaslucht gasopslagkast (3), GKW aanvoer (1 warmtepomp, 1 laserkoeling, 1 LBK's), GKW retour (1 warmtepomp, 1 laserkoeling, 1 LBK's), CV aanvoer warmtepomp (1), CV retour warmtepomp (1), Stikstof (1 FB, 1 HD, 1 LD), zuurstof (1), Menggas (1 40%-02/N2,1 40%-H2/He,1 400ppm-NO/N2, 1 16%-co2/n2, 1 150ppm-HC/N2, 1 4500ppm-CO/N2, 1 18%-02/n2), helium (1), hemelwaterafvoer (1), Alternatieve brandstof (1), Benzine (1), Diesel (1), Synthetische lucht (1), reoturlucht gassenbord afzuiging (4), bedrijfswater (1), reserve (5). Some of these systems are represented in figures 11, 12, 13, 14 and 15.

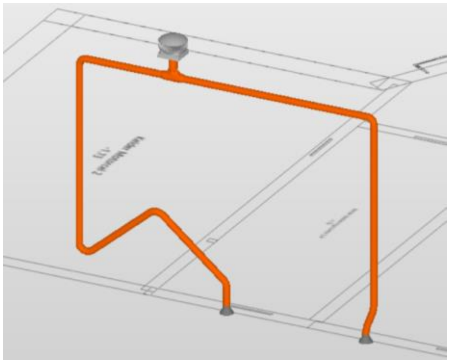


Figure 11. Air return point

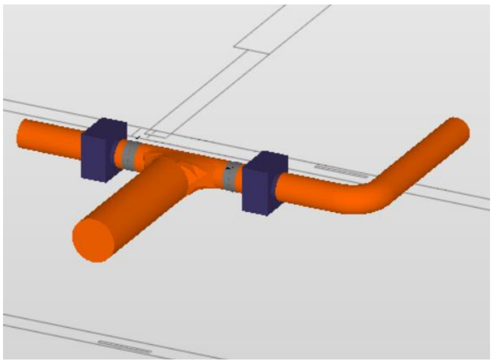


Figure 12. Air treatment return

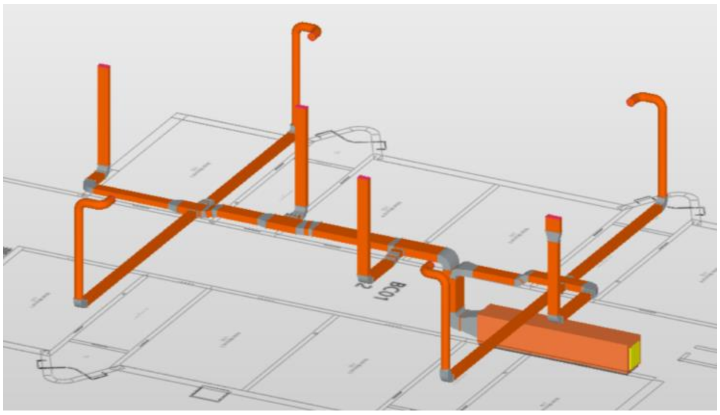


Figure 13. Air supply system

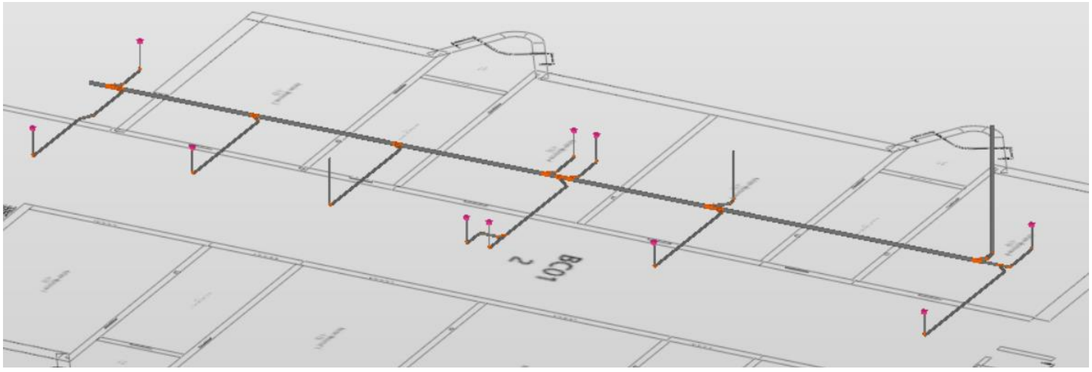


Figure 14. waste water system

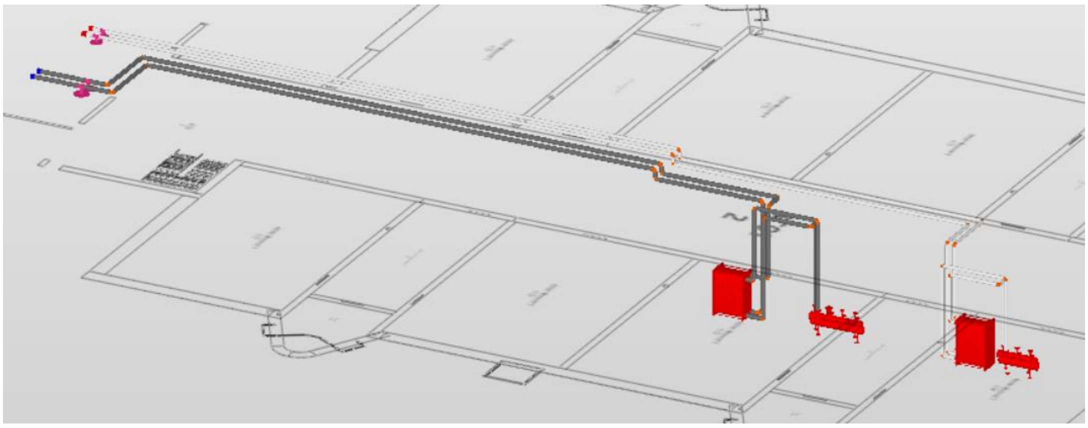


Figure 15. water pumps

The electrical system contains lighting fixtures (124 LED lights, 6 emergency lights, 14 emergency signs, 8 'ruimte niet betreden' signs and 1 Façade light), switches (15), Outlets (40 16A 400v, 26 CEE form 5p, 399 WCD 1 - voudig, 4 WCD 1 – voudig opb, 11 WCD 2 – voudig opb), cable conduits, and distribution cabinets (11). No information was found about the connection between these elements or their distribution in electrical circuits probably because the Philips Dynalite system will be implemented later in the building.

The models were evaluated regarding the properties associated with its elements and the classification system used, according to what was agreed in the information delivery specification. Moreover, it was evaluated the relationships included in the model and the existence of duplicated elements or intersections between them.

## 2. Elements' properties and classification system

The content of the information associated with the elements represented can only be assessed when there are clear requirements that were agreed between the stakeholders. For this project, these requirements are specified in the information delivery specification, which is an annex of the BIM protocol. It is divided into basic properties that are common to all elements and specific properties that are related to certain groups of elements, according to the project phase. The classification system used is defined in that document as well because it was included as property value and not as a classification reference.

However, it is stated in the document that it only applies to architectural and structural elements. Thus, there is no agreement in the properties to be included in the MEP model. Overall, it was checked if the property is present in the elements, with a valid value, and if it was included in the correct property set.

## 2.1 Generic properties

According to the *BIM project information delivery specification,* some properties are common to all elements in the model in the design phase. They are description, assembly code, name, type, and material. These properties are distributed in two property sets VAA_properties (description, assembly code) and Identification (name, type, material). However, a specific property set *identification* doesn't exist in the model. The name and type are attributes of the entity definition, and the material is established by associating the entity with a material specified in the IFC.

The Assembly code is the property where the STABU code of the element is written. The IFC model contains classification entities that are referenced to elements in the model but according to the document, what is not specified in it must be disregarded. Additionally, this evaluation verifies if the element has a STABU code but not if the code is correct.

**Assembly code and description**

Overall, the elements have an assembly code in the property set VAA_property. An opposite example is the elements that compose the load-bearing stair which connects the levels -2 to 0. The railing of the stair is described as beams and columns, none of them has an assembly code value (Figure 16 and 17).

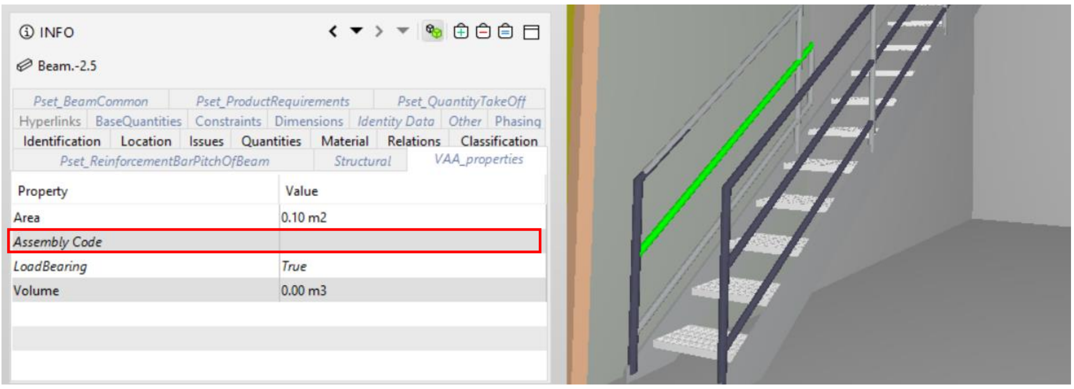



Figure 16. Beam in stair railing without assembly code

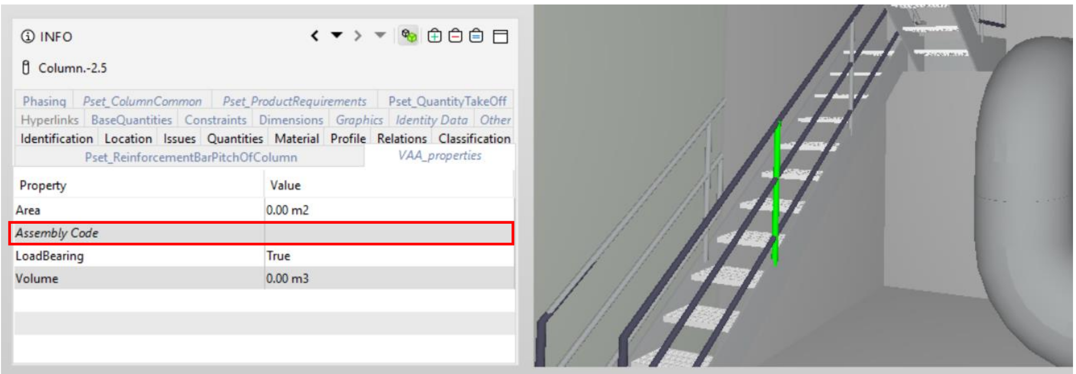



Figure 17. Column in stair railing without assembly code

The architectural model has 245 slabs, 7 discrete accessories, 55 walls, 217 columns, 40 fasteners, 2 members, 34 beams that don't have a STABU code. Generally, when an element is missing the STABU code it is also missing the description. The only exception is the slab GUID *2Mpy2JQwTE4eWqjmC37tzl* that has assembly code but no description and Slab GUID *2OC6h$ZTiL6u1ZBcpvagNZ* that has description but no assembly code.

**Name, type, material**

All elements have name and type attributes. On the other hand, the material property is missing in 12 beams, 13 railing, 1 stair, 4 walls, 1 pipe, and 16 plates. It could be that the material was not defined yet for some elements but there are elements missing material property when similar elements have it. An example is the railings to access the -2 level corridor (Figure 18 and 19).

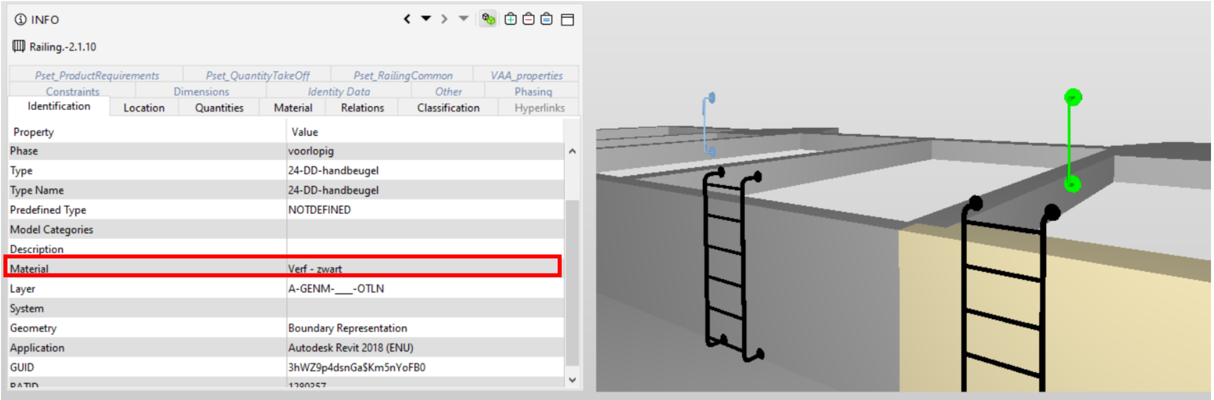



Figure 18. Railing in -2 level corridor with material described

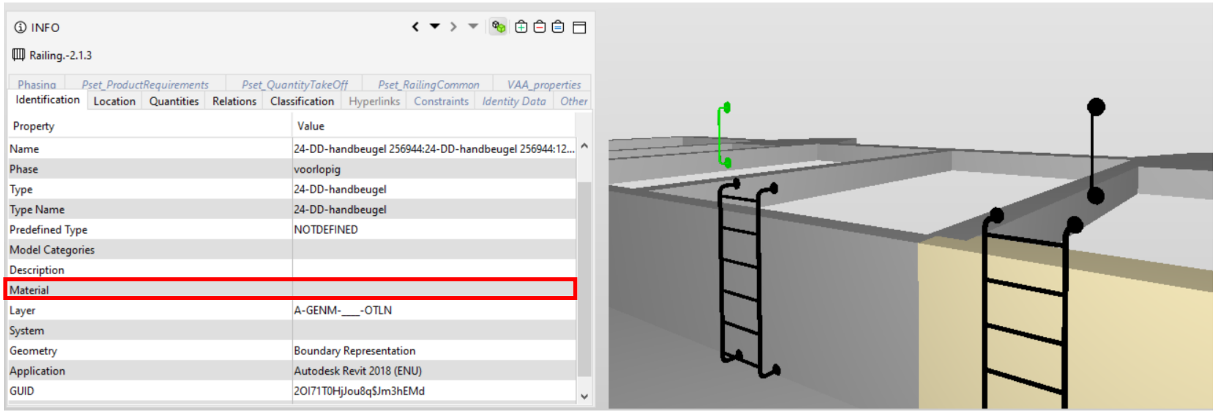



Figure 19. Railing in -2 level corridor without material described

**MEP model**

The requirements described in this document do not apply to the MEP systems. Nevertheless, for comparison, the MEP model was also evaluated for the same parameters. Similarly to the architectural model, name and type are present in all elements. No STABU code is available but the property assembly code exists in the property set *00_VS_Checklist-NL-sfb en IFC-export* in a large number of elements. However, the property value is the NL-sfb code. This information is in the model twice, the second time as a classification entity, and a reference to this classification. There is no description of property value, and the material property is only available for a small number of elements.

All elements in both models have global unique identifiers (GUID), even though it was not mentioned as a requirement.

## 2.2 Specific properties

Due to the scope of this work, two entities were investigated regarding their specific properties. Rooms and lighting fixtures.

The rooms are modeled as space and must contain the room's number (property set *Identification*), area, unbounded height, and perimeter (property set *Dimensions*). Regarding the lighting fixtures, there is no information about the properties that must be included in them. Hence the evaluation was backward by observing the model to verify its content.

**Number**

All spaces have a number as an attribute of the entity. However, the numbering system is not consistent throughout the project and they are not unique per space. For example *gang* in the -2 level has a number *-1T33*, on the same floor, *BC01* has number *5*, the same number used for one of the *ATEX zonering zone 2* on the ground floor, and the *kelder motorcel 7* has number *-1.74*.

Also, the number is used to artificially indicate some relationship between spaces. For instance, *Motorcel 3* has number 0.76, the two *Gashok* near it have number 0.76a and 0.76b. However, this relation is not modeled in the IFC file.

**Area, unbounded height and perimeter**

The property set dimension exists and is present in all spaces with the area and perimeter property values. The property unbounded height is missing in all ATEX zones and in more 6 spaces.

**Lighting Fixtures**

Beyond the common properties and the geometric representation, the other information added to the lighting fixtures was the element's reference. It is located in the Pset_DistributionFlowElementCommon property set. However, it merely repeats the property type's value. A smaller number of elements also have the manufacturer property in the Pset_ManufacturerTypeInformation. No numbering system is used to identify the lighting fixtures or any other equipment in the MEP model.

## 3. Relationships

The elements in the architectural model are associated with the building story level. There are two exceptions. The railings inside corridor -1T33 and the rolling shutter inside corridor 0.T32 which are associated with the spaces they are in. Additionally, the spaces are independent areas not organized in zones. The elements in the MEP model are also associated with the building story but the last top 2 levels have a different name from the architectural model. 01 laag dak is named 01 eerste verdieping, and the 02 hoog dak is named 02 tweede verdieping.

This connection to the building story can become difficult as sometimes elements start on a floor and continue on the next floor. This generates elements that share the same space to be classified on different floors. For example, Figures 20 and 21, where a wall is associated with level -1 and the door placed in it is associated with level 00.

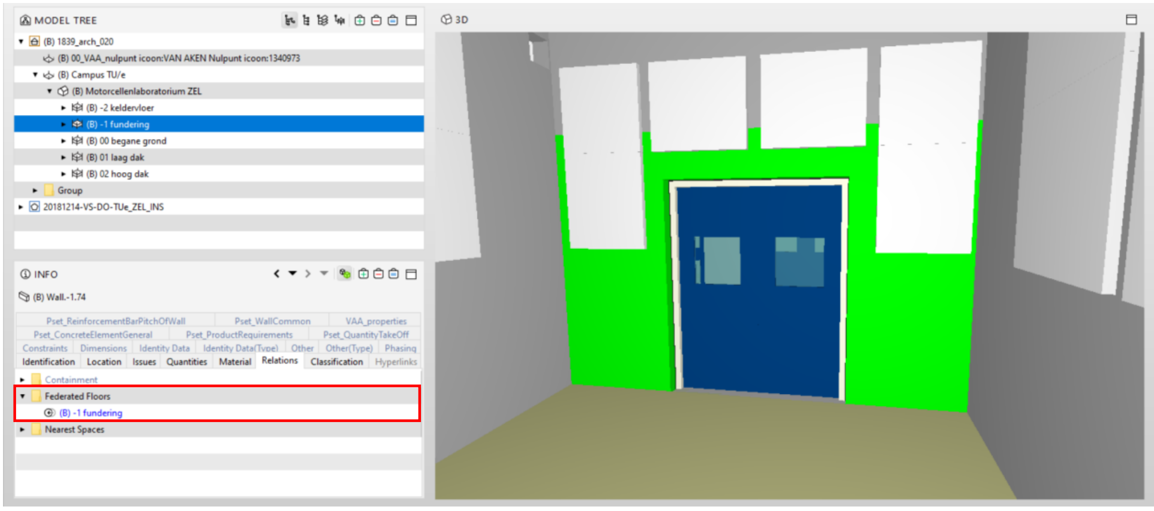


Figure 20. Element relationship to the -1 fundering floor

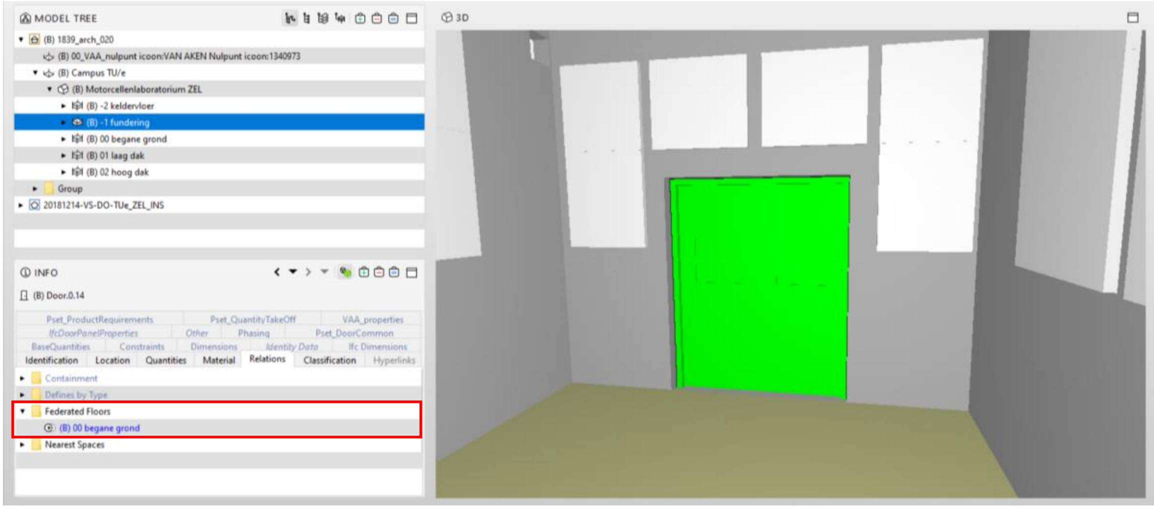


Figure 21. Element relationship to the 00 begane grond

Moreover, some elements are modeled in parts but there is no aggregation relationship between these parts. The best example is the stair that connects the levels -2, -1, 0. Every step is an independent element associated directly to the floor level (Figure 22 and 23).

Figure 22. Third stair’s step associated to floor level

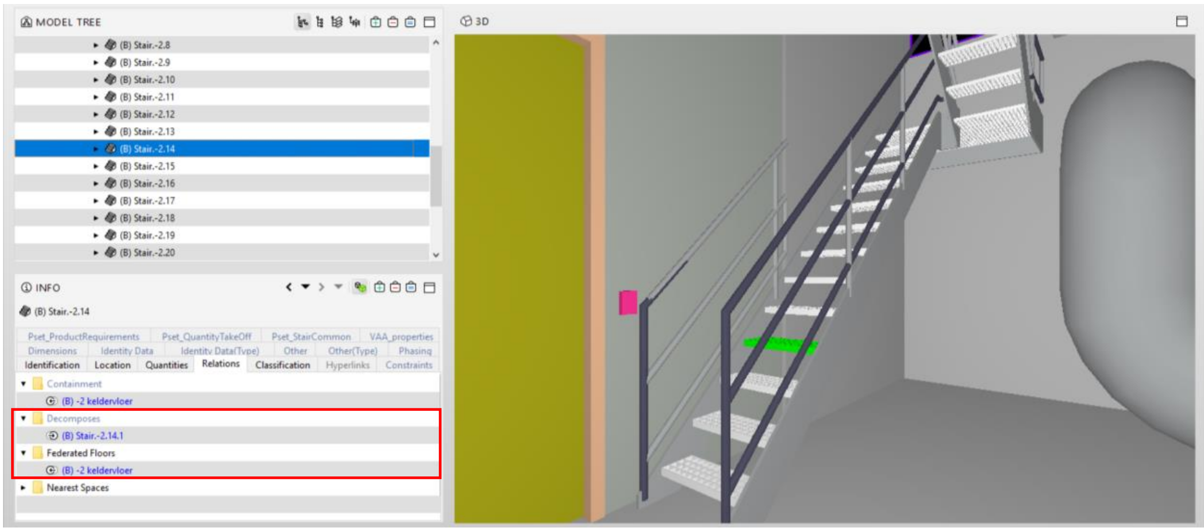


Figure 23. Fourth stair’s step associated to floor level

Some structures in the architectural model are organized in groups. That is the case of roof structures of the motor cells, façades, and the fire compartmentation spaces.

In the MEP model, mechanic and plumbing elements were associated in systems. Electrical panels with outlets were organized in groups but cable carriers and lighting fixtures, besides emergency signs and alarms are just associated with the floor.

## 5. Intersections and duplicated elements

The architectural model has fewer partial clashes between walls, slabs, and railings (Figure 24, 25, 26). It was identified eight minor clashes between walls, one clash between slabs, two between railings, one clash between a beam and a slab, and on between a slab and a wall. This could cause some discrepancies when extracting quantities from the model.

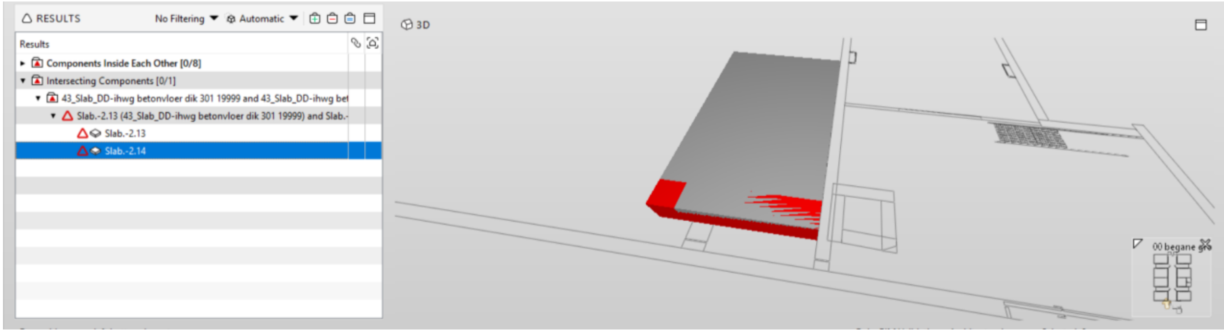



Figure 24. Intersection between slabs

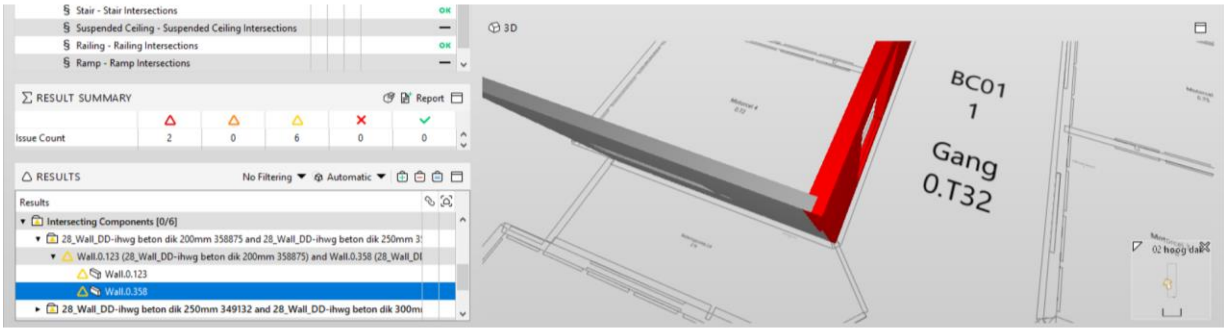



Figure 25. Intersection between walls

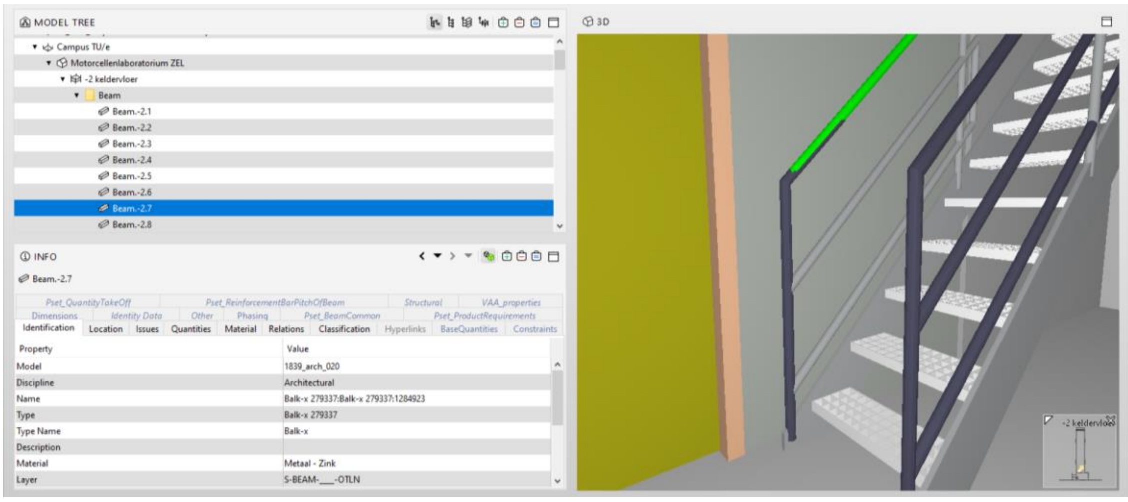



Figure 26. Intersection between railings

Similarly, the MEP model has only smaller partial clashes (Figure 27 and 28) and a duplicated cable carrier (Figure 29). It was identified 106 smaller clashes in the MEP model and one duplicated cable carrier.

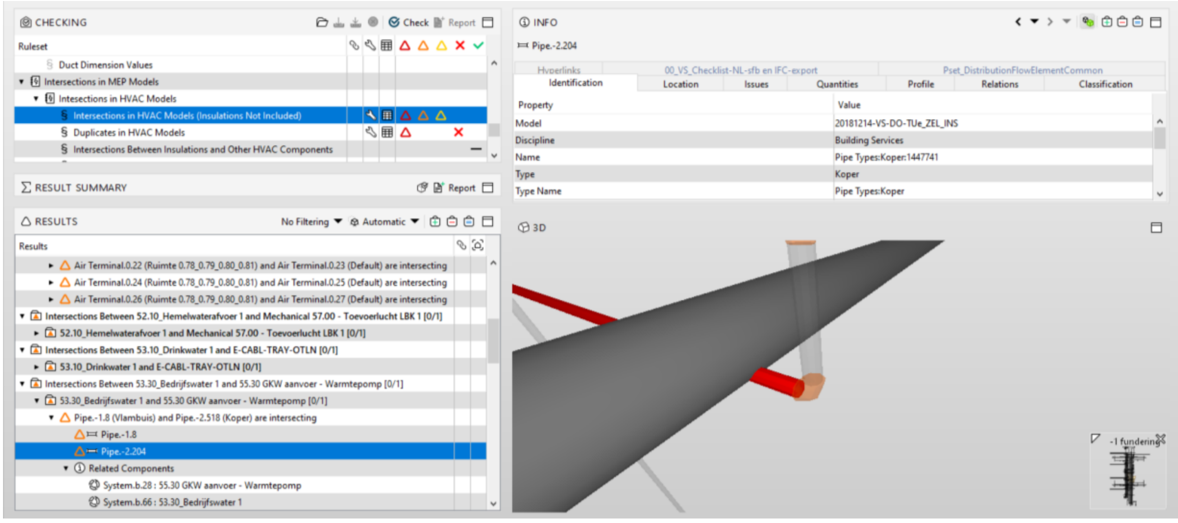


Figure 27. Intersection between two pipes

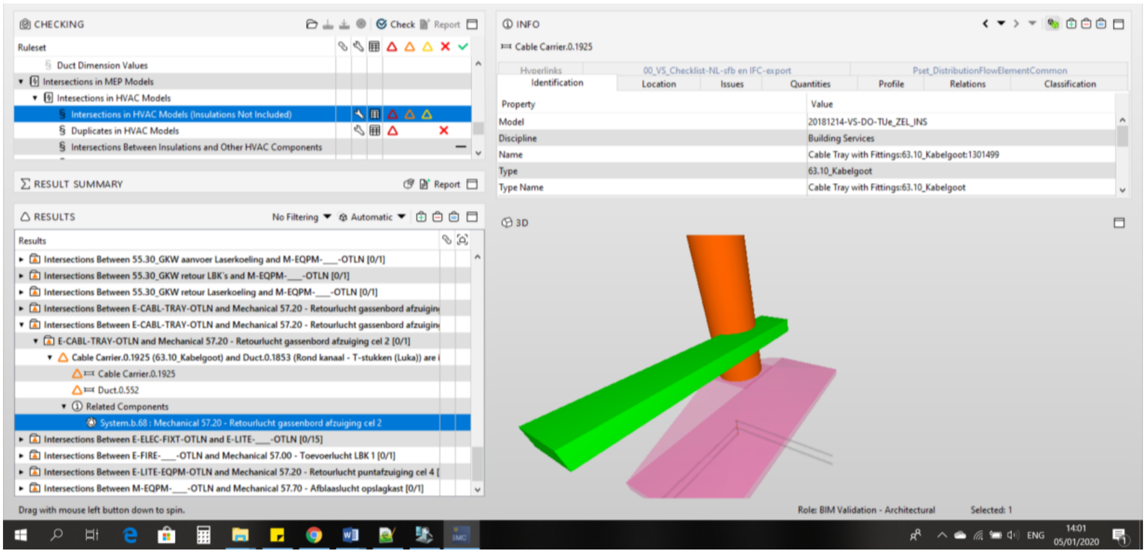


Figure 28. Intersection between pipe and cable carrier

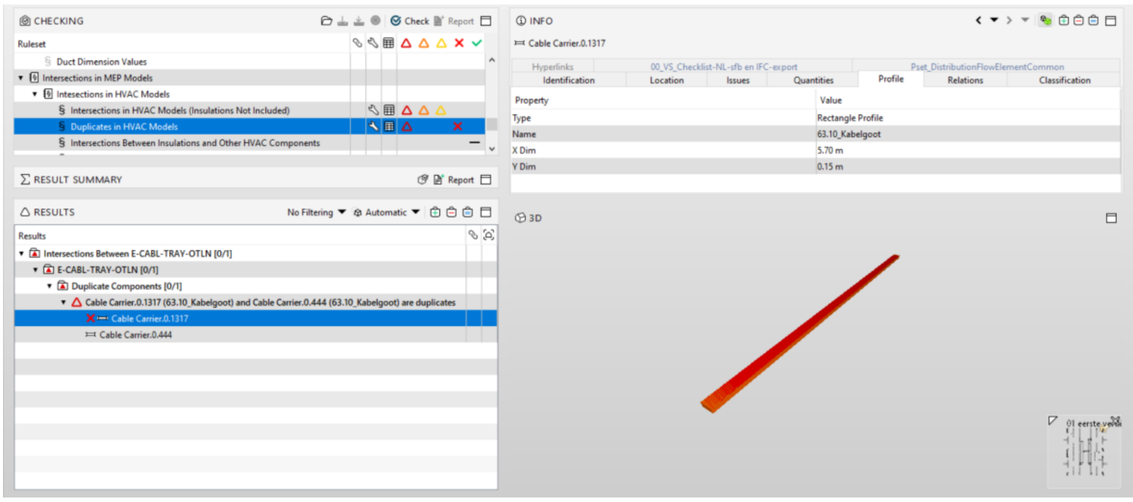


Figure 29. Duplicated cable carrier

There is no significant intersection between the architectural model and the MEP model. The identified intersections reflect the interaction between these models. For instance, a pipe that goes through a wall. The hole through which this element is going to pass through the wall is not modeled in the architectural model and hence it appears as a clash between the wall and the pipe.