Knowledge-based automatic OpenBIM data exchanging framework supporting holistic building design

Ali Sabah Khudhair

School of engineering

Cardiff University

A thesis submitted to Cardiff University for the degree of Doctor of Philosophy 2022

Acknowledgements

I express my heartfelt gratitude first and foremost to my supervisors, Professor Haijiang Li and Professor Hanxing Zhu, without whose helpful support, this thesis would not have been possible. Particular thanks must go to Prof. Haijiang Li for his constant advice and guidance to keep this research on the right track and support me to keep the big picture in the future.

I also must acknowledge the support of the BIM innovation team members and the entire BRE Institute for Sustainable Engineering, including academics and fellow PhD students.

This thesis would equally have been impossible if not for the support extended to me by my family. During my struggling time, the encouragement and support I received from my family have been deeply flattering. Thanks for providing frequent, kind distractions to keep me sane throughout these years.

Again, I appreciate all the people that support my research.

Ali Khudhair

2022

Abstract

For complex building design, there are many factors required to be considered in order to achieve the best possible design. During the process, there are many different professionals using different tools, along with different information; it is therefore very difficult to enable efficient collaboration and decisions. Through a comprehensive literature review, the dissertation identifies that data exchange within the BIM context is still facing interoperability issues due to inconsistency in defining data exchange requirements. Moreover, the decision-making within the BIM context focuses on individual aspects, which are developed separately from each other and require end-users to have prior knowledge about other domains. Despite the effort to develop a multi-objective decision-making knowledge base, most research did not provide a method that can work in parallel with a BIM model, which can provide decisions based on the data collected from the BIM model automatically. Most methods require manual input to process the data in a BIM model. It hence requires a smart way to leverage diverse information sources to work together.

Based on that, the dissertation determines the research motivation and formulates detailed research questions and hypotheses to establish an automatic data exchanging framework that combines both data exchange method and semantic web technology to eliminate inefficiencies in data exchange and improve the decision-making in the early design stage. A common data analysis (CDA) referencing various concepts such as the standardised Information Delivery Manual (IDM), model view definition (MVD) and the concept of the semantic intersection was designed to conclude “single truth of information” and “partial truth of information” data sets that form the basis for the proposed framework from a data processing perspective. Furthermore, the requirements needed for multi-objective knowledge were also investigated.

Following the analysis, firstly, a data exchange method that can extract the critical data from a BIM model based on the IFC schema was implemented. Secondly, a multi-objective knowledge base is constructed, which can assist engineers who lack knowledge associated with sustainability and cost in comparing different design choices while considering design conditions to develop an ideal design in the early stage. The main outcome of the research lies in providing a multi-objective knowledge base that can closely connect with real-project data. Consequently, an automatic data acquisition method was developed to align the proposed knowledge base with the data exchange method to extract data from an IFC file and merge them with the data presented in the developed ontology automatically in order to eliminate the human involvement by decreasing manual input.

It was shown that the proposed framework could provide different data sets and process the IFC-based BIM model correctly without data loss. The data acquisition method helped produce a more dynamic knowledge base that connects real project information to static information related to cost and sustainability efficiently. Consequently, this approach is proved to be more efficient than a manual approach by adding data to the knowledge base. The SWRL rules helped automate all the manual calculations and generate new facts based on the data in an IFC file. The built-in rules allow the end-user to review and compare different design alternatives by considering various factors at an early stage. All the developed tools and functions were tested and went through framework validation. This combined framework is unlike previous approaches where data and instances are entered manually one by one. It presents a more direct way to work with IFC-based BIM models in order to evaluate various aspects.

Contents

List of Figures

List of Tables

List of Abbreviations

AEC = Architectural Engineering and Construction

NBS = National Building Specification

BIM = Building Information Model

CAD = Computer Aided Design

IFC = Industry foundation classes

CDE = Common Data Environment

API = Application Programming Interface

WOS = Web of Science Database

IFD = International Framework for Dictionaries

DSR = Design Science Research

IDM = Information Delivery Manual

MVD = Model View Definition

OWL = Web Ontology Language

URI = Uniform Resource Identifier

IfcOWL = Industry Foundation Classes Web Ontology

RDF = Resource Description Framework

RIBA = Royal Institute of British Architects

SWRL = Semantic Web Rule

SQWRL = Query-Enhanced Web Rule Language

AI = artificial Intelligence

ML = Machine learning

BC = blockchain

DA = Data analytics

CC = Cloud Computing

IoT = Internet of Things

LS = Laser scanning

bSDD = BuildingSMART Data Dictionary

GUI = Graphical user interface

List of Publications

Ali Khudhair, Haijiang Li, 2019." Knowledge-driven holistic decision making supporting multi-objective innovative design". In the book "Industry 4.0 – Shaping the Future of The Digital World", Proceedings of the 2nd International Conference on Sustainable Smart Manufacturing 2019.

Ali Khudhair, Haijiang Li, Ren Guoqian, and Song Liu. 2021. "Towards Future BIM Technology Innovations: A Bibliometric Analysis of the Literature" Applied Sciences 11, no. 3: 1232. https://doi.org/10.3390/app11031232

Ali Khudhair, Haijiang Li, Thomas Bower, Guoqian Ren. "A theoretical holistic decision-making framework supporting collaborative design based on common data analysis (CDA) method". Journal of Building Engineering, Volume 46, 2022, 103686, ISSN 2352-7102, https://doi.org/10.1016/j.jobe.2021.103686.

Ren Guoqian, Haijiang Li, Song Liu, Jaliya Goonetillake, Ali Khudhair, Steven Arthur. "Aligning BIM and ontology for information retrieve and reasoning in value for money assessment". Automation in Construction, Volume 124, 2021, 103565, ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2021.103565.

Chapter 1. Introduction

1.1. Problem statement

The construction industry is changing its traditional business methods, with information being exchanged digitally rather than in paper form. However, moving forward with this digitalisation within the AEC industry requires companies to adopt new techniques and technologies to help them collaborate more effectively and enhance the decision-making process within their environment, especially in the early design stages.

BIM has been utilised to provide better collaboration and integration in a project (Costa and Madrazo, 2015; Ebrahim P Karan and Irizarry, 2015). It is described as "a data-rich, object-oriented, intelligent and parametric digital representation of the facility, from which views and data appropriate to various users' needs can be extracted and analysed to generate information that can be used to make decisions and improve the process of delivering the facility" (America, 2010). However, interoperability, which is described as the ability to exchange data seamlessly across various disciplines and stakeholders (Venugopal, Charles M. Eastman and Teizer, 2015a), has been acknowledged as an important issue in BIM due to heterogeneous tools (Grilo and Jardim-Goncalves, 2010; Sun et al., 2017). Consequently, the BIM model needs to be transferred to a model that can be understood by other design and analysis tools and also need to be supported by a decision-making method to produce an ideal design (Bahar et al., 2013).

The information delivery process plays a significant part in enhancing collaboration by identifying what and when information needs to be exchanged and who is responsible for that information (Schapke et al., 2018). However, the data exchange process faces several concerns that restrict its progress, mainly related to data mapping, which requires an informed understanding of various disciplines and domains. Many BIM models are created in a given project by various stakeholders. Each of these models represents an individual subset of the entire building, and it is called a domain-specific partial model (Preidel *et al.*, 2018). Despite the different use of these sub-models, they share some commonalities that are not exclusive to a specific domain. However, although several attempts have been made to develop data exchange requirements for BIM models, there is still a lack of homogeneity since no methods for classifying and sharing those requirements are clearly outlined (Pauwels, Zhang and Lee, 2017).

Furthermore, decision-making has mainly focused on individual aspects such as economic, environmental, safety, etc. Currently, these aspects are developed separately from each other and require end-users to have prior knowledge about other domains. Consequently, the decisions are based most of the time on the offered information and engineering knowledge about the targeted subject. However, for complex building design, there are many factors required to be considered in order to achieve the best possible design. During the process, many different professionals use different tools/models, along with different information; therefore, it is very difficult to enable efficient collaboration. However, providing a linked design as much as possible can save a lot of work and effort among disciplines and design teams.

Despite the effort to develop a multi-objective knowledge base within the BIM context, most research did not provide an approach that can work in parallel with a BIM model automatically. Most of the research requires manual input to process the data in a BIM model. Hence, a smart way is required to leverage diverse information sources/ models to work together, e.g., through automatic information exchanging supported by a holistic knowledge base.

1.2. Research motivation

OpenBIM concept is a "universal approach to the collaborative design, realisation, and operation of buildings based on open standards and workflows" (Choi et al., 2016). It came as an expansion for BIM to enable Interoperability (openBIM - buildingSMART International, no date). It has commenced the path for innovative technologies and concepts to merge within the AEC industry.

The development of OpenBIM concepts is not focused only on the technical aspects. Instead, it covers the joint efforts issues, including the legal, semantic, and organizational aspects, to reach a high level of collaboration and integration. Succar described BIM as “a set of interacting policies, processes and technologies” (Succar, 2009). Technologies include technical developments such as software, hardware, equipment, and networking system. Processes represent human resources and the stakeholders involved during the whole project lifecycle. Not to forget to mention the policies, which correspond to the documentation part and guidelines such as research centres' contracts and regulations. Hence, the topic of BIM should be handled at the digital transformation level since it is not just technological adoption but the integration of technologies, processes, and people. Thus, the OpenBIM ecosystem can be described as a system of technologies, processes and policies that should network and dynamically operate together to embrace continuous changes in the AEC industry.

Data exchange and decision-making play a major factor in the development process of the BIM ecosystem. However, they have been facing several difficulties. Consequently, in this research, the OpenBIM concept and the concepts backing its development are investigated by reviewing the existing concepts, technologies, tools, standards, and available online resources to see where the AEC industry is currently in terms of OpenBIM developments, interoperability, and decision making.

1.2.1. Improving data exchange through BIM and IFC Standard

The massive growth of information, which is compiled from different design tools, has triggered challenges in data exchange and held back decisions in a project. Lee (2011) mentioned four main issues behind data exchange: (1) Incomplete coverage of a data model, (2) issues raised with using translators due to the lack of guidelines while developing these tools, (3) system errors due to the use of various vendors tools inside the same organisation, (4) Software domain complications since the used tools were developed for a specific domain and lack of knowledge about other domains. The above issues show that the data exchange within BIM is still facing difficulties in reaching a high level of improvement.

There are three types of data exchange (Aldegeily, Hu and Ph, 2018; Ramaji and Memari, 2018): First, data exchange using the same authoring tools. Secondly, exchange through an application programming interface (API) and thirdly, exchange using a common data schema such as industry foundation classes (IFC). Since the IFC schema covers various domains, it is not convenient to implement the entire schema in software vendors. Consequently, a concept such as Model View Definition (MVD), which is described as a subset of the IFC schema to specify the requirements of the exchange data to serve a specific domain (Eastman et al., 2010), has been used as a solution to enhance data exchange. However, Lee (2009) stated that MVD is a document rather than a subset that describes how the IFC model specification is applied to data exchange between different application types. Moreover, the data exchange requirements that were defined using the above concept were developed independently and require end-users to have prior knowledge about other domains

(Lee, Eastman and Solihin, 2016). Lai and Deng (2018) stated that although many pieces of research have been presented in the area of data exchange and MVDs, several issues still exist within these topics since data exchange within the BIM context has been mainly focused on specific disciplines.

Cheung et al. (2012) indicated that “In computer applications, it is rather common for users to employ one tool to deal with a type of task and another tool for a different type of task even though the two tools may have overlapping functions to handle both tasks” (Cheung et al., 2012). Hence, various stakeholders create many BIM models in a given project to achieve different objectives. Despite the different use of these models, they share some commonalities that are not exclusive to a specific domain. The more commonalities exist within two systems, the less data loss will exist between the two systems (Gielingh, 2008). However, Lee (2011) stated that Information generally flows from the more informed to the less informed. Taking into account, it can flow in the opposite direction in certain situations. Hence, the stream of information is not all the time unidirectional.

Even though several attempts have been made to develop data exchange requirements for BIM models, there is still a lack of uniformity while developing those requirements (Lee, Eastman and Solihin, 2016). Therefore, this research presents a common data analysis method (CDA) referencing various concepts, such as the standardised Information Delivery Manual (IDM), MVD and the concept of semantic intersection, to conclude “single truth of information” and “partial truth of information” data sets that form the basis for an automatic data exchanging framework in order to enhance collaboration and the decision-making process.

1.2.2. Enhancing decision-making through a linked data approach

The decision-making process contributes significantly to project success. The design stage requires decisions concerning conceptualisation, modelling, analysis, designing, detailing and cost estimation. These decisions are made to help designers and stakeholders to develop an ideal design with less effort and time. Since some decisions rely primarily on other disciplines, the lack of knowledge from stakeholders about other disciplines can slow down the decision-making process. For instance, architects and structural engineers contribute significantly to the decisions made within a project (Østergård, Jensen and Maagaard, 2016). However, some engineers, such as structural engineers, may lack knowledge about some areas, such as sustainability or cost, which can affect the decisions made in a project.

For example, sustainability plays an essential role in today's infrastructure. The construction of buildings in the United Kingdom utilises over 40% of the country's energy and releases about 330 million tonnes of carbon dioxide (Mockienė, Keras and Gilys, 2015). The research has revealed that the embodied carbon dioxide in building materials, especially concrete has a significant impact on the environment. It can, directly and indirectly, affect other aspects such as the cost of the building and design safety. Consequently, linking different data sources that come from different aspects can help in obtaining the best feasible solution at the early stages of the design by looking into several factors holistically, especially since having more than one design solution is possible most of the time. Hence, structural engineers can reduce the carbon content in a project and the project's cost by assessing alternative construction materials while considering design criteria.

Moreover, IFC, which is a rich BIM schema for data exchange, and known as the industry standard for interoperability (Amor, 2015), was not designed to determine new information from a BIM model. It was designed to deliver information to end-users without any reasoning functionalities. Sherif, Jinkook and Chuck stated that " although more cost estimating applications are moving toward IFC compatibility, IFC does not solely cover all components required to generate an estimate, as estimating requires not only quantity take-off data, but other types of associated databases" (Sherif, Jinkook and Chuck, 2011). Consequently, IFC needs to be supported by other technologies or formats to enhance its performance, which has necessitated utilising technologies from computational areas to work with BIM models. A study by Pauwels et al. (2017) showed that the semantic web and Linked Data have the potential to contribute to applications that involve information from various disciplines. For instance, the semantic web has several features (Ren, Ding and Li, 2019): (1) It provides a framework and language for designers to organise and represent information in a human and machine-understandable format. (2) It establishes a hierarchical structure of the concepts in a particular domain and describes the connections between these concepts. Consequently, it can be used to align concepts from different AEC disciplines and enhance the IFC format performance.

The semantic web has been applied in the area of cost estimation (Lee, Kim and Yu, 2014; Liu, Lu and Al-Hussein, 2016; Abanda, Kamsu-Foguem and Tah, 2017; Niknam and Karshenas, 2017). It has been used for enhancing energy management (McGlinn *et al.*, 2017), building evacuation design (Boje and Li, 2018), improving coordination and communication between engineers by recognising conflicts in the BIM design process (Liu, 2013), and safety in facility management and maintenance (Wetzel and Thabet, 2015). It has also been utilised to support environmental monitoring and compliance checking among different information systems (Zhong *et al.*, 2018). Consequently, technology such as the semantic web has the potential to improve interoperability within BIM models by implementing domain knowledge into the BIM model, which can provide semantic enrichment of the BIM model. However, most of the research was developed separately to serve a single objective decision. There is a lack of a multi-objective knowledge base within the BIM context. Furthermore, most of the research did not provide a technique that can work in parallel with a BIM model, which can automatically provide decisions based on the data collected from the IFC-based BIM model. Most of the research requires manual input to process the data in a BIM model. Therefore, in this research, the proposed data acquisition method can link the IFC-based BIM model with the proposed holistic knowledge base, which can help stakeholders compare different design choices.

1.3. Research hypothesis

In light of the research problems and motivation identified above, this research aims to develop a data exchanging framework that combines a data exchange method and a semantic web approach to eliminate inefficiencies in data sharing and improve decision-making in the early design stage. The overarching aim and hypothesis adopted in the research are as follows:

To establish an automatic data exchanging framework that orchestrates different functions holistically through automatic information exchange supported by an ontological approach for holistic spatial co-ordination building design.

1.4. Research questions

Based on the research objective in this thesis, the research hypothesis is broken down into the following research questions:

Q1: What are the concepts, technologies, and tools existing within the BIM ecosystem to improve interoperability and decision making in the AEC industry? And how are those concepts and tools backing BIM development while considering their scope and limitations? - **Chapter 2**

Q2: What is required to identify a data exchanging framework to support the collaborative design and decision-making from a data processing perspective? - **Chapter 4**

Q3: What needs to be considered to build a data exchange method to convert from A model to B model in order to automatically realise data exchange? - **Chapter 5**

Q4: What needs to be considered for designing a holistic knowledge base that considers various aspects such as design conditions, sustainability, and cost to support building design? - **Chapter 5**

Q5: Can the holistic knowledge base be aligned with the data exchange method to provide an automated framework? - **Chapter 5**

Q6: Can the proposed framework provide the necessary information automatically and at the same time help end-users to compare different design choices related to sustainability and cost factors while considering design conditions based on the existing data in an IFC-based BIM model? - **Chapter 6**

1.5. Structure of the thesis

This thesis is divided into several chapters, each pursuing answers to the main research questions.

Following the introduction in Chapter 1, Chapter 2 aims to answer the first research question by introducing a thorough literature review that is relevant to the research topic. Section 2.1 elaborates on the BIM concept in terms of BIM maturity and the benefits and limitations of BIM in the AEC industry. In Section 2.2, several technologies were investigated in relation to BIM. Section 2.3 discusses BIM-level interoperability and the work that has been accomplished to deliver a high level of data exchange in the AEC industry. Finally, Section 2.4 discusses the semantic web and multi-objective knowledge base. However, due to the large-scale of BIM topics and the focus of this research, this literature is by no means an exhaustive review. However, it indicates the many developments taking place in this area and their limitations. At the end of this chapter, the main findings of the review, which are closely related to the research gaps, are given.

Chapter 3 presents the overall arching methodology through which this research was carried out to clarify the principles and methods utilised in this research. The main methodology adopted in this research is Design Science Research (DSR) (Johannesson and Perjons, 2014), which is usually utilised in categories of artefacts referring to engineering and computer science disciplines to solve a generic challenge experienced in practice (Johannesson and Perjons, 2014).

Chapter 4 is related to the required information and data needed to develop a data exchanging framework to support holistic spatial co-ordination building design and thus aims to answer the second research question. A CDA referencing various concepts such as the IDM method, MVD and the concept of the semantic intersection was designed to understand for each profession what sort of data is required and what information needs to be exchanged. The method implemented in this chapter is an expansion of the literature combined with a hands-on method by analysing several BIM models from a data perspective. The findings in this chapter are used as a base for developing and implementing the proposed framework.

Chapter 5 defines the technical contribution of this research, which is meant to answer research questions Q3, Q4 and Q5. A data exchanging framework that combines a data exchange method and semantic web technology is implemented. This chapter is divided into two main parts: to answer the third research question, Section 5.1 shows the definition and implementation of the data exchange method. A simple tool is also developed to validate the developed method in this section. For answering the fourth and fifth research questions, Section 5.2 discusses the development and implementation of the multi-objective knowledge base in addition to the automatic data acquisition method proposed between the developed ontology and the data exchange method in an effort to extract information from an IFC file and merge them with the proposed ontology automatically.

Chapter 6 addresses the testing and validation of the proposed framework in response to the sixth research question. After conducting the technical developments in chapter 5, the intended framework, including its functionalities, is proposed to be validated using a complex IFC-based BIM model to obtain accurate results based on the designed scenarios. The use case scenario proposed in this research was designed as close as possible to reflect a real-life situation. The validation process was carried out to prove that the proposed framework is functional and reliable for data exchange and holistic decision-making. Section 6.1 discusses the BIM model, which was used for the validation, and the objectives of the scenario-based testing. In Section 6.2, the data exchange method and the developed holistic knowledge base are tested to check consistency and ensure no data loss is given. The ontology reasoner pellet, a plugin function included in protégé, was utilised to check that the developed ontology is syntactically correct. Moreover, the data acquisition method was tested to check whether it can interpret the ontology developed and automatically align it with the data extracted from the IFC file to produce results that consider design conditions, sustainability, and cost factors.

Chapter 7 concludes the work presented in previous chapters by outlining the main findings within the context of the research hypothesis. After that, the research limitation and future work are discussed. Finally, the research contributions are summarised.

Chapter 2. Literature review

This chapter presents a literature review in five sections. Section 2.1 elaborates on the BIM concept in terms of BIM maturity and the benefits and limitations of BIM in the AEC industry. Moreover, merging technologies with BIM can result in a robust decision-making framework and help in improving the data exchange process. However, most of these technologies are not fully embraced by the construction industry since the participants still lack knowledge about these technologies. Consequently, in Section 2.2, several technologies were investigated in relation to BIM. Based on this investigation, the semantic web was selected for this research. Section 2.3 discusses the concept of interoperability and the work that has been accomplished to deliver a high level of data exchange in the AEC industry. Finally, Section 2.4 discusses the semantic web and multi-objective knowledge base. However, due to the large-scale of BIM topics and the focus of this research, this literature is by no means an exhaustive review. However, it indicates the many developments taking place in this area and their limitations. At the end of this chapter, the main findings of the review, which are closely related to the research gaps, are given.

2.1. Building Information Modelling for collaboration

The AEC industry plays an essential component in the UK economic world. However, large investments in the construction sector can be challenging, especially since minor project changes can cost people involved significant time and effort. BIM has commenced the path for innovative technologies and concepts to merge within the AEC industry. It is in a constant expansion loop since the construction industry is changing its traditional business methods, with information now being exchanged digitally rather than in paper form. However, the transformation is currently slow and faces many barriers.

2.1.1. BIM maturity levels

BIM milestones are required in order to establish a fully collaborative working environment (*BIM Levels explained* | *NBS*, 2021). Consequently, BIM maturity, Figure

1, has been developed. It started from stage 0, where no collaboration exists, and a project was designed and constructed based on 2D drawings using CAD tools. However, BIM is defined as a “methodology to manage the essential building design and project data in digital format throughout the building's life-cycle” (Howard, 2006). This data involves the geometrical and no geometrical data throughout the entire project lifecycle starting from conceptual design and covering all other stages (UK BIM Alliance, 2016). Thus, the requirements for information to be exchanged among various users have requested the engagement of new concepts and tools to enhance the decision-making and collaboration in a project. Consequently, stage 1, which represents object-based modelling, was the beginning of BIM, where design software moved from 2D drawings to 3D drawings. The focus was on innovating the design to make it better and more efficient.

Following stage 1, BIM maturity level 2 (stage 2), which presented as model-based collaboration, showed the exchanging of federated BIM models using concepts related to a common file format such as IFC and common data environments (CDE). The UK Government made BIM level 2 mandatory in the united kingdom in 2016 with an objective to transform the data exchange and management process in the UK construction industry (UK BIM Alliance, 2016). The mandatory BIM level 2 has helped the UK in saving a massive amount of money (Modelling and Plan, 2015).

Finally, BIM level 3 (stage 3), network-based integration, requires entirely integrated BIM models hosted by a CDE. Furthermore, some argue that BIM is a tool, not a project delivery method. Consequently, the concept of integrated Project Delivery (IPD), which is a “project delivery approach that integrates people, systems, business structures and practices into a process that collaboratively harnesses the talents and insights of all participants to reduce waste and optimize efficiency through all phases of design, fabrication and construction” (AIA California Council, 2007), was proposed to work with BIM to pull the tool’s capabilities (AIA California Council, 2007).

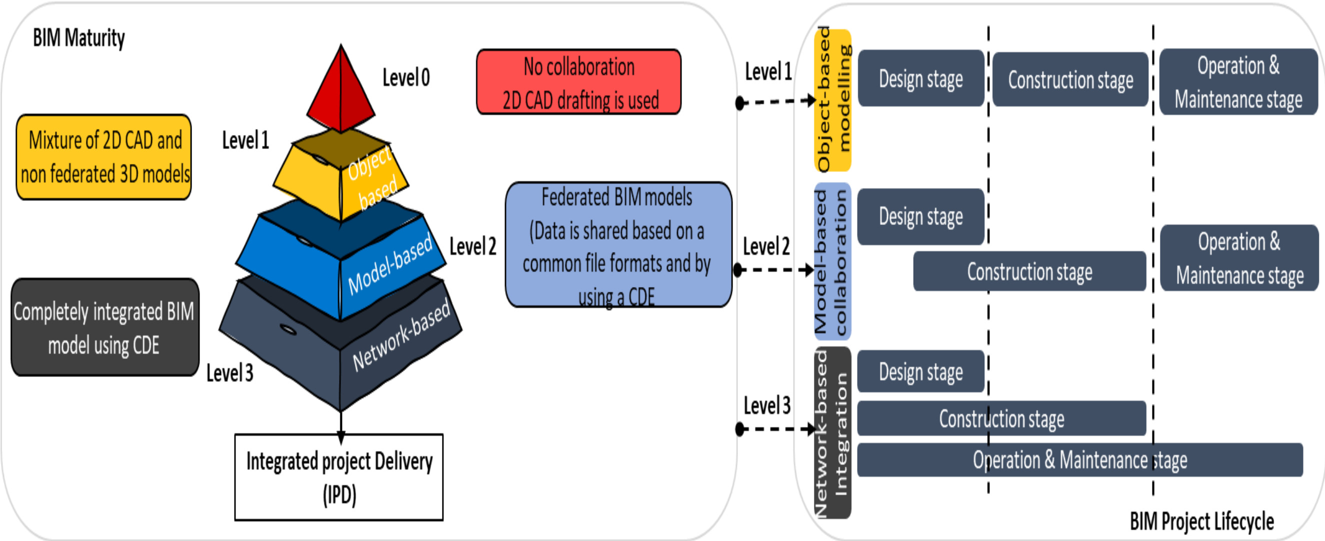


Figure 1: BIM maturity in relation to BIM project lifecycle

Furthermore, in order to control and achieve a higher level of BIM maturity, several UK standards have been released. For instance, the UK 1192 series have been released. The first workflow standards were the BS1192 series, Figure 2. It starts with BS1192:2007, which was the core standard for collaborative processes and is used to standardise information handling required for BIM Level 2 implementation (Bradley et al., 2016). In this standard, several considerations need to be implemented for BIM operation. For instance, roles and responsibilities of each design participant, naming conventions, using a CDE and others. Following this standard, BS 1192-4, which is a document that defines expectations for the exchange of information throughout the lifecycle of a facility, was released in 2014.

After that PAS1192 standards series were released, which form the main series of BIM standards along with the CIC BIM Protocol, the digital plan of works and the Uniclass classification system (Bradley et al., 2016). PAS1192 came with different parts starting from PAS1192-2 in 2013, which provides specific guidance for the information management requirements associated with projects delivered using BIM (Bradley et al., 2016), and reaching up to PAS1192-7 in 2017, which is used as a specification for defining, sharing, and maintaining structured digital construction product information. As shown, each standard delivers various practices within the project lifecycle, Figure 2. BS 1192:2007 is not clearly recommended as a standard for BIM in the UK. However, it is jointly co-dependent with other standards such as PAS 1192-2:2013. The difference between PAS1192-2 and BS1192:2007 standard is that PAS1192-2 describes information exchange with BIM. It launched new BIM methods for information management, such as Employer’s Information Requirements

(EIR), which is a document that shows the client requirements regarding project delivery and the data exchange format used. Whereas BS1192:2007 provides guidelines for delivering all the information throughout the project lifecycle. PAS1192 was the most widely used BIM workflow in the UK. However, a specific BIM workflow cannot be recommended to all organisations.

Furthermore, the ISO 19650 series uses the UK 1192 series as its basis. After publishing ISO 19650-1 and ISO 19650-2, the international working group is currently working on ISO 19650-3 and ISO 19650-5, which focus on the management of information during the operational phase of assets and on the adoption of a security-minded approach to the management of information relating to sensitive assets, respectively. The development of these two standards is based on PAS 1192-3 and PAS 1192-5.

It is seen that each level showed improvements as the BIM concept developed. However, same as any other concept, several challenges and barriers accompany its adoption, which encourages research efforts to find solutions to overcome those challenges and issues. Consequently, this BIM maturity requires the engagement of new technologies, concepts, and solutions to improve data exchange and decision-making process since the complexity of BIM has created a barrier to adopt it in the AEC industry.

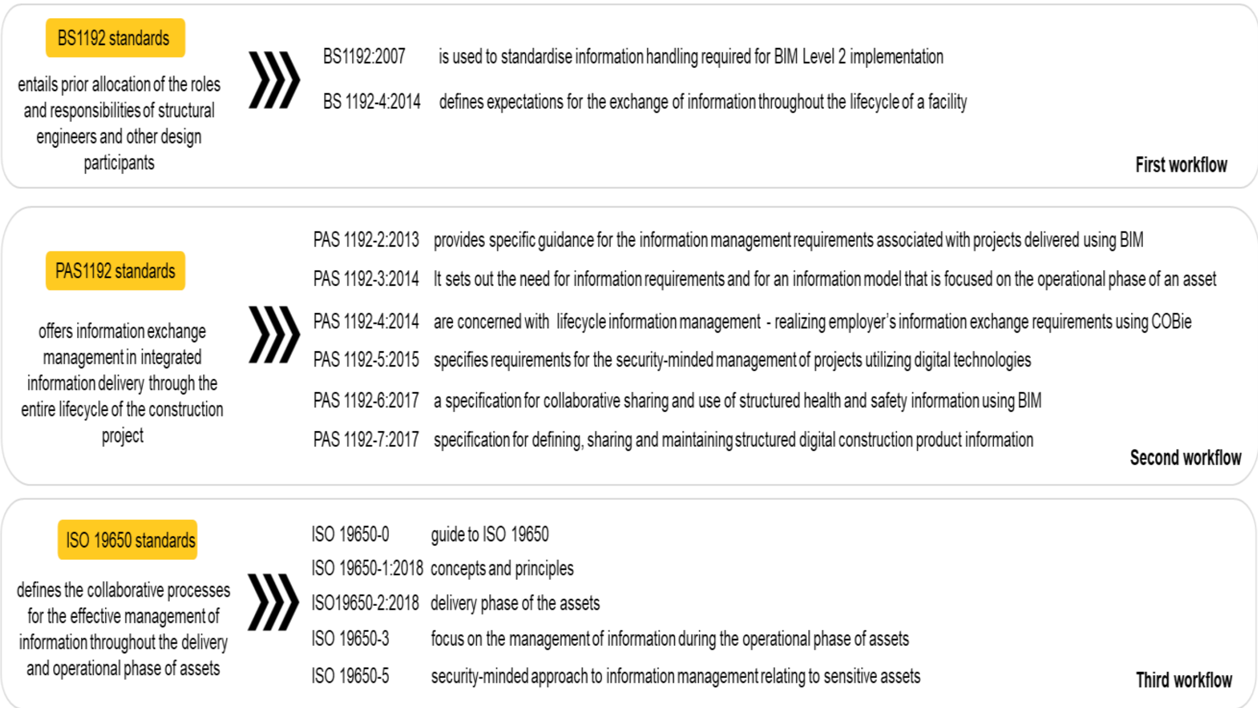


Figure 2: OpenBIM standards in the UK

2.1.2. Advantages and limitations of BIM in the AEC industry

BIM can be characterised as a repository for various information and knowledge, which can be essential for project success and valuable throughout a project's lifecycle (Steel, Drogemuller and Toth, 2012; Rokooei, 2015), especially because various stakeholders require different information to be exchanged in a project (Zhong *et al.*, 2018). BIM has brought several advantages to the AEC industry. For instance, it can help with enhancing collaboration on a project by bringing stakeholders closer together and supplying them with visualisation functionalities (Sun et al., 2017). Consequently, it helps synchronise the design and construction plans and detect design errors (Rokooei, 2015). If utilised in an appropriate manner, it can influence numerous aspects such as cost estimation, schedules, sustainability, compliance checking, design analysis, and thermal performance (Nguyen and Kim, 2011; Steel, Drogemuller and Toth, 2012; Chi, Wang and Jiao, 2015). However, there are still several challenges facing BIM in the AEC industry. For instance, interoperability and decision making have been acknowledged as important issues in BIM due to the existence of heterogeneous tools and systems (Grilo and Jardim-Goncalves, 2010; Sun *et al.*, 2017). Being unable to exchange data seamlessly limits collaboration in the AEC industry, which in return affects the decisions made within a project.

Moreover, the cost of BIM software tools is one of the factors limiting the application of BIM. Sun et al. (2017) also pointed out other factors such as BIM model ownership, model accessibility, data management issues, and data isolation, not forgetting to mention the security issues within BIM and the lack of a decision-making approach (Bhatija, Thomas and Dawood, 2017). There is a necessity to overcome BIM limitations since the usage of BIM has become mandatory in some countries. The blend of new technologies in the construction field can help resolve several limitations that restrict BIM adoption in the AEC industry. However, this can take several years to accomplish. There is still a lack of understanding of how these technologies are linked to BIM and how they can be leveraged toward future BIM innovations to enhance interoperability and the decision-making process.

2.2. Enhancing data exchange and decision making through cutting-edge technologies

Digital Built Britain describes BIM as “ a collaborative way of working, underpinned by the digital technologies which unlock more efficient methods of designing, delivering and maintaining physical built assets” (Modelling and Plan, 2015). Technology advancement in the construction industry has changed over the last decade, which helped in improving the data exchange and decision-making process within a project. Technologies such as artificial intelligence (AI), cloud computing (CC), ontology, blockchain (BC), data analytics (DA), Internet of Things (IoT), laser scanning (LS), and machine learning (ML) have brought tremendous benefits to the construction environment.

According to Digital Built Britain, which represents the next stage of the United Kingdom digital construction revolution, BIM has excellent potential to be combined with the IoT, semantic web and DA, which can result in better infrastructure and improve the utilisation of the facilities (Modelling and Plan, 2015). Not forgetting to mention the impact of ontology in improving the decision-making process within a project, which helped in converting domain information into knowledge. LS also plays a role in closing the gap between as-is BIM and as-built BIM, which helps with linking up late lifecycle stages to the design stage. Furthermore, technologies such as ontology and CC have shown their potential to overcome semantic issues (Venugopal, Charles M Eastman and Teizer, 2015) and improve collaboration within BIM (Jiao *et al.*, 2013), respectively. However, using the Internet as a platform for exchanging data among several team members can raise a major issue, which is security. Security has been identified as an important topic within BIM (Bhatija, Thomas and Dawood, 2017). In recent years, BC has been introduced to the research community to overcome the security issues with BIM. However, most of the research that was conducted on this topic was either conceptual (Turk and Klinc, 2017), a survey (Nawari O Nawari and Ravindran, 2019), or a literature review (Li, Greenwood and Kassem, 2019). Hence, it is currently a hot research topic. Therefore, this section aims to provide a review of publications to identify the association between the above technologies and BIM and how they can enhance data exchange and decision-making in the AEC industry.

Exploring field knowledge is an excellent way to discover gaps and highlight the most vital research areas. To evaluate the advancements and research areas in the construction industry, Oraee et al. (2018) recommended bibliometric analysis for targeting specific areas of the construction industry. Bibliometric analysis is a document analysis method that is applied to determine the topics related to a field based on the profiles, relationships, and clusters in the research (Zou, Yue and Vu, 2018). Therefore, in order to address this aim, a bibliometric analysis focusing on the co-occurrence of keywords related to various technologies, their links with BIM, and their related research themes was presented based on WoS database papers from 2010 to 2019.

Zhao (2017) carried out a bibliometric analysis on BIM research wherein he acknowledged that BIM study has predominantly concentrated on categories of engineering, civil engineering, architecture, and construction and building technology along with current emerging categories such as management and sustainability. Moreover, Zhao (2017) identified the hot topics of BIM research, e.g., CC, LS, and Ontology. Furthermore, Santos et al. (2017) emphasised the innovative expanding in the BIM research field and found that topics related to BIM tools, BIM adoption, energy simulation, interoperability, and ontology are the standout subjects in BIM research. Although some of the above papers have discussed technologies such as Ontology, CC, and LS, these technologies were not the main focus of the papers.

The research can cover a wide range of topics and involves many technologies. However, the literature in this section is not meant to be exclusive. To restrict the scope of this section, eight research topics have been chosen: AI, CC, Ontology, BC, DA, IoT, LS, and ML. These technologies can potentially help with the usage of BIM during the whole lifecycle of a building, which in turn can assist with taking a further step to improve data exchange and decisions made within a project. This section methodology, as illustrated in Figure 3, comprises four main stages: paper retrieval (stage 1), which includes the initial total number of articles retrieved, followed by the removal of irrelevant publications, which includes two stages (2 and 3), where only specific types of publications and the relevant categories are selected, respectively. In stage 4, a bibliometric analysis is conducted. Further details of these stages are as followings:

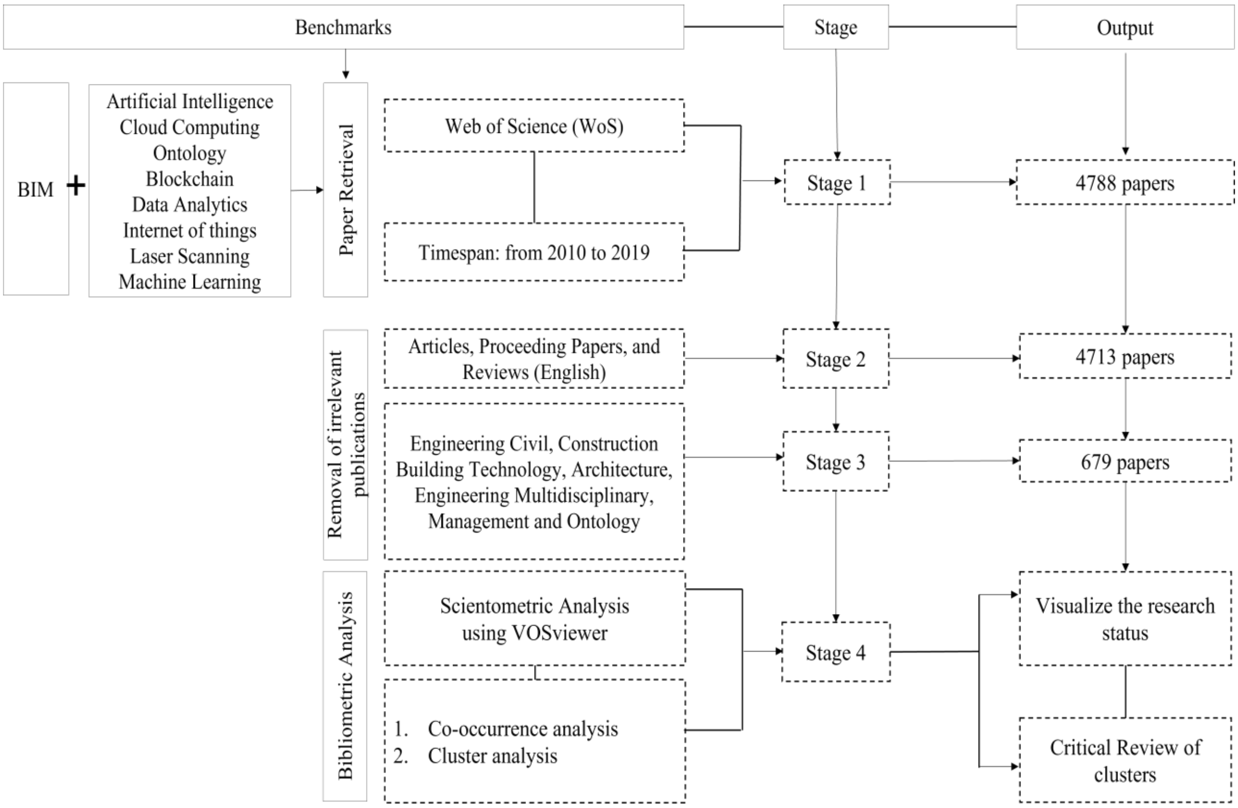


Figure 3: Research methodology used in this section

1. **Paper Retrieval.** The literature exploration was performed on the WoS database since it has more than 71 million records and over 10 million conference papers. The search was based on keywords using the OR and AND operators search benchmark. For instance, ((BIM OR building information modelling) AND ((artificial intelligence OR AI) OR (cloud computing) OR (ontology) OR (blockchain) OR (data analytics OR DA) OR (internet of things OR IoT) OR (laser scanning) OR (machine learning))). The literature involves an analysis of articles issued from 2010 to 2019 (ending on 2nd May). The result of the first stage was 4788 research publications.

2. **Removal of irrelevant publications (stage 2).** The aim of refining the search is to remove a large amount of irrelevant data that might not contribute to this study. The collected papers were based on the available articles, proceedings, and reviews since these sorts of documents can provide a comprehensive view of the existing research. Furthermore, only publications in English were collected since VOSviewer (VOSviewer - Visualizing scientific landscapes, 2021), which is a software tool established by the Centre for Science and Technology Studies at Leiden University that is used for the analysis of scientometric data, supports only English documents. A total of 4713 papers were identified.

3. **Removal of irrelevant publications (stage 3).** New groups were selected for this article, besides the ones identified by Zhao (2017), such as multidisciplinary engineering, management, and ontology. The final literature volume was 679 papers. In the WoS database, bibliographic data can be downloaded for at most 500 publications at a time. Thus, the documents were retrieved in two files. For each publication, the full record, including cited references, was obtained by using the “tab-delimited format” that is supported by VOSviewer.

4. **Bibliometric Analysis (stage 4).** Due to the enormous expansion in research, it is challenging to analyse papers manually. Hence, the VOSviewer was utilised as the analysis tool in this study, and a common quantitative and qualitative method was used to categorise and evaluate the literature. The VOSviewer supports two counting methods, the fractional counting method and the full counting method. The counting method used in this section is the fractional counting method since it is more convenient to avoid single terms that appear often in one document (Eck and Rousseau, 2014). The software supports distance-based maps and allows the user to choose the type of analysis. Five types of analysis exist in this software, co-authorship analysis, co-occurrence analysis, citation analysis, bibliographic coupling analysis, and co-citation analysis. Each of these can be used to deliver a specific need and focus. However, in this study, the focus is to identify the association between the existing technologies and BIM. Hence, co-occurrence analysis has been selected as the main focus of this study since it makes a major contribution to the aim of this research. Co-occurrence analysis, which is centred on the study of keywords, is used to analyse the word co-occurrence in at least two different articles (Li *et al.*, 2017). The connections between keywords are based on how many times keywords are used together in documents (Eck and Rousseau, 2014). Based on the keywords identified in the co-occurrence analysis, a cluster analysis was conducted to determine the research themes.

2.2.1. Co-occurrence analysis

The co-occurrence analysis is utilised to provide the main keywords of the collected articles and is based on three units of analysis. However, in this study, the “all keywords” unit of analysis is considered. The following settings have been applied in the VOSViewer tool: The minimum number of occurrences of a keyword was set to 12. The results indicated that 51 out of 2911 keywords met this threshold. Moreover, a manual normalisation was used to eliminate spelling errors and word repetition to ensure the consistency of the analysis. The final number of keywords was 38, as illustrated in Table 1 and Figure 4. Table 1 reviews the quantitative dimensions of the revealed keywords, which consist of the average publication year, the average citations, and the average normalised citation. The co-occurrence network of keywords generated from the final number of keywords is illustrated in Figure 4. The larger the number of abstracts and titles that have the same two terms in common, the closer these terms are to appear on the map (Eck and Rousseau, 2014). The tool divided the keywords into five clusters, as illustrated in Table 2, of which four out of five will be discussed later in Section 2.2.2.

Table 1: The quantitative dimensions of the discovered keywords.

keywords	occurrences	Percentage (%)	Total link strength	Avg. pub. year	Avg. citation	Avg. norm citation
BIM	137	11.77	120	2016	8	1.09
ontology	106	9.11	89	2015	9	0.92
management	65	5.58	62	2016	12	1.69
model	57	4.90	48	2016	14	1.44
system	56	4.82	54	2016	13	1.47
Construction	55	4.72	54	2015	25.5	1.98
design	42	3.61	39	2015	14	1.42
Laser scanning	37	3.18	35	2014	19	1.25
framework	36	3.09	37	2016	10	2.00
Knowledge management	18	1.55	15	2015	10	1.26
performance	33	2.83	27	2016	13	2.04
buildings	18	1.55	15	2015	13	1.46
safety	12	1.03	12	2016	12	1.84
recognition	15	1.28	14	2015	28	1.92
photogrammetry	14	1.20	8	2015	26	1.77
reconstruction	21	1.81	18	2016	25	1.93
visualization	21	1.81	18	2015	22	2.58
interoperability	21	1.81	19	2015	17	1.70
infrastructure	12	1.03	11	2015	16	1.25
prediction	13	1.12	11	2016	16	1.25
point clouds	22	1.89	22	2015	12	1.34
cloud computing	36	3.09	16	2014	11	1.14
classification	23	1.97	18	2016	10	1.02
simulation	27	2.32	24	2016	10	1.58
integration	15	1.28	14	2016	10	1.42
semantic web	21	1.81	20	2016	9.5	1.35
knowledge	21	1.81	19	2016	9	1.04
linked data	12	1.03	12	2016	8	1.38
algorithm	12	1.03	9	2015	7.6	0.92
machine learning	36	3.09	25	2017	8	1.34
information	36	3.09	32	2016	7	1.45
technology	18	1.55	16	2016	7	1.18
optimization	17	1.46	16	2016	5	0.76

architecture	13	1.12	10	2015	5	0.62
internet of things	20	1.72	15	2017	4	1.12
artificial intelligence	16	1.37	7	2014	3	0.20
internet	12	1.03	11	2017	3	1.50
big data	17	1.46	12	2017	2	1.12

The average publication year has been examined to detect the recentness of a keyword. For instance, keywords focusing on LS, CC, and AI were published around 2014, which suggests that the research community discovered the value and necessity of these technologies to boost the construction industry. Keywords related to interoperability, infrastructure, knowledge management, design, construction, and ontology with a significant appearance of keywords related to LS such as recognition, photogrammetry, and point clouds were circulated in 2015. In 2016, several themes emerged, such as management, performance, safety, reconstruction, prediction, semantic web, knowledge, and optimisation. Moreover, in more recent years, themes such as big data, IoTs, and ML have caught researchers' attention. The average publication year gives an idea of the expansion of mindsets in the research community. For instance, the research field has expanded from focusing on new buildings to focusing on new technologies to create BIM models for existing buildings and topics related to various infrastructures. Moreover, more attention has been given to issues of data exchange and how information can be converted into knowledge.

The following keywords have a high average citation: construction (25.5), laser scanning (19), recognition (28), photogrammetry (26), reconstruction (25), visualisation (22), interoperability (17), infrastructure (16), and prediction (16), which shows that there is a significant research focus on existing buildings, 3D rendering and visualisation, and the exchange of heterogeneous information between different stakeholders. In contrast, keywords such as big data (2), artificial intelligence (3), the Internet of things (4), optimisation (5), machine learning (8), linked data (8), simulation (10), cloud computing (11), safety (12), and ontology (9) received a lower average number of citations, which indicates that numerous technologies are yet not completely taken into consideration. For instance, technologies such as AI and ML can play an important role for big data, which represents large sets of structured and non-structured data, especially that many companies rely on big data analytics to discover the areas where they need to improve. Furthermore, the average number of normalised citations indicates which keyword has developed a higher yearly impact in the research area (Table 1). Scores below 1 indicate a low impact, whereas ratings above 1 indicate a higher impact. Because of the settings used in the VOS viewer tool, and the limited number of articles on topics such as BC, the results in Table 1 and Table 2 did not show this technology. However, since a combination of the qualitative and quantitative approaches is used, this topic will be discussed later in this section. Most of these technologies can help with extracting and managing the vast amount of data generated in a project. The construction sector needs to understand the importance of these technologies and how they can be leveraged since they can form the tools required to provide a better decision-making framework.

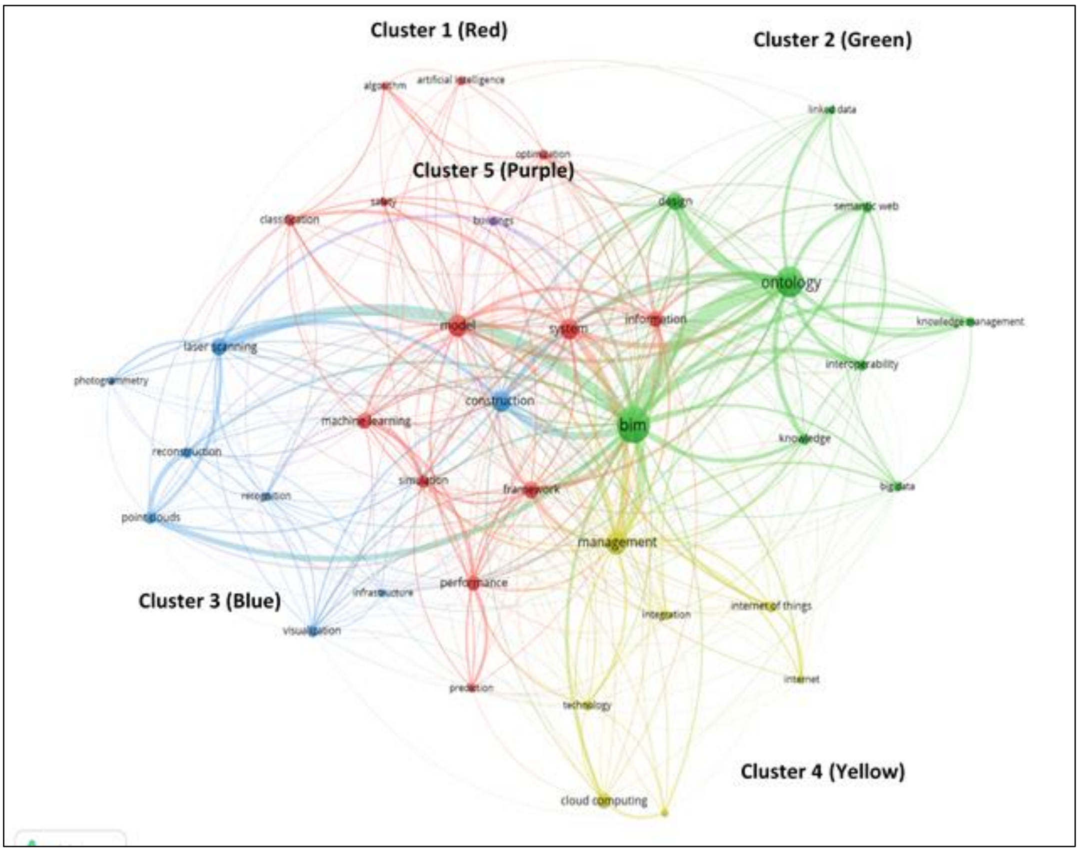


Figure 4: Network visualisation of keywords' co-occurrence.

2.2.2. Cluster analysis

Table 2: Clusters classification based on VOSViewer.

No	1	2	3	4	5
Items per cluster	13	9	8	7	1

Keywords	Algorithm Artificial intelligence Classification Framework Information Machine learning Model optimisation Performance Prediction Safety Simulation system	Big data BIM Design Interoperability Knowledge Knowledge management Linked data Ontology Semantic web	Construction Infrastructure Laser scanning Photogrammetry Point clouds Recognition Reconstruction visualisation	Architecture cloud computing Integration internet Internet of things Management Technology	buildings

A cluster represents a set of closely related nodes, and each node is assigned precisely to one group (Eck and Rousseau, 2014). The clusters, as shown in Figure 4 and Table 2, helped to break down the literature into separate groups that emphasise a particular aspect. Table 2 shows that the maximum number of keywords per cluster was 13, while the minimum was one. The keywords that are listed in the same group appear to show close similarities regarding the research topics. The significant gap between groups, as shown in Figure 4, is because of the lack of connections between some topics or due to the lack of relationships between them. A qualitative analysis was considered. However, it cannot cover the total collected literature because of the large volume. Thus, samples of the collected papers were reviewed to give insight into the current research (Figure 5). Figure 5 summarises the usage of technologies in the project lifecycle. BIM and other technologies have shown the coverage of many resolutions such as energy consumption, clash detection, code compliance, semantic interoperability, real-time monitoring, and other resolutions. In this section, Clusters 1–4 are discussed.

Cluster 1. In this cluster, the keywords identified were Artificial intelligence, optimisation, performance, prediction, safety, and Machine learning. The massive amount of data generated and the fragmentation of information as the BIM model develops have made the decision-making process complex. It has brought about the necessity of utilising technologies from computational areas to help with managing BIM models. BIM cannot be the definitive technological resolution for the construction industry (Shourangiz et al., 2011). Shourangiz et al. (2011) stated that BIM tools require AI to assess design alternatives. The use of AI and ML can be a solution to several concerns in the construction industry, such as performance, prediction, and safety so that better decision-making can be provided at earlier stages in a project.

For example, AI was used for occupancy prediction (Chen, Masood and Soh, 2016; Jiang et al., 2016; Ryu and Moon, 2016), predicting building energy consumption (Marasco and Kontokosta, 2016), evaluating the performance of sustainable buildings by predicting long-term weather patterns (Chakraborty, Elzarka and Bhatnagar, 2016), and forecasting construction costs and schedule (Wang, Yu and Chan, 2012). However, these studies showed no evident use of BIM. Hence, merging these technologies with BIM can result in a robust decision-making framework for the abovementioned areas of research.

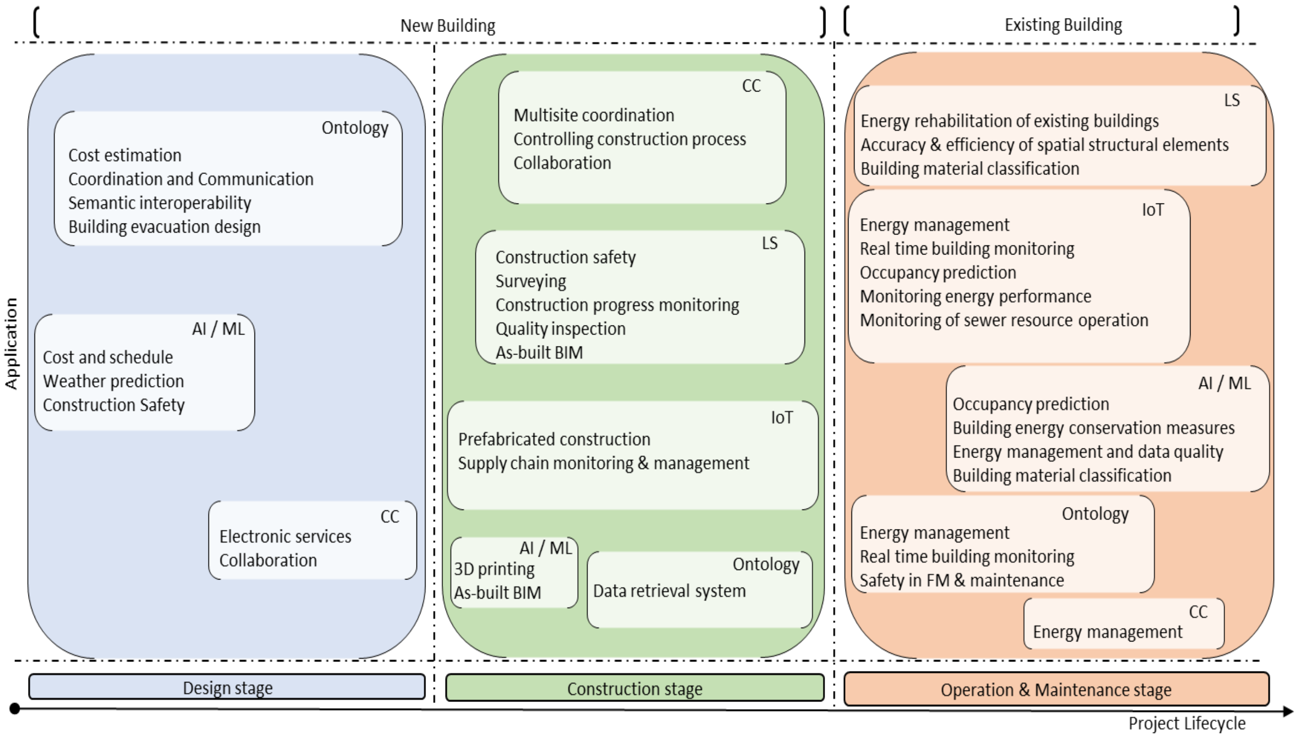


Figure 5: Existing research into technology immersion in the project lifecycle.

On the other hand, some papers have shown the usage of AI and ML in conjunction with BIM. For instance, information quality inside a facility can be improved by providing an online work order reporting approach (McArthur et al., 2018). Tixier et al. (2017) used machine learning to extract injury information from accident reports to predict construction safety. The method used by Tixier et al. (2017) can help with identifying risk injuries by detecting crashes in the early stages. Tan (2018) stated that merging 3D printing with BIM and AI can be a promising approach to solve 3D printing immaturity problems. McGlinn et al. (2017) used ML, ontology, and sensors to put forward a building energy management solution. The use of sensors and ML can help facility managers with the control of a building's energy consumption.

Moreover, ML was used to support 3D scanning to identify differences between 3D scans and 3D building models (Tamke et al., 2016). It seems that BIM necessitates the deployment of such technologies to enhance the decision-making process in a project.

Cluster 2. The main keywords found in this cluster were big data, semantic web, ontology, interoperability, and knowledge management. Several relevant reviews were identified on the semantic web and ontology. For instance, Sorensen and Christiansson (2010) have reviewed existing ontologies that can help to form a digital link between the virtual models and the physical models. They indicated that ontology could play an essential part in data sharing in the project lifecycle by including different stakeholders (Sørensen, Christiansson and Svidt, 2010). Pauwels et al. (2017) examined the expansion and application growth of semantic web technologies in AEC domains for several reasons, such as the necessity of enhancing interoperability problems. Data exchange necessitates the understanding of the industry procedures and also the information needed in these procedures (Venugopal *et al.*, 2012). The study by Pauwels et al. (2017) showed that the semantic web has the potential to contribute to applications that involve information from various disciplines.

Extensive research has been done in the area of ontology. For instance, studies have focused on semantic enhancement, information sharing, and online resource retrieval (König, Dirnbek and Stankovski, 2013; G. Gao *et al.*, 2015; Venugopal, Charles M. Eastman and Teizer, 2015b). These studies considered the Industry Foundation Classes (IFC), which is a BIM schema for data exchange. However, although it is a rich schema, Venugopal et al. (2015b) stated that it is not adequate for realising robust data exchange. They utilised ontology to resolve the doubt that exists in IFC (Venugopal, Charles M. Eastman and Teizer, 2015b). Furthermore, the study by Karan and Irizarry (2015) has shown that ontology is an alternative that can enhance interoperability among the geospatial and BIM domains. Costa and Madrazo (2015) applied ontology to establish a connection between BIM models and product catalogues based on IFC.

Ontology has been applied in the area of cost estimation (Lee, Kim and Yu, 2014; Liu, Lu and Al-Hussein, 2016; Abanda, Kamsu-Foguem and Tah, 2017; Niknam and Karshenas, 2017). The ontology developed by Lee et al. (2014) reduced human intervention during the cost estimation process. However, they mention the necessity of revising and updating the ontology with the possibility of engaging experienced engineers to improve the accuracy (Lee, Kim and Yu, 2014). Furthermore, ontology has been used for enhancing energy management (McGlinn *et al.*, 2017), building evacuation design (Boje and Li, 2018), improving coordination and communication between engineers by recognising conflicts in the BIM design process (Liu, 2013), and safety in facility management and maintenance (Wetzel and Thabet, 2015). Dibley et al. (2012) developed an ontology to support real-time building information monitoring, where they used ontology and sensors to manage the building data smartly. It shows that these two technologies can complement each other. Furthermore, Zhong et al. (2018) proposed an ontology framework to support environmental monitoring and compliance checking among different information systems. However, the above-mentioned ontologies were developed separately to serve a single objective decision. Despite the effort to develop a multi-objective decision-making knowledge base within the BIM context, most research did not provide a method that can work in parallel with a BIM model.

Another term that appeared in this cluster is big data. Lee (2017) stated that big data could help companies improve their business operations and services. It includes three main dimensions: (1) Volume, which represents the size of the available data; (2) variety that is linked to the heterogeneous data sources; (3) Velocity, representing the speed at which data is generated. This term was related to several technologies such as DA, AI, IoT and ML. For instance, DA is used to help the decision-making process. It can help to discover and extract important patterns and values from a massive volume of data. Furthermore, Lee (2017) mentioned that using this technology can save cost, improve quality, and help make better decisions.

Cluster 3. In this cluster, eight themes appeared, and the keywords identified were laser scanning, construction, reconstruction, and infrastructure. Looking at the review that was conducted on LS, Patraucean et al. (2015) provided a review of the as-built modelling process to explain the existing challenges of automatic as-built BIM generation. The focus of this review was on geometric modelling. However, Yuan et al. (2020) mentioned that non-geometric information of building elements, including building materials, is becoming an important part of as-is BIM. Furthermore, Wong et al. (2018) identified possible research directions concerning digital technology in facility management, where technologies such as 3D LS, point cloud, and IoT were considered. The main findings were the need to improve the interoperability of data from as-designed to as-built data and the necessity to enhance the accuracy of point cloud data (Wong, Ge and He, 2018). On a similar topic, considering the use of 3D point cloud data in the construction industry, Wang and Kim (2019) conducted a thorough review of the application of 3D point cloud data, in which they identified that

3D point cloud data could be useful for construction progress tracking, building performance analysis, construction safety management, and building renovation. They mentioned the importance of collecting semantic information through text mining techniques and integrating it with other sensors, which can be further linked to technology such as virtual reality (Wang and Kim, 2019). These studies reveal the possibility of merging technologies such as AI, ML, and ontology to aid in the use of LS and BIM.

Furthermore, the use of LS covers areas such as construction progress monitoring (Han, Degol and Golparvar-Fard, 2018), energy efficiency (Wang and Cho, 2015), energy rehabilitation of existing buildings (Lagüela et al., 2013), surveying (Mill, Alt and Liias, 2013), and quality inspection (Bosché and Guenet, 2014). Not forget to mention its application in construction management and facility management (Yuan, Guo and Wang, 2020). Yuan et al. (2020) stated the potential of such technology to enhance building material classification accuracy, noting that only a few studies have adopted this technology for building material classification. They also showed the importance of merging technology, such as ML with LS. In 2018, Liu et al. (2018) proposed a new approach using LS to improve the accuracy and efficiency of spatial structural elements; it can also be used for structural systems such as buildings, bridges, and culverts. Using LS and BIM can be an essential factor in achieving a complete project visualisation for new and existing buildings. However, Gao et al. (2015) noted the inconsistency between as-designed BIM and as-built BIM conditions since a building is not continually formed according to the design data indicated.

Cluster 4. Seven themes were shown, and the main keywords identified were the Internet of things and cloud computing. IoT has proven its usefulness in research concerning several construction tasks, e.g., prefabricated construction (Zhong *et al.*, 2017; Li *et al.*, 2018), behavioural modelling (Ciribini *et al.*, 2017; Pasini, 2018), supply chain monitoring and management (Dave *et al.*, 2016), monitoring energy performance (Wu *et al.*, 2015; Kang, Lin and Zhang, 2018), and real-time monitoring of sewer resource operation (Edmondson *et al.*, 2018). Moreover, Chen et al. (2018) established a warning system for fire rescue using IoT to restore the real-time conditions that can help in creating rescue plans.

On the other hand, several research publications on CC were identified. For instance, CC was utilised to support augmented reality on a construction site, which showed the possibility of providing a better collaboration among multidisciplinary users (Jiao *et al.*, 2013). Furthermore, it was used as a decision support tool for energy management (Cho *et al.*, 2015) and as an integration tool for electronic services (Jardim-Goncalves and Grilo, 2010; Grilo and Jardim-Goncalves, 2011). Fang et al. (2016) developed a system based on BIM and CC that showed great potential in applications like security control, monitoring tasks, and safety management at construction sites. Petri et al. (2017) highlighted the value that a federated cloud could provide to the construction industry. It can be the solution to project management and data sharing between different teams along the project lifecycle (Petri *et al.*, 2017). On a similar subject, a semi-structured interview on the use of cloud computing as an integrated platform for BIM applications mentioned the importance of having an integrated system over the web (Redmond *et al.*, 2012). However, the authors concluded that several challenges might arise with this system, such as security and a lack of acceptance by companies. Hence, technologies such as BC, which have been used to solve security issues, can back up such technology and help it be more adopted in the future. However, most of the research that was conducted on this technology was either conceptual (Turk and Klinc, 2017), a survey (Nawari O Nawari and Ravindran, 2019), or a literature review (Li, Greenwood and Kassem, 2019).

Few papers have been found regarding solving security issues within BIM. Turk and Klinc (2017) presented research on how and when BC can benefit the construction industry. The authors mentioned the need to develop a security model on top of the construction applications. However, this research was conceptual. Zheng et al. (2019) mentioned that less effort is made when It comes to information security. They proposed a novel approach where BIM and BC have been used to secure the information. Li et al. (2019) conducted a review on BC technology. They identified several challenges such as "lack of collaboration and information sharing; poor levels of trust between parties; low productivity; late payments; lack of enforcement of regulations; and issues surrounding ownership and intellectual property rights." The authors also stated that using this technology should also be accompanied by improvements across the legal, social and process dimensions. Furthermore, Dakhli et al. (2019) reviewed the possibility of using BC to save costs for a real estate developer.

2.3. Interoperability and data exchange

The massive increase in information compiled from various design tools has posed challenges in data exchange and complicated the decisions made within a project, which forced organizations to find ways to improve collaboration and decision-making between stakeholders. Bavafa (2015) has identified three aspects of information quality, Figure 6: Information accessibility, information accuracy and information interoperability. First, information accessibility is associated with information retrieval, based on building a knowledge base where information is formed and transformed into knowledge. Secondly, information accuracy is related to reducing inaccuracies in data sharing. Finally, information interoperability is the ability to exchange data seamlessly across various disciplines and stakeholders (Venugopal, Charles M. Eastman and Teizer, 2015a). However, interoperability is an issue that cannot be solved immediately. It more resembles a lifetime process that should be maintained and updated as new technologies and concepts will be available in the industry (Turk, 2020).

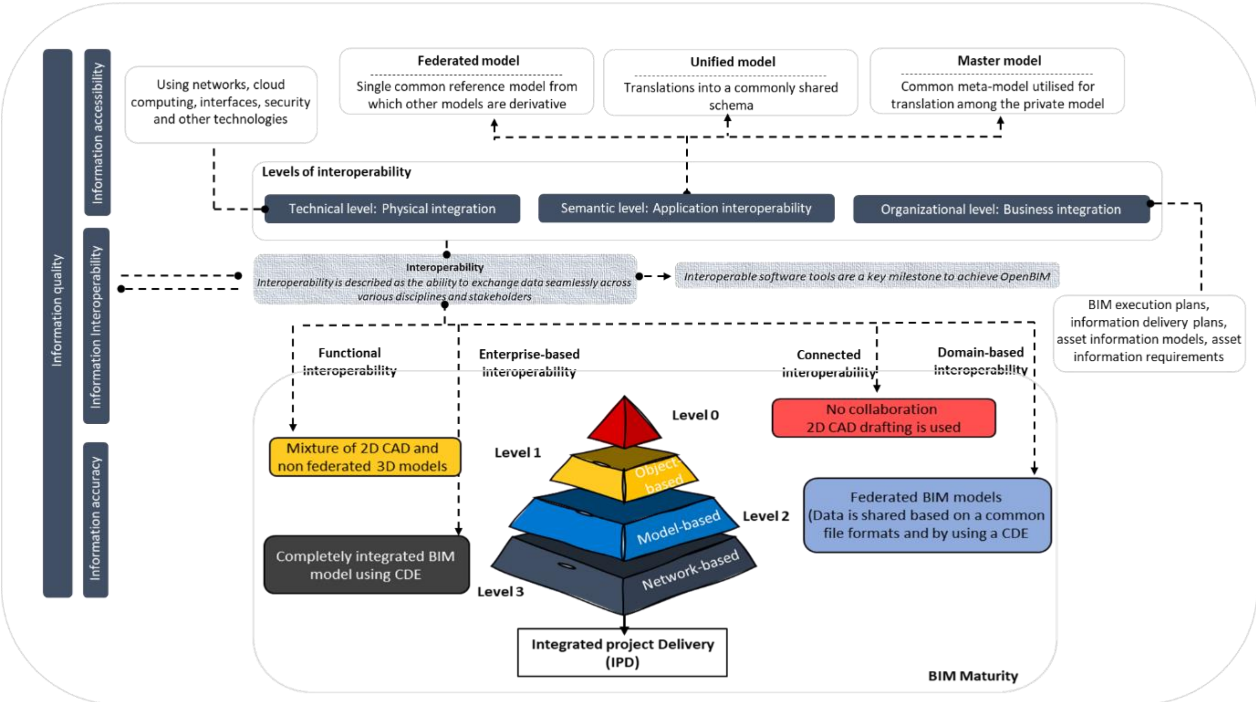


Figure 6: Interoperability in the AEC industry reproduced from Turk (2020)

Manso, Wachowicz and Bernabé (2009) defined the level of interoperability as a “set of criteria and associated processes for assessing information system capabilities and implementation in the context of the degree of interoperability required”. Paviot *et al.* (2011) mentioned three levels of interoperability that need to be considered to realise the full potential of interoperability, Figure 6. Firstly, the technical level, which involves providing communication between systems by adopting various technologies and tools. Secondly, the semantic level focuses on having a shared, common vocabulary that will be exchanged among the participants involved in a project without losing its meaning when various tools are involved. Turk (2020) stated that the major issue with semantic interoperability fixated on the openness of systems. Moreover, he identified various classifications of interoperability realisation: (1) Federated model, which is based on a single common reference model; (2) Unified model that is based on using open standards to exchange information such as IFC; (3) Master model using proprietary information model database. Thirdly, the organizational level focuses on procedures and guidelines for accessing and using data. It is shown that enhancing interoperability in an organization is concentrated not only on developing tools but also on solving issues related to openness of systems and business integration (Turk, 2020).

Furthermore, Steel, Drogemuller and Toth (2012) classify interoperability into four levels: The first level is limited to provide a successful file exchange among tools, while the second level goes further by focusing on parsing the exchanged file correctly. The third level concentrates on the visualisation aspects of the exchanged model among different tools. The fourth level is the most critical, where models must be semantically rich. The fourth level requires understanding the intention behind the exchange of models with the consideration of data consistency to avoid any data loss (Steel, Drogemuller and Toth, 2012).

The work that has been accomplished to deliver a high level of BIM data exchange in the AEC industry, which contributes to the semantic level of OpenBIM and interoperability, can be classified into two groups: (1) Defining and standardising information delivery, (2) Developing tools and platforms to back up the delivery of data across various stakeholders.

2.3.1. Defining and standardising information delivery

Around 1995, the Industry Alliance of Interoperability (IAI) was established. In 1997, it was given a new name, International Alliance for Interoperability, to demonstrate the advantages of data interoperability in assisting BIM. However, in 2005, it was renamed to BuildingSmart international (bSI), which is a global community of chapters, members, partners, and sponsors committed “to delivering improvement by the creation and adoption of open, international standards and solutions for infrastructure and buildings” (buildingSMART - The Home of BIM, 2021). BuildingSmart has three core programs they purse within the international organization (buildingSMART - The Home of BIM, 2021): (1) standards program focusing on standardizing, processes, and information exchanges; (2) compliance program, which is about certifying software and people against the developed standards; (3) user program that focuses on engaging and working with people in order to adapt and use all the developed standards within BuildingSmart. They have made up 20 chapters so far, which aim at identifying the need and requirements needed to improve the open standards (buildingSMART - The Home of BIM, 2021).

OpenBIM concept consists of several concepts and components that have been developed over the past years, Figure 7. For example, Industry foundation classes (IFC), information exchange methodology (IDM), which is used to “capture and specify processes and information flow during the lifecycle of a facility” (buildingSMART - The Home of BIM, 2021), and international framework for Dictionaries (IFD), which provides a flexible method of linking existing databases with construction information. Following the IFD standard, a reference library bSDD based on IFD, which is a library of object concepts and their properties that helps users find the right classifications, properties, and values by standardising all types of entities, properties, and classification, was established.

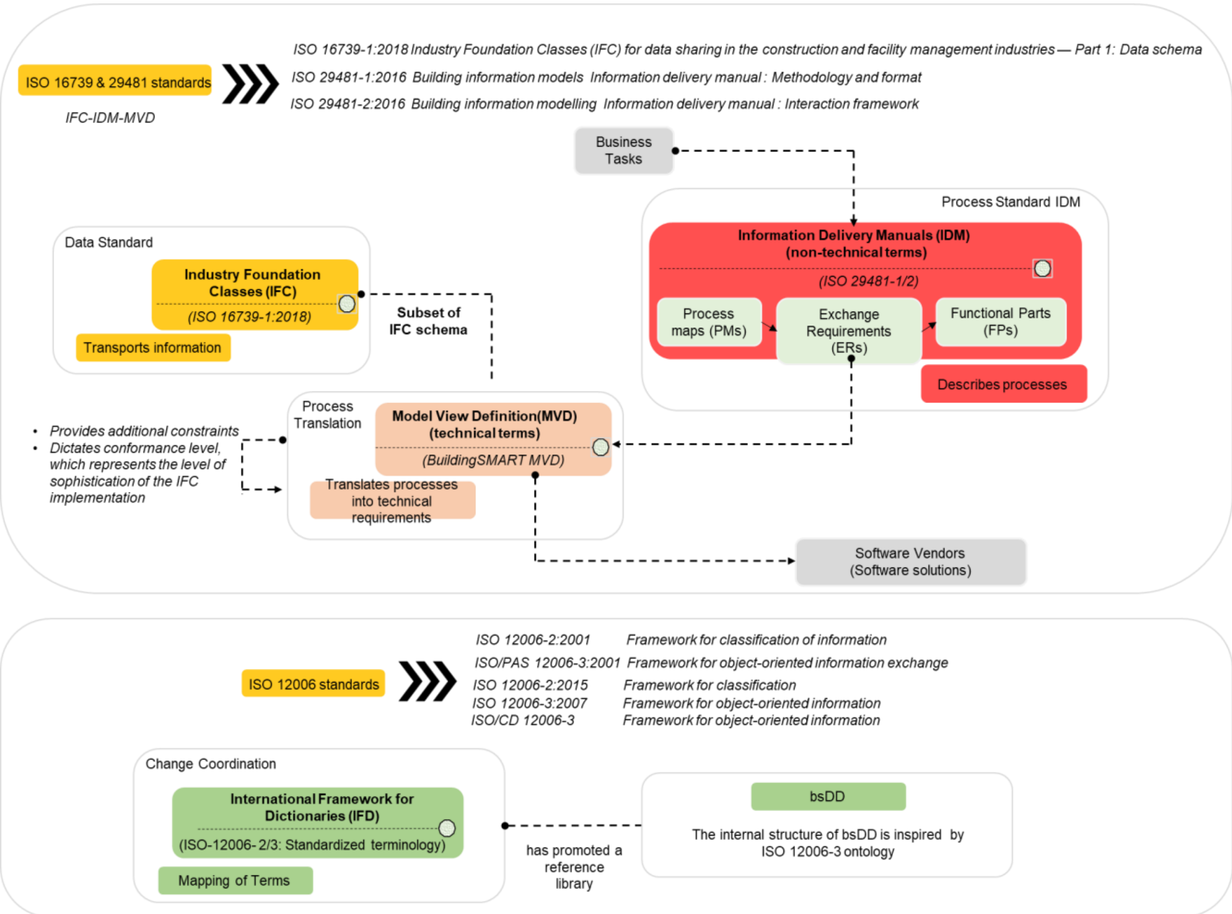


Figure 7: The relationship between IFC, IDM, MVD and other concepts

The first stable versions of IFC were published in the late 1990s. Starting from 2000, IFC2 was released, and various versions were released after that until the release of IFC2x3 TC1 in 2007, Table 3. After that version, a six-year period was the gap between the released version of IFC2x3 TC1 and IFC4, which indicates there was a major development in the schema. The latest versions show a vast expansion of the IFC schema. For instance, in the beginning, the schema was built to cover the building domain. In comparison, the latest versions have been expanded to include the infrastructure domain. However, this expansion makes the schema more complex and requires more effort to understand, especially by non-experts' audiences.

Table 3: Various versions of the IFC schema

IFC version	**Release year**	**Main enhancement**
IFC2	2000	Introducing the concept of a core model and domain extensions
IFC 2 Add1	2001	Fixing issues related to IFC2
IFC 2x2	2003	Included several extensions related to architecture domain, building control domain, construction management domain, facility management domain, HVAC domain structural domain
IFC 2x2 Add1	2004	Fixing issues related to IFC 2x2
IFC 2x3	2006	Improving the quality of the old versions of IFC 2x2
IFC 2x3 TC1	2007	Fixing issues related to IFC 2x3
IFC 4	2013	Enhancing the schema capability in its main architectural, building service and structural elements and enabling the extension to infrastructure. Moreover, enabling new BIM workflows, product libraries, BIM to GIS interoperability and thermal simulation
IFC 4 Add1	2015	Fixing issues related to IFC 4
IFC 5	-	Underdevelopment – focusing on infrastructures such as port and harbour.

Since the IFC schema covers various domains, it is not convenient to implement the entire schema in software vendors. It is important to define which IFC type is required to meet a particular user or domain requirement. To solve the concerns within the IFC schema, BuildingSMART proposed the IDM and the MVD concepts (Abualdenien, Pfuhl and Braun, 2019), Figure 7. They have been utilised to define and standardise information delivery. IDM composes of a project map (PM), exchange requirements (ERs), and functional parts (FPs) (Lee, Park and Ham, 2013). For instance, the PM helps define the overall and detailed workflow of tasks in a given discipline or among more than one. In this map, what information need to be created and exchanged can be defined. In contrast, the FP can help link this information to a schema by matching it to the correct entity in that schema to support software solutions, which form the initial steps that can help develop what is called MVD. However, the flexible nature of the IFC schema gives room to map the same information in different ways (Venugopal, Charles M. Eastman and Teizer, 2015a), which depends mainly on the developers, especially since there is no clear, logical connection between the units of information in the exchange requirements of an IDM and those of MVDs (Lee, Eastman and Solihin, 2016). Moreover, those concepts aimed to deliver a subset of the IFC schema to the end-user without inferring new statements. Consequently, using technology such as ontology can provide semantic enrichment of the IFC model by deducing new facts.

Furthermore, the number of MVDs is expanding since the construction industry is more eager to utilise BIM, where the information will be exchanged digitally rather than in document form. However, Lai, Zhou and Deng (2019) indicated that the current Industry is still a shortage of MVDs to deliver design data to the collaborative design stage. The development process of an IDM-MVD is complicated and time-consuming (Venugopal, Charles M. Eastman and Teizer, 2015a), which causes several challenges (Lee, Park and Ham, 2013) that constrain their embracement (Ramaji and Memari, 2018). For instance, due to the continuous rise in the number of ERs and FPs as the development process reaches completion, it becomes hard for the developers to re-use or track these ERs and FPs (Lee, Park and Ham, 2013), which leads to the duplication of time and effort spent on finding those ERs again by tracking them down or recreating new ones. Moreover, Gui et al. (2019) stated that MVD is designed for one-time data delivery and is valuable for the developers, but it is challenging to be understood by non-experts and end-users. Producing an MVD entails knowledge from the developers regarding the intricate structure of the IFC schema that can be hard to understand by non-experts (Won et al., 2013).

2.3.2. Developing tools and platforms for data exchange

Several research articles have focused on emerging tools and platforms. Bearing in mind that the first group (Section 2.3.1) can play a significant part in the development process of these tools by providing them with clearly defined input data. Deng and Chang (2006) developed a method to create a structural model from an architectural model based on the IFC schema. They pointed out that the differences in model representation and input format are the main reasons behind the lack of integration between design disciplines (Deng and Chang, 2006); thus, adopting a common format by software developers can be an excellent solution for interoperability issues. On a similar subject, Qin, Deng and Liu (2011) used IFC and extensible mark-up language (XML) technology, which is used to store and maintain data in a text file format, to build a framework to manage the information between architectural and structural disciplines. XML technology has an excellent structure to store and maintain information. They identify that the lack of a unified data exchange method limits the data exchange among diverse disciplines and results in integration issues (Qin, Deng and Liu, 2011). They stated that the process of automating the data exchange between IFC models comprises two aspects: the IFC parser, which is utilised to read the IFC physical file and the IFC model schema, which is used to create the equivalent objects defined in the IFC file in a machine comprehensible format.

Chen et al. (2005a) formed an IFC-based data web server for collaborative design between the architectural and structural domains. They stated that to augment the building design process, the IFC demands to be reinforced by additional tools or platforms. Moreover, a tool was developed by Liu, Li and Zhang (2010) to improve the data transfer from IFC-format architectural model to the PKPM structural analysis tool, which is one of the leading structure design software tools in China (Liu, Li and Zhang, 2010). Liu, Li and Zhang (2010) were able to enhance the collaboration between the architectural and structural domains. Furthermore, Wang, Yang and Zhang (2015) developed an IFC-based software tool for structural model conversion, which helped extract the required information by the structural domain to form the required structural model.

Hu et al. (2016) proposed a unified data model approach and developed a web-based platform based on IFC and several algorithms to solve interoperability issues between architectural, structural, and structural analysis models. They stated that there is a lack of techniques based on utilising a common data model, where all the data is standardised, which can play a substantial part in providing an improved data exchange. On a similar focus, Ramaji and Memari (2016) developed an approach to transform the architectural model into a structural analytical model using the architectural coordination view as a starting point for this conversion. According to Ramaji and Memari (2016), information exchange can be divided into direct data exchange, which does not require semantic modifications, and interpreted data exchange, which requires semantic enhancement.

There have been few studies that explore providing information to other downstream processes. Won et al. (2013) proposed an algorithm to extract a partial model from an IFC-based model without using the data structure in the IFC schema, where they used a pre-specified set of building elements (IDM) as an input. They mentioned that an extraction algorithm is semantically successful if it can preserve the same semantic relationships before extraction without any data loss. Furthermore, Zhang et al. (2013) used web ontology language (OWL) to develop an algorithm to extract a partial BIM model. They mentioned that processing the IFC file against the IFC ontology is the most crucial step in the developed algorithm.

On a similar topic, Nepal et al. (2013) used ontology-based feature modelling to extract construction-specific data from a BIM model. Gui et al. (2019) developed a method to extract domain-specific information to remove unrelated IFC information. MVD was used to provide the algorithm with the required data. They stated that although several collaboration platforms have been developed with a central BIM database, the model becomes hard to manage as the model size increases and results in inefficiency in data sharing. Moreover, Lai, Zhou and Deng (2019) developed an algorithm to transfer structural design data for collaborative design, where they also proposed an ER Matrix based on XML. It is seen that although authors mentioned the applicability of such methods for other domains, most of the research emphasis on architectural and structural models.

Furthermore, BIM realisation can be achieved either by a single data model or a series of closely linked federated models (Beach *et al.*, 2017). However, Preidel et al. (2018) indicated that direct utilisation of a single shared model is not recommended since it results in a complicated large model that can be hard to handle. On a similar subject, a collaboration between project stakeholders can be categorised into two main components: file-based collaboration or model-based collaboration. The file-based exchange of BIM information caused several construction industry issues, such as data transfer inefficiency, lack of interoperability, and data inconsistency (Das, Cheng and Kumar, 2015). Whereas Munkley, Kassem and Dawood (2014) indicated that using BIM servers and cloud computing are the main approaches utilised to facilitate model-based collaboration.

Several developments related to BIM servers have been developed to enhance collaboration in the AEC industry. Beach et al. (2018) indicated that these servers could be classified into two groups. First, centralised data repositories include Graphisoft BIM Server, Graphisoft BIM Cloud, Autodesk BIM 360, BIM 360 Glue, Forge, Onuma system, and 3DRepo. Secondly, distributed data repositories (decentralised), where data is stored across multiple servers, such as Autodesk Revit server and Bentley ProjectWise. Although certain developed tools such as Autodesk Revit Server and Graphisoft BIM Server deliver numerous essential features to enhance collaboration, they were developed initially to work with tools produced by vendor-specific such as Autodesk and Graphisoft (Schapke et al., 2018). Hence, they cannot work efficiently with tools from other vendors. Besides, they do not offer functionality for acquiring non-building-related information (Das, Cheng and Kumar, 2015). Das, Cheng and Kumar (2015) stated that BIM files generated using a vendor-specific tool could only be divided into sub-models if an organisation developed an API. Consequently, open BIM efforts such as IFC have been introduced to overcome such challenges, given that some of the mentioned tools implemented IFC in their tools. However, in most of them, IFC was not the essence of the tool. Moreover, other platforms also showed advantages, such as the BIM server, an open-source web-based platform. However, Das, Cheng and Kumar (2015) mentioned that this platform does not facilitate dynamic splitting and merging of BIM models.

2.4. Semantic web and multi-objective knowledge base

The decision-making process is a challenging task that requires the development of a framework based on historical knowledge and expertise. It is based most of the time on the available information and engineering knowledge about the targeted issue. A survey conducted by Bhatija, Thomas and Dawood (2017) showed that many AEC industry stakeholders are still unaware of knowledge management ideas. However, the majority agreed on the importance of sharing and exchanging knowledge among project participants and pointed out how the integration can be shifted from just information exchange to knowledge exchange.

The study by Pauwels et al. (2017) showed that the semantic web has the potential to contribute to applications that involve information from various disciplines. Ontology is a new semantic technology that is widely adopted in various areas such as knowledge engineering, natural language processing, collaborative information systems and knowledge management. It helps engineers to translate the domain knowledge into a format that machines can understand. Chong Johnson and Chong Johnson Lim (2015) have classified the application of ontology into three groups, Figure 8: firstly, design information annotation, search, and retrieval. Secondly, design knowledge representation. Thirdly, design information federation and interoperability.

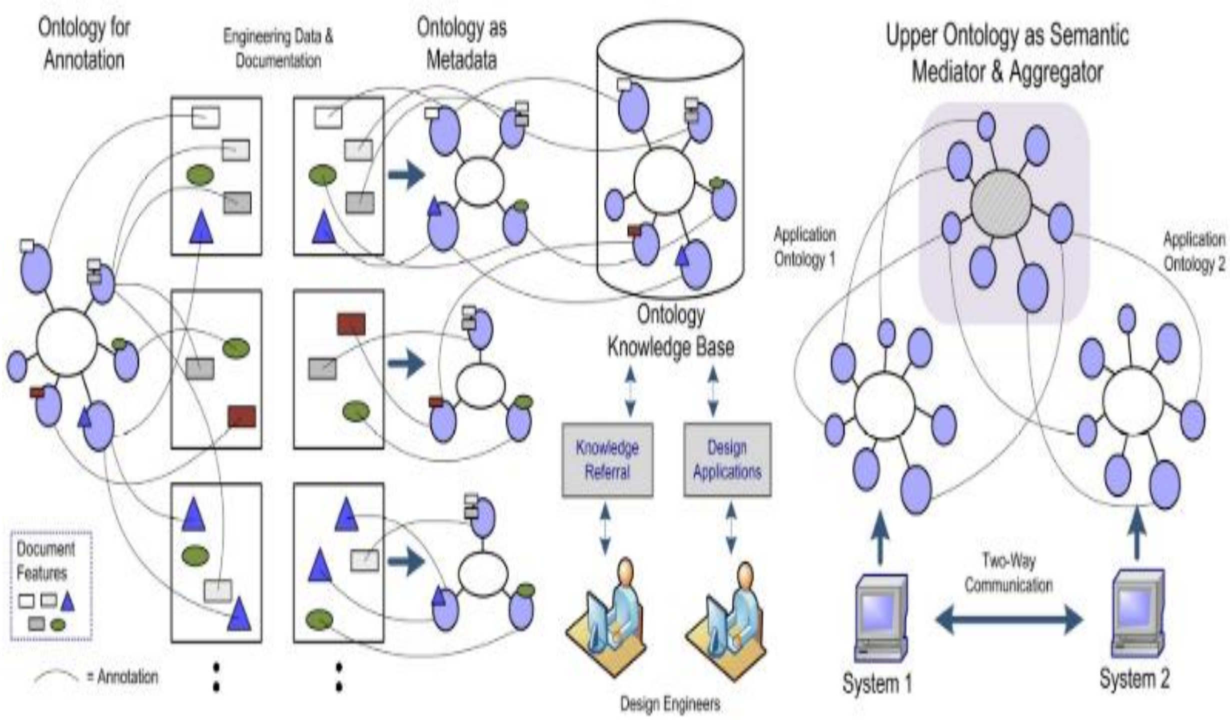


Figure 8: Ontology Applications in Design Engineering adopted from Chong Johnson and Chong Johnson Lim (2015)

The design stage requires decisions concerning conceptualisation, modelling, analysis, designing, detailing and cost estimation. These decisions help designers and stakeholders develop an ideal design with less effort and time, especially since some decisions rely mainly on other disciplines. Ontology has been used as a tool to enhance design information and provide knowledge (Chong Johnson and Chong Johnson Lim, 2015). For instance, it has been applied in the area of cost estimation (Lee, Kim and Yu, 2014; Liu, Lu and Al-Hussein, 2016; Abanda, Kamsu-Foguem and Tah, 2017; Niknam and Karshenas, 2017), energy management (McGlinn et al., 2017), and building evacuation design (Boje and Li, 2018). It has reduced human intervention during the design process. Therefore, utilising such technology can play an essential role in enhancing the IFC format performance and the decision-making process. However, the ontologies were developed separately to serve a single objective decision.

The decision making within the BIM context has mainly focused on individual factors such as economic factors, environmental factors, safety, and others. Currently, these factors are developed independently and require end-users to have prior knowledge about other domains. However, some engineers, such as structural engineers, may lack knowledge about some design areas, such as sustainable design, which can affect decision-making. Despite the effort to develop a multi-objective decision-making knowledge base within the BIM context, most of the research did not provide a method that can work in parallel with a BIM model, which can deliver choices based on the data collected from the IFC-based BIM model automatically. Consequently, Ontology has the capability to solve interoperability issues within BIM models by implementing domain knowledge into the BIM model, which can deliver semantic enhancement of the BIM model. Thus, it can help support BIM models with a holistic decision-making method that considers various perspectives and overcome the concerns about how BIM can handle various semantic information.

2.4.1. Underlying resources for holistic knowledge base development

2.4.1.1. Design perspective

The stability of a structure is one of the major concerns that need to be considered and verified while designing any building. The Design evaluation relies mainly on data collected from manuals, standards, regulations, and the designer's experience. This evaluation can be based on mathematical equations or data gathered from tables, graphs, or statements. Due to the limited time and resources for this research, it is difficult to explore all types of building structures. Thus, columns are selected since they are one of the most critical components in building design and play a strong part in the stability of a structure. For instance, according to Eurocode 2 standards (Eurocode 2, 2004), to make sure that the design of this type of element is feasible, the ultimate axial load capacity of the concrete column needs to be greater or equal to the axial load applied.

If the **ultimate axial load capacity** (Ned) is **equal to** or **greater than** the **axial load applied** → then the **selected column** provides enough **strength.**

To calculate the ultimate axial load capacity of the concrete column, the concrete load capacity and reinforcement load capacity need to be considered. Consequently, the ultimate axial load capacity will be the sum of both. This equation can be represented as follows:

$$\text{Ned (ultimate axial load capacity)} = 0.567 \times \text{Ac} \times \text{fck} + 0.87 \times \text{As} \times \text{fyk}$$

$$\text{Ac} = \text{Ag} - \text{As}$$

$$\text{Ag} = \text{Width} \times \text{Length}$$

$$\text{As} = [3.14 \times (\text{D}^2 / 4)] \times \text{no of bars}$$

Ac = Total net area of column cross-section

Ag = Total gross area of column

As = Total area of the longitudinal reinforcement

fck = Concrete characteristic strength

fyk = Reinforcing bar characteristic yield strength

On the other hand, some of the design conditions are not based on equations. It is represented in the form of statements, tables, or charts. This check is usually carried out manually, which can be time-consuming since engineers need to retrieve this information from design codes. Manual retrieval of this information can result in mistakes since it is most of the time based on human judgment and experience. Consequently, those conditions need to be converted into a machine-readable statement. For instance, to consider fire resistance requirements in the design of a concrete column, the minimum width of the column and the minimum concrete cover need to be taken into consideration. For example, according to Eurocode 2 standards (2004), Table 4, if a column and its cover have minimum width equal to or greater than 350 mm and 25mm, respectively, the standard fire resistance per minute is equal to 120. In other words, 2 hours of fire resistance requires 350 mm of minimum column size and 25 mm minimum cover. This condition can be represented as follows:

If the **Minimum** column **width AND Minimum** concrete **cover** are **equal to** or **greater than** 350 mm and 25 mm, respectively → then column **fire resistance** is equal to **120 minutes**

Table 4: Fire resistance time based on Eurocode 2 (2004).

Fire resistance per minute	Minimum column width (mm)	Minimum concrete cover (mm)
R60	200	25
R90	300	25
R120	350	25

Furthermore, environmental restrictions also play a major role in selecting the suitable material for a column. According to Eurocode 2 (2004), there are different exposures that can cause corrosion and damage to the concrete, such as carbonation-induced corrosion, chloride-induced corrosion, chloride-induced corrosion from seawater, freeze attack and chemical attack. For different exposure conditions, a minim strength of concrete needs to be utilised. For instance, in the case of carbonation-induced corrosion, the minimum concrete strength class that can be used is C25/30. In other words, C25/30 is the minimum strength that can be used if an element is exposed to carbonation-induced corrosion.

2.4.1.2. Sustainability perspective

Sustainability evaluation in building design consists of three main aspects: environmental, social, and economic. Those aspects cover various factors such as carbon emission, cost of the material used, resource consumption, construction safety of the workers, etc. However, the fragmentation of sustainability data and lack of knowledge require significant effort from the user to retrieve them since they are stored in databases and tables in various formats and locations. Moreover, current sustainability evaluation tools require a full design detailed model, which means most of the time, this evaluation will be accomplished in later design stages after completing the design. However, finding the best possible solution at the early stages of the design by looking into several factors together can play a significant role in the decision-making process, especially since having more than one design solution is possible most of the time. Thus, structural engineers can help in reducing the embodied carbon and cost of the project by assessing alternative construction materials compared to various building element sections.

The building sector in the United Kingdom utilises over 40% of the country's energy and emits about 330 million tonnes of carbon dioxide (Mockienė, Keras and Gilys, 2015). Research has discovered that the embodied carbon dioxide content in building materials, especially concrete has a major effect on the environment. Reducing carbon emission is linked with the type of materials used in a project. It can affect directly and indirectly other factors such as cost and design safety. In this research, two types of concrete material, which are regular concrete (NSC) and high-strength concrete (HPC), are considered to elaborate on the intended concept. The NSC covers the C25/30 and C35/45 concrete, while HPC covers C80/95 and C90/105. According to BS EN206-1 (2006), several rules related to the mixing ratio of NSC exist. Consequently, the mix proportions for the selected concrete were calculated following the available papers and data collected from standards. Moreover, HPC has a different mix ratio than NSC. Therefore, the HPC mix ratios suggested by Lim, Yoon and Kim (2004) and Larrard and Sedran (2002) are considered in this research.

The embodied CO2 for concrete is equal to the total embodied carbon contents of its components, which consist of embodied carbon during production, transportation, and construction. Since this research covers the early design stage, embodied carbon during construction is not considered and requires further investigation. Following the equation adapted from Zhang et al. (2018) and Yang, Song and Song (2013), the embodied CO2 for a concrete type can be calculated as follows:

$$\text{Embodied CO2 for concrete} = \mathrm{Embodied\ CO2(production)} + \mathrm{EmbodiedCO2(transport)} + \mathrm{EmbodiedCO2(construction)}$$

To calculate the embodied carbon content during production, several articles and databases were considered, such as Müller, Haist and Vogel (2014), Chen et al. (2010), in addition to the ICE database (2011) since it supports data for different concrete mix ratios, including water usage, unit weight, and embodied carbon content.

By using the formula adopted from Yang, Song and Song (2013), the embodied CO2 during production for C35 can be calculated as follows:

$$\text{Embodied CO2 (production)} = \sum_{i=1}^{n} \text{Wi x CO2i} = 300\text{x}0.93 + 1915\text{x}0.004 + 165\text{x}0.0003 = 286\ \text{kgCO2/m3}$$

For instance, knowing that embodied carbon content (CO2i) in the selected cement, aggregate, and water is 0.93 (kgCO2/kg), 0.004 (kgCO2/kg), and 0.0003 (kgCO2/kg), respectively, 300 kilogrammes of CEM I 32.5, 1915 kilogrammes of aggregate, and 165 kilogrammes of water comprise the unit cubic metre of C35 concrete, which equals to 286 kgCO2/m3 during production. Moreover, the average value of carbon dioxide emissions during material transportation was estimated to equal 20 kgCO2/m3. Therefore, the embodied CO2 in C35 is equal to 306 kgCO2/m 3. The final calculations of the selected concretes are shown in Table 5.

Table 5: CO2 and cost calculation of NSC and HPC concrete reproduced from Zhang et al. (2018)

Item	**$CO_{2\text{-}i}$ ($kgCO_2/kg$)**	**Cost ($US/kg)**	**W_i(kg/m^3)**			
			NSC		**HPC**	
			C25	C35	C80	C90
CEM I 32.5	0.930	0.22	240	300	-	-
CEM I 52.5	0.476	0.25	-	-	510	580
Aggregate	0.004	0.015	1955	1915	1673	1600
Water	0.0003	-	165	165	165	165
Steel	1.86	0.6	-	-	-	-
CO2 (production) (kgCO2/m 3)	-	-	249	286	264	297
CO2 (transport) (kgCO2/m 3)	-	-	20	20	20	20
CO2 (kgCO2/m 3)	-	-	269	306	284	317
Cost ($US/m 3)	-	-	97	95	242	260

Following the calculation of the embodied CO2 of NSC and HPC concrete in Table 5, the total embodied carbon content of an element can be calculated as follows:

$$\text{Total Embodied CO2e (Element)} = \text{Volume} \times \text{Embodied CO2e per unit}$$

Furthermore, the following equations have been considered:

The volume of a rectangular element

The volume of the element: $\text{Volume} = \text{Ac} \times \text{Height}$

Weight of the element

Weight of the element: $\text{Weight} = \text{Density} \times \text{Volume}$

2.4.1.3. Cost perspective

Cost estimation plays an essential role in keeping the project within the planned budget. It can help in creating the first glance on bills of quantities and bills of materials. Having an initial cost of an element earlier in the design stage can help select various design alternatives giving the stakeholders the chance to modify the structure. Despite that dataset for cost estimates is produced from the architectural model, the architectural model could provide only a few data such as space, element area, floor height, building parameter and gross area (Sherif, Jinkook and Chuck, 2011). However, what if this information or entity is not available in the exported IFC-based BIM model. How can the user get hold of those data to proceed with his decision?

According to Ramaji and Memari (2016), information exchange can be divided into direct data exchange, which does not require semantic modifications, and interpreted data exchange that requires semantic enhancement. Therefore, using an ontology, especially SWRL rules, can help deduce new information based on existing ones. Consequently, new facts can be deduced even if some information was not included in the exported file. For instance, the area of the element can be calculated based on the rules embedded, which will work as a substitute for IfcQuantityArea, which can be further used to calculate other information such as the volume and weight of an element.

The cost perspective can be divided into two aspects: (1) the cost of the concrete material used, which can vary based on the concrete strength. This cost will be calculated in this research based on the concrete material selected. (2) the cost of the labours required, which can be calculated based either on the quantities, including concreting work, reinforcement work and formwork, or task duration. Yaman and Taş (2007) divided a building cost estimate process into three main steps: Firstly, classifying a building into its functional elements such as footings, columns, beams, walls, slabs, and other elements. Secondly, measuring the total quantity of each functional element. Finally, calculate the total cost by multiplying the total quantity of each functional element with the unit cost of each functional element. Therefore, in order to calculate the cost of material for an element, the volume of the element needs to be multiplied by the cost per cubic meter of the material used and can be represented as:

Total Cost of material based on the price of the material used

$$\text{Total Cost of material } = \text{Volume } \times \text{ Cost per unit}$$

On the other hand, estimating the total labour cost during construction earlier in the design stage can improve the management side of the project, which significantly affects the budget of the project. However, labour cost is often defined by labour according to labour skills, which is different from the cost of material prices, which are provided by manuals or suppliers (Abanda *et al.*, 2011). Consequently, the need to develop a knowledge base that can model labour costing is of great importance, and it is necessary to consider this aspect in the early design stage.

In this research, the total labour cost of a column can be calculated by considering the labour cost of column concreting, the labour cost of column reinforcement and the labour cost of column formwork. First, the labour cost of column concreting includes two main factors: the total volume of concrete, which will be calculated based on the section's dimension, and the worker's pay rate. This can be represented as follows:

The labour cost of column concreting

$$\begin{aligned}&\text{The labour cost of column concreting (\$)}\\&= \text{Total volume of concrete in } (\mathrm{m}^3) \times \text{payrate of worker per } \mathrm{m}^3\end{aligned}$$

Secondly, the labour cost of reinforcement used in a column also needs to be determined. This can be calculated by considering the total weight of steel bars and the pay rate of a worker. Knowing that the weight of steel bars can be computed by multiplying the density of steel material (7850 kg/m^3) by the total volume of the longitudinal reinforcement. This can be calculated following this equation:

The labour cost of column reinforcement

$$\text{the labour cost of column reinforcement} \\ = \text{Total weight of steel bars in } kg \times \text{payrate of a worker per } kg$$

The volume of steel bars

$$\text{Volume of steel bars} = \text{As } \times \text{Height}$$

Weight of steel bars

$$\text{Total weight of steel bars} = \text{Density steel} \times \text{Volume of steel bars}$$

Finally, the labour cost of column formwork can be estimated by multiplying the total area of the column formwork used by the worker's pay rate. The total area of column formwork can be calculated by considering the width, length, and height of the column. This can be represented as:

The total area of column formwork

$$\text{The total area of shuttering work } (\text{m}^2) \\ = [\ 2 \times (\text{area of width side}) +\ 2 \times (\text{area of length side})]$$

$$= [\ 2 \times (\text{width side} \times \text{height}) +\ 2 \times (\ \text{length side} \times \text{height})]$$

The labour cost of column formwork

$$\text{The labour cost of column formwork} = \text{Total area of column formwork in } \text{m}^2 \times \\ \text{payrate of worker per } \text{m}^2$$

Consequently, the total labour cost of a column can be represented as follows:

The total labour cost of a column

$$\text{The total labour cost of a column } = \text{labour cost of column concreting } \times \text{ labour cost of column reinforcement} \times \text{labour cost of column formwork}$$

It is work mentioning that the total cost of a column will be the combination of the cost of the material used, which can vary depending on the type of concrete used and the estimated total labours cost. This can be represented as:

Total cost of a column

$$\text{The total cost of a column} = \text{Total Cost of material used} + \text{Total labour cost of a column}$$

2.5. Summary of literature findings

This chapter introduced an overview of the status quo of research in the field of BIM, interoperability, and semantic web. Following the literature analysis conducted in this chapter, several findings were identified:

Firstly, the flexible nature of the IFC schema gives room to map the same information in different ways, which depends mainly on the developers. This flexibility results in issues related to data mapping, especially since there is no clear, logical connection between the units of information in the exchange requirements of an IDM and those of MVDs. The development process of an IDM-MVD is complicated and time-consuming (Venugopal, Charles M. Eastman and Teizer, 2015a), which is causing several challenges (Lee, Park and Ham, 2013) and leads to a shortage of MVDs to deliver design data to the collaborative design stage. The lack of a unified data exchange method limits the data exchange among diverse disciplines and results in integration issues. Consequently, enhancing exchanging data requires an informed understanding of various disciplines and domains in order to provide a unified dataset.

Secondly, the MVD concept is a document rather than a subset that describes how the IFC model specification is applied to data exchange between different application types, which aims to deliver a subset of the IFC schema to the end-user without inferring new statements. It was not designed to deduce new information from a BIM model. Consequently, it demands to be reinforced by additional tools and methods to enhance its performance.

Thirdly, in order to have a high-performance building within the budget of a project, it requires the engagement of various aspects such as sustainability, cost analysis, energy performance and others. However, there is a lack of a holistic decision-making knowledge base that considers different sources of information together. Several studies showed that ontology has the potential to contribute to applications that involve information from various disciplines, which can improve collaboration and the decision-making process within BIM. However, most of the developed ontologies were developed separately to serve a single objective decision and require manual input to process the data in a BIM model. Despite the effort to develop a multi-objective knowledge base within the BIM context, most of the research did not provide a technique that can work in parallel with a BIM model, which can provide decisions based on the data collected from the IFC-based BIM model automatically. Consequently, using technologies such as ontology can help support BIM models with a multi-objective decision-making method that can further enhance interoperability, the IFC format and the decisions made within a project.

Chapter 3. Research methodology

Based on the previous review findings, this chapter will initially present the overall arching methodology through which this research was carried out to clarify the principles and methods utilised in this research. Following this, a more detailed method for each research question will be discussed in the relevant section.

To clarify the related methodology applied in this thesis, epistemology, which is part of the research 'onion' (Saunders, Lewis and Thornhill, 2015), was selected as the core research philosophy. Several philosophy packages exist. For instance, Positivism focuses on things that can be measured, quantified, and scientifically tested. It refers to a hypothesis that is further broken down into research questions. This hypothesis can be tested and validated through quantifiable data and can also be replicated in order to generate similar results. In contrast, realism does not recognise the scientific method as the final perfect solution. It states that new methods can be explored to solve a certain issue. Despite the difference between various approaches, interpretivism, which is more focused on the nature of human participation in social science, leans towards qualitative studies. However, the pragmatism method argues that a researcher would not use a single methodology to conclude the actual reality. It pointed out that mixed research should be combined to achieve the stated target. Since the research carried out in this thesis contains certain elements of both positivism and interpretivism, the pragmatist research philosophy has aligned with the research project.

Furthermore, based on the research Onion model, the research approach is divided into the deductive and inductive approaches. The deductive method seeks the answers to the questions from the very beginning. Whereas. The inductive method refers to the searching for patterns resulting from observations made based on the collected data. Based on the observation, the created theories will be tested using the research hypothesis proposed. In this thesis, an abductive approach, which combines both deductive and inductive approaches, is selected.

Finally, there are several research strategies such as Design Science Research (DSR), Action Research Methodology (AR), Survey and Case Study, etc. However, since this research is implemented in the information technology research domain and requires a mixed method to reach the objectives, the DSR (Johannesson and Perjons, 2014) was selected. DSR is usually utilised to categories of artefacts referring to engineering and computer science disciplines to solve a generic challenge experienced in practice (Johannesson and Perjons, 2014). This methodology allows different methods to reach the research target (Johannesson and Perjons, 2014), which allows more flexibility in developing and implementing the desired solution. Despite the similarities this methodology reveals with action research methodology (AR), which is a qualitative research approach that involves the direct engagement and collaboration with an organisation in order to diagnose issues and provide solutions to those problems (Rezgui, 2007), they are distinct in several aspects (Niehaves, Ortbach and Tavakoli, 2012). For instance, Livari and Venable (2009) indicated that DSR is utilised to solve purely technical problems. In comparison, AR Is utilised to solve socio-technical challenges. Furthermore, in DSR, participants can be assumed, which is not the case in AR, where involvement is a prerequisite (Niehaves, Ortbach and Tavakoli, 2012).

The DSR comprises several steps, Figure 9, which are Problem examination, Requirements definition, Design and development, Demonstration, Evaluation and Communication (Johannesson and Perjons, 2014). The overall arching methodology in this research is shown in Figure 10:

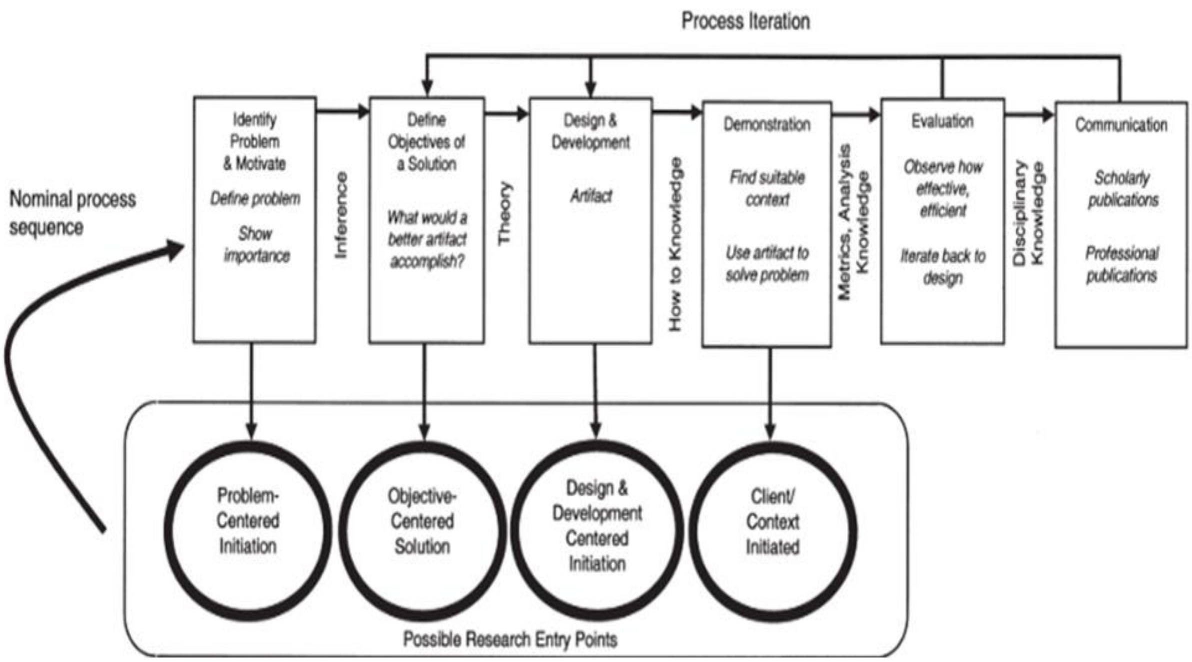


Figure 9: Design Science Research (DSR) methodology adopted from Peffers et al. (2007)

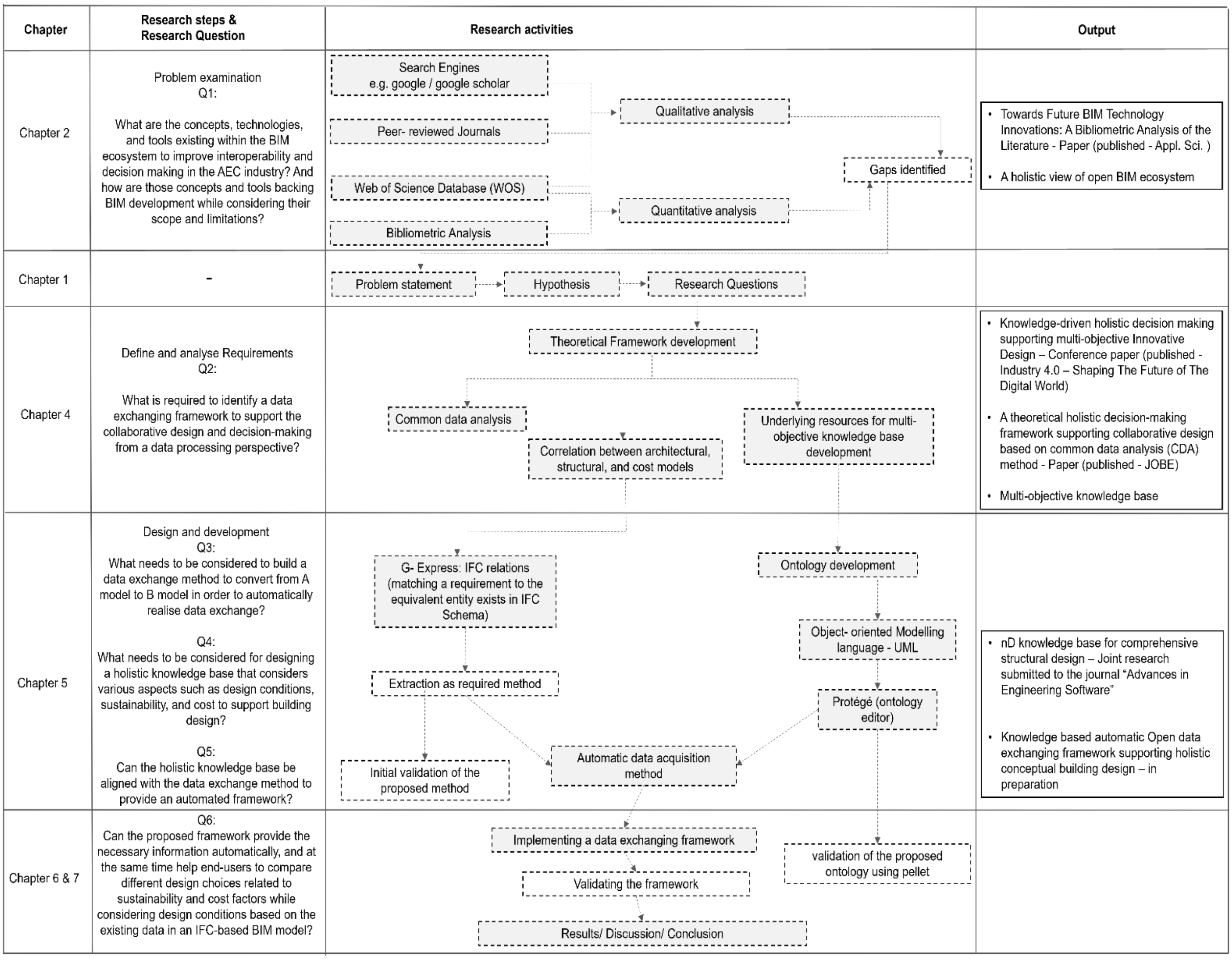


Figure 10: Overall arching Research Methodology

(Step 1) Problem examination. The problem, challenges, and related research were reviewed and investigated in Chapters 1 and 2, which informed that data exchange within the BIM context is still facing difficulties in reaching a high level of development. The AEC industry still lacks guidelines and methods that show what information is necessary for a specific task and what is the common information shared among different design models. Even though several attempts have been made to develop data exchange requirements for BIM models, there is still a lack of homogeneity, especially since the flexible nature of the IFC schema gives room to map the same information in different ways (Venugopal, Charles M. Eastman and Teizer, 2015a). A clearly defined "single truth of information" is still not acknowledged yet. Therefore, a better understanding of how different models can work together by identifying a "single truth of information" that can be shared among different domains and throughout the lifecycle could be valuable to deliver the foundation for a theoretical data exchanging framework from a data processing perspective.

Moreover, decision making within the BIM context has been mainly focused on individual aspects. Currently, these aspects are developed independently of each other, which results in a single objective decision, and require end-users to have prior knowledge about other disciplines, which can affect the decision-making process due to a lack of knowledge from stakeholders about other disciplines. Consequently, there is a lack of multi-objective knowledgebase for holistic decision-making within a BIM workflow that can work in parallel with a BIM model. Furthermore, IFC was not designed to deduce new information from a BIM model. It needs to be supported by additional technologies, methods, and formats to enhance its performance. Ontology has the potential to improve interoperability issues within BIM models by implementing domain knowledge into the BIM model, which can provide semantic enrichment of the BIM model. The contents of Chapter 2 responded to Q1, which is as follows:

Q1: What are the concepts, technologies, and tools existing within the BIM ecosystem to improve interoperability and decision making in the AEC industry? And how are those concepts and tools backing BIM development while considering their scope and limitations?

(Step 2) Define and analyse Requirements. To define the required information within the stated context, this step is divided into several sub-steps, which aim to answer Q2:

Q2: What is required to identify a data exchanging framework to support the collaborative design and decision-making from a data processing perspective?

A CDA referencing various concepts such as the IDM method, MVD and the concept of the semantic intersection was designed to understand for each profession what sort of data is required and what information needs to be exchanged in order to conclude the “single truth of information” and “partial truth of information” data sets that form the basis for a theoretical data exchanging framework to support the collaborative design and decision making. This method implemented in Chapter 4 is an extension of the literature mixed with a hands-on approach by examining several BIM models from a data perspective whilst considering the foreseen suggested resolution. The information defined in this sub-step is validated based on experts’ opinions, existing resources, and literature.

Moreover, the design requirements and data needed to develop a multi-objective knowledge base that considers various aspects are investigated in Chapter 2. This knowledge base can help stakeholders in making decisions associated with sustainability and cost while considering design conditions. Despite the effort to provide a multi-objective knowledge base, the proposed approach can enhance the decision-making process by providing the stakeholders with rational solutions with less time and effort since it eliminates the manual input aspect. The proposed knowledge base relied heavily on the available resources because of this research time limit. The findings from the “Define and analyse Requirements” step form the foundation for developing and implementing an automatic data exchanging framework.

(Step 3) Design and development. Following the analysis and the requirements identified in step 2, the technical developments of the proposed framework are given in Chapter 5, which aims to answer Q3, Q4 and Q5:

Q3: What needs to be considered to build a data exchange method to convert from A model to B model in order to automatically realise data exchange?

Q4: What needs to be considered for designing a holistic knowledge base that considers various aspects such as design conditions, sustainability, and cost to support building design?

The innovation of the proposed framework, Figure 11, lies in providing a data exchanging framework that combines both a data exchange method and semantic web technology to eliminate inefficiencies in data sharing and improve the decision-making in the early design stage by providing the stakeholders with rational solutions with less effort and time.

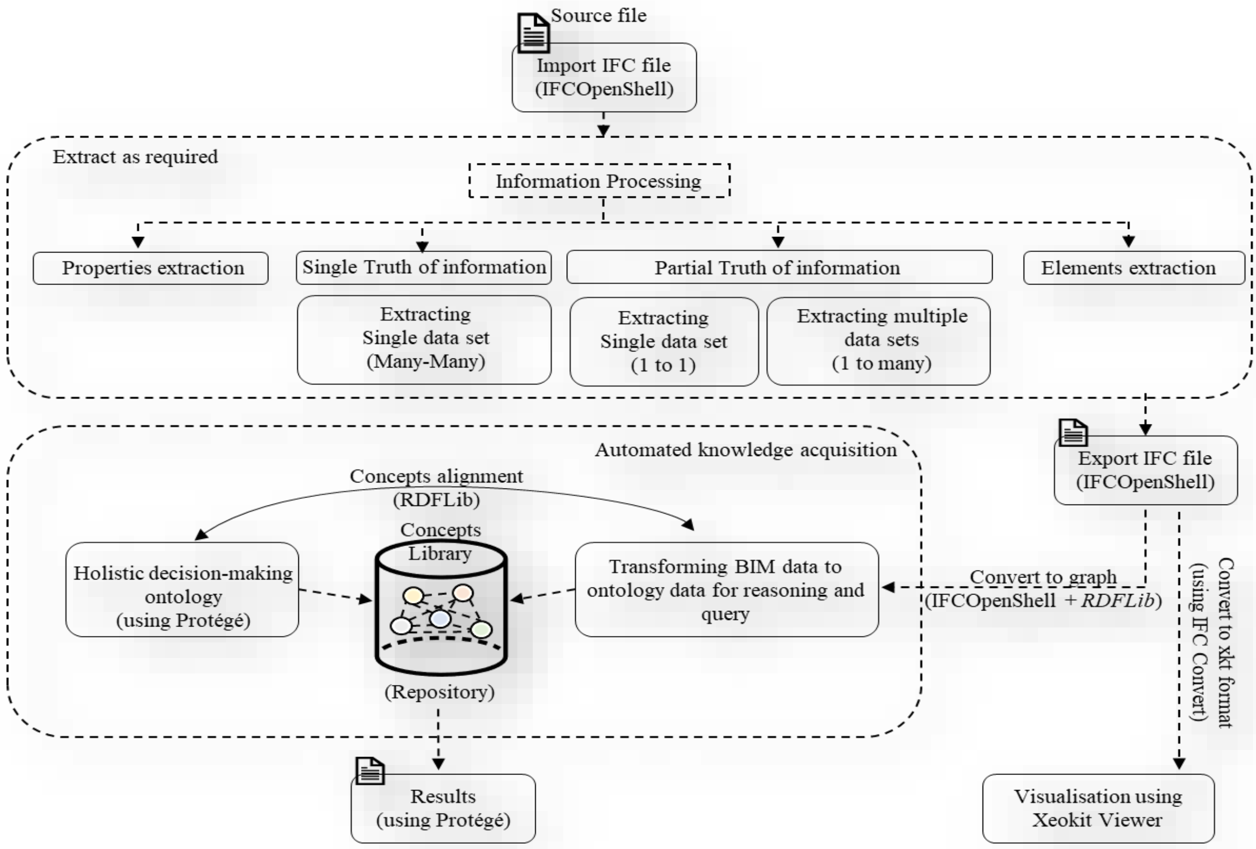


Figure 11: Knowledge-based automatic Open data exchanging framework

In this research, an object-oriented modelling notation approach based on G-express, which is a graphic modelling notation that is used for object-oriented information modelling (Ag, 2003), was utilised to map the defined requirements to the IFC data structure and to draw all the relationships among them. Following the mapping process, a data exchange method was developed to extract the necessary information from an IFC-based BIM model corresponding to the data sets defined. The method was implemented using a library such as IFCOpenShell, which is an open library source for python language. Further details about this library and other tools will be provided later (Section 5.1.2). After the data sets are extracted and saved in the appropriate IFC format, the framework will provide the end-users with two functionalities (Figure 11): (1) visualisation option via a web browser using Xeokit viewer, which is an open-source 3D graphics SDK built to view huge BIM models in the browser; (2) converting the exported IFC file into triples using a semantic web approach, python language and IfcOpenShell.

In the meantime, a multi-objective knowledge base is designed using the Object-oriented Modelling language UML. Protégé, which is a free, open-source ontology editor and framework for building intelligent systems, is used to design and edit the proposed ontology. The proposed knowledge base considers various types of data together, such as design conditions, sustainability and cost for collective decision making, in order to compare different design choices. Protégé has several plugins which will be used in this study, such as Semantic Web Rule (SWRL), which helps in deducing new facts based on the existing information; Query-Enhanced Web Rule Language (SQWRL), which helps the end-users to query the required information; and pellet, which will help in making sure that the developed ontology is syntactically valid.

One of the main contributions of this research is to align the multi-objective knowledge base with the data exchange method in order to extract the critical information from an IFC- based BIM model and merge them with the data existing in the proposed ontology in an effort to eliminate the human involvement by decreasing manual input. Consequently, the converted IFC file triples will be further aligned with the developed multi-objective knowledge base to produce design choices. Thus, the proposed framework can closely connect with real-world data by utilising the IFC-based BIM model. The proposed automatic data acquisition method was achieved using a matching classes name convention and a Uniform Resource Identifier (URI). This method was implemented using the Rdflib library, which is a Python library used for working with RDF files. All the developed tools and functions were tested and went through system validation. The data acquisition method is further discussed in Section 5.2.2 and aims to answer Q5:

Q5: Can the holistic knowledge base be aligned with the data exchange method to provide an automated framework?

(Step 4) Demonstration and evaluation. The intended framework, including its functionalities, was delivered to the software process to demonstrate its application. After conducting the technical developments in chapter 5, a use case scenario was set to demonstrate the intended framework. The developed framework consists of several functionalities such as data repository, data exchange method, visualisation, access rights administration, and multi-objective knowledge base. Bearing in mind that the developed platform is a proof of concept and still requires further enhancements. However, due to the limited time of this research, the ontology part is still not integrated with the proposed platform, and a separate procedure is carried out to elaborate the intended concept.

The application of a case study can assist in supplying additional assessments of the research developments. Hence, a scenario-based case testing was carried out on a real airport BIM model to validate the intended concept in this research. This step aims to answer Q6:

Q6: Can the proposed framework provide the necessary information automatically and at the same time helps end-users to compare different design choices related to sustainability and cost factors while considering design conditions based on the existing data in an IFC-based BIM model?

The testing and evaluation carried out in this step can be classified as follows:

- Testing and evaluating the data exchange method by applying it to the airport model to extract several data sets to check consistency and ensure no data loss is noted.
- Testing and evaluating the developed knowledge base, which is done: (1) through a reasoner plugin within protégé, which is Pellet, to make sure it is syntactically correct, and no inconsistencies exist. This testing was done to check whether the automatic data acquisition method can interpret the developed ontology and automatically align it with the data extracted from the IFC file without any inconsistencies. Consequently, this validation is done in two stages: before and after applying the data acquisition method to make sure the ontology is still semantically and syntactically correct. This validation plays a significant part in the knowledge development phase since it helps to make sure that the used terms/concepts are uniform and consistent throughout the ontology development stage. (2) Check the efficiency of the framework to provide a multi-objective knowledge base that considers single to multi-objective decisions and its ability to work with different sources of information that come from different standards and databases. The SWRL rules were tested by processing several queries in order to show the reliability of the proposed framework in providing multi-objective decisions and also providing new information that is not covered in the exported model.

Chapter 4. Theoretical framework development

This chapter provides the initial information requirements to provide a theoretical data exchanging framework from a data processing perspective in order to improve data exchange and decision-making in the AEC industry. To define the required information within the stated context, this chapter is divided into several sections intending to answer Q2.

4.1. Common data analysis (CDA)

A CDA was designed to understand for each profession what sort of data is required and what information needs to be exchanged to define a “single truth of information” and “partial truth of information” data set that form the base for a data exchanging framework to support the collaborative design and decision making. The research data were collected in two stages, Figure 12, as follows:

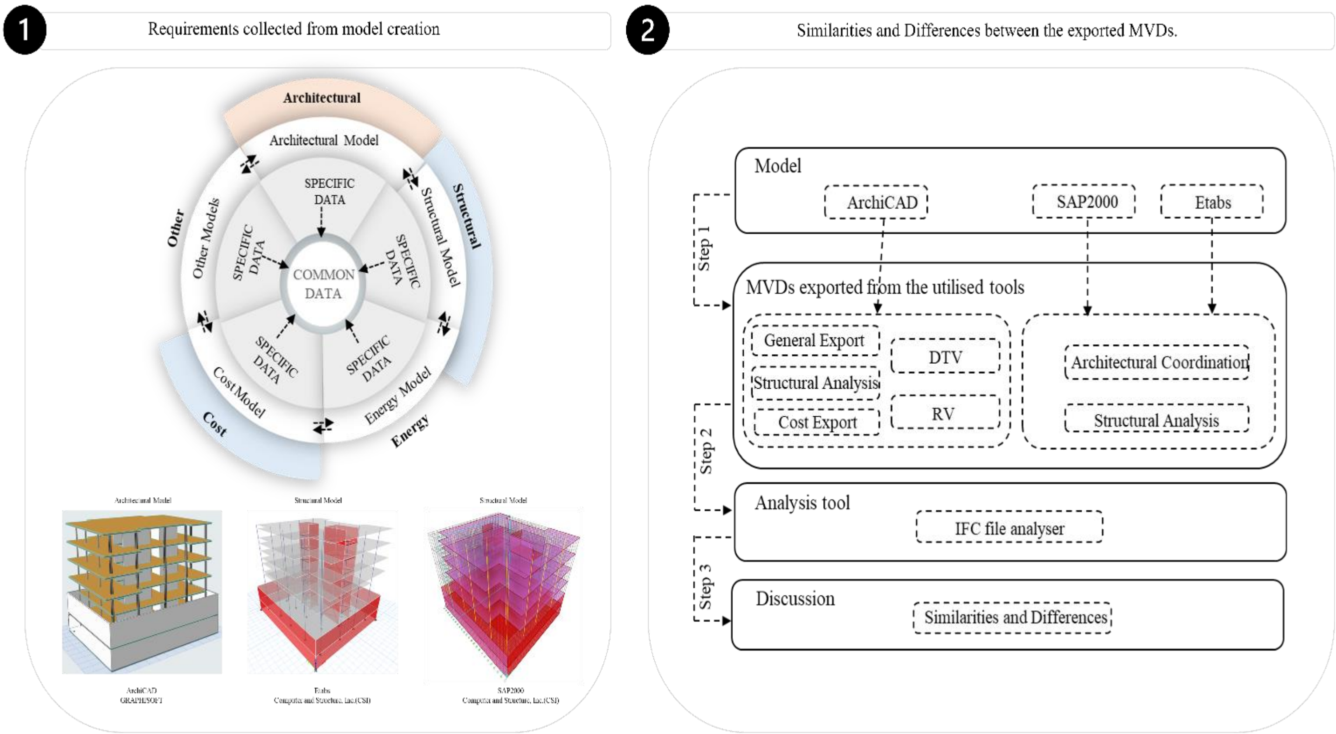


Figure 12: Common data analysis (CDA)

1) **Requirements collected from models**. The IDM-MVD method provides the essential structure for information delivery and exchange requirement. Hence, to analyse the information required, the IDM concept was considered in this

investigation, and an overall process map reproduced from RIBA was produced to provide the flow and sequence of tasks (Section 4.1.1). Moreover, three BIM models were chosen and investigated in this study based on experts' Points of View to understand what sort of information was required when these models were created and what information is required for exchange by different end-users. The architectural model is considered the source model, whereas the structural and cost models represent the delivered models to the end-users. To support the findings of this stage, previous work in this area was used to validate and approve those results.

2) **Similarities and Differences between the exported MVDs.** Several MVDs have been officially released, such as IFC2x3 Coordination View 2.0 (CV), IFC2x3 Structural Analysis View (SA), IFC4 ADD2 Design Transfer View (DTV 1.1) and IFC4 ADD2 Reference View (RV). These MVDs are generated independently and not interconnected. By using the models created in stage one, several MVDs were exported using the functionalities embedded within the selected software tools. Each of the exported IFC files was imported into IFC File Analyser for analysis, which is a tool developed by the National Institute of Standards and Technology (NIST) to generate spreadsheet files from an IFC file (NIST, 2011) to understand the IFC schema structure by identifying what IFC entities, relations and properties are used in the exported MVDs. Hence, this stage can support defining "single truth of information" and "partial truth of information". This stage will be further discussed later in this chapter (Section 4.1.2).

4.1.1. Requirements collected from models

In the design stage, the exchange of information can significantly affect the decision-making process, which can affect the downstream stages. The design stage necessitates specific inputs to create the required outputs. Architectural design and structural design are extensive and intricate processes. Changes at these two design stages are more common and have a higher overall impact on design than those at the downstream stages. A structural design process consists of several stages, starting with the conceptualisation moving to the modelling and analysis, then followed by designing, detailing, drafting and cost estimation (Ramaji, 2016), Figure 13. The spatial co-ordination stage (RIBA, 2020) involves the development of the primary structural model, such as selecting the structural system, building elements, material, cost and the location of the building elements. The information needed at this stage is mainly based on the information stated by architecture. Once the basic structural model is created, the structural engineer enriches the model with more specific information to create a structural analysis model. The architect and the engineers start to perform numerous analyses. For instance, they study the structural behaviour of the proposed system or conduct energy analysis of the building. At this stage, geometric data is not enough since more information is required to evaluate the model, such as loads, load combinations, materials, boundary conditions, etc. After the structural analysis model passes all the checks, it is time to design, detail and draft all the elements included in the project and ensure they meet the design code specifications. However, this stage is influenced by the engineer's personal experience.

Not forgetting to mention cost estimation, which plays an essential role in the decision-making process. It can help in creating the first glance on bills of quantities and bills of materials. Having an initial cost of the project earlier can help select various design alternatives, give stakeholders the chance to modify the structure, and spinoff the project towards its goals, which can help control the project's budget and save money during the construction stage. However, the cost estimation process is not done only at the start of the project; instead, it needs to be updated and modified as the project moves further from one stage to another, Figure 13. Choi, Kim, and Kim (2015) categorised the cost estimation for each stage as conceptual estimation in the planning phase, schematic estimation in the schematic design phase, and detailed estimation in the design development phase.

Moreover, Xu, Liu and Tang (2013) classified the data required for cost estimation into five components: product data, cost item data, quantity data, resource data, and price data. Consequently, fully automating the cost estimation process is difficult because of the dynamic resources needed and human intervention (2013). On a similar topic, some publications divided a building cost estimate process into three main steps: Firstly, classifying a building into its functional elements such as footings, columns, beams, walls, slabs, and other elements. Secondly, measuring the total quantity of each functional element. Finally, calculate the total cost by multiplying the total quantity of each functional element with the unit cost of each functional element (Yaman and Taş, 2007; Gokce and Gokce, 2013).

Despite these models having different roles, all those models have related tasks and require data to be exchanged seamlessly across them. Therefore, as Steel, Drogemuller and Toth (2012) stated, models need to be exchangeable on a semantic level to move further with BIM developments. Several software tools are available in the AEC industry, which provide users with many features and functionality. However, not all these tools support IFC file export functionality yet. Based on the BuildingSMART software implementation database (2021c) and due to their popularity, the following software tools were selected for this study: ArchiCAD (version 21) to represent the architectural model, SAP2000 (version 21.1.0) and Etabs (version 18.0.2) to represent the structural model. These software tools were developed by Graphisoft and Computers & Structures, Inc, respectively. However, this chapter is focused on the data requirements and not on the abilities of these software tools, knowing that each software is standalone software that is created to perform a sophisticated analysis.

In order to identify the required data, certain questions need to be answered. For instance, is the data required by the structural model, the architectural model or common among them? The same questions apply to any model that is used in other disciplines. Three models were created based on the selected software tools, as shown in Figure 14. The selection of two structural software tools instead of one was to investigate not only the unified information between architectural and structural models but also between the structural software tools themselves. The created models consist of the most common elements and related attributes in building construction, such as slabs, columns, and walls. Moreover, the structural models created using SAP2000 and Etabs consisted of a multi-story concrete building with uniformly distributed live and dead loads. Note that the ends of columns connected to the ground are assigned as fixed support. The data collected from all three models are shown in Tables 6 and 7. Furthermore, to support the findings in this section, previous work in this area was used to justify and endorse those results.

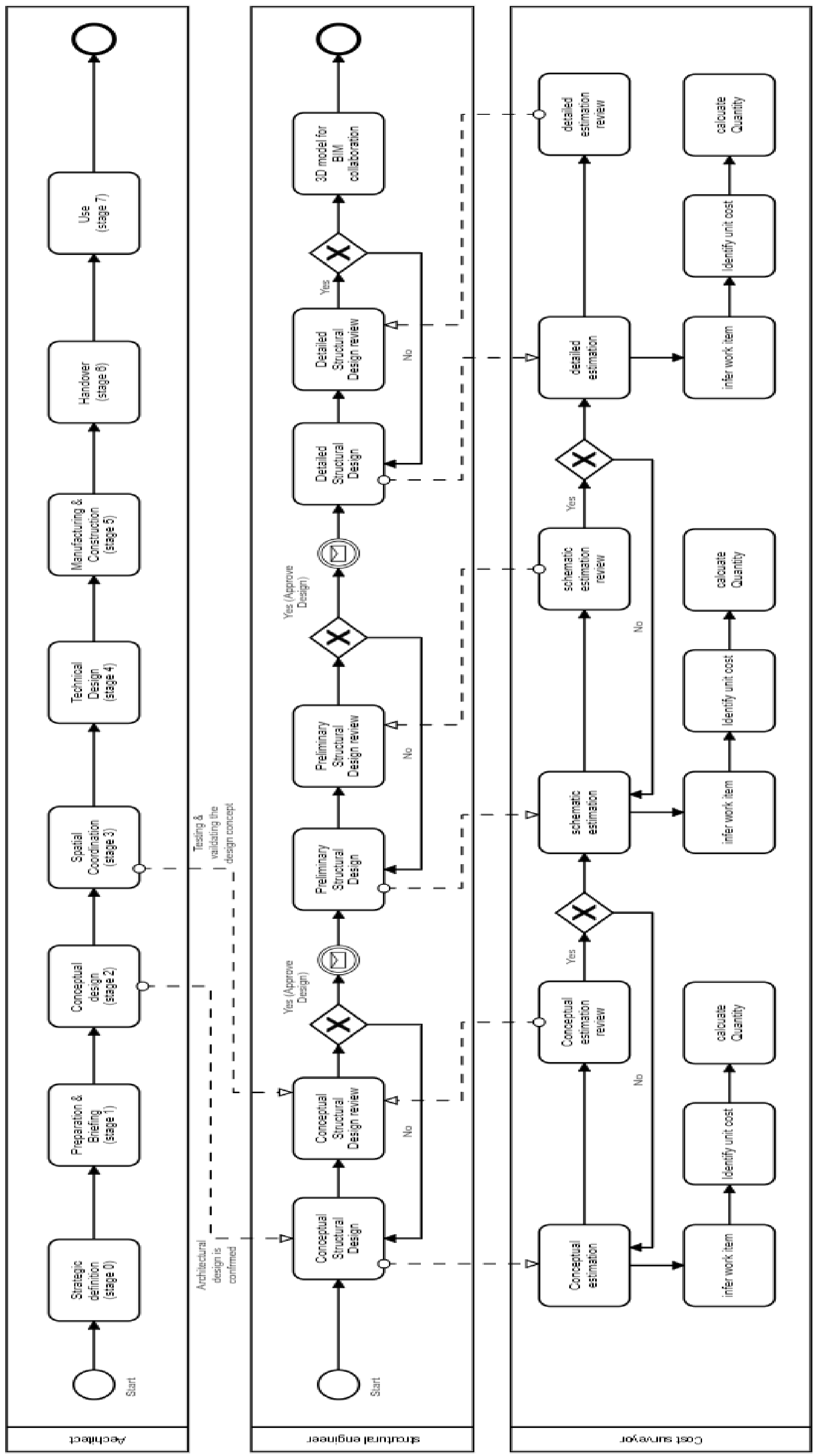


Figure 13: Overall process map reproduced from RIBA (2020)

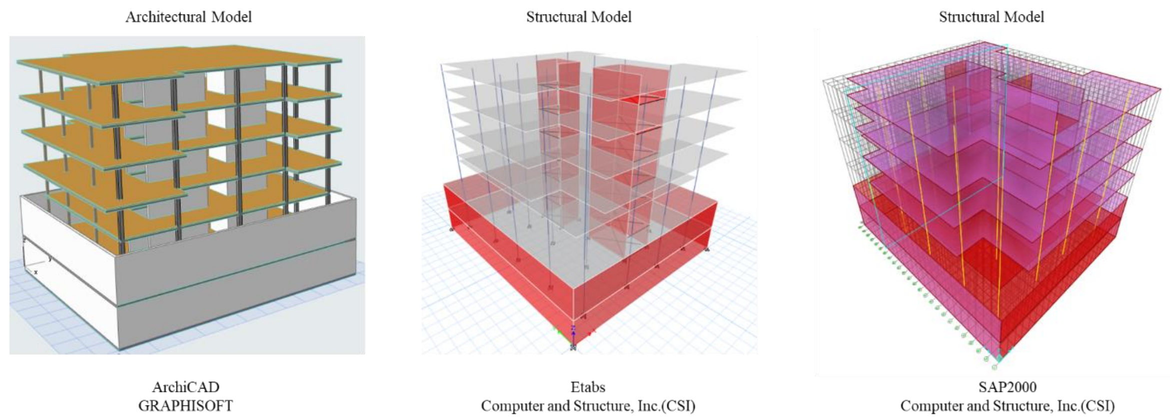


Figure 14: Models created based on the selected software tools

Table 6: Data required by ArchiCAD.

Project details	Unit system		Measuring unit type (SI/ Metric)
	Project & engineering description		ID
			Contact info
			Description
Building Structure	Grid spacing		No of grid lines (x direction, y direction)
			Spacing of Grid (x direction, y direction)
	Simple story data		No of stories
			Typical story height
			Bottom story height
Visualising (Geometry)	Drawing functions		To represent members and surfaces
Column	Geometry & Positioning	Structure	Structure Type (circular, complex profile)
			Dimensions (width and depth)
			Building material (concrete/steel)
			Complexity (Vertical or slanted)
		Top and home story	Top offset to top linked story
			Bottom offset to home story
			Assigning to column layer
Wall	Geometry & Positioning	Structure	Structure Type (composite or complex profile)
			Building material (concrete/Steel)
		Geometry method	Straight /Trapezoid/ Polygonal
		Wall complexity	Straight / Slanted/ Double
		Top and home story	Top offset to top linked story
			Bottom offset to home story
			Assigning to Wall layer
		Reference line	Outside face / Inside face/ Centre
Slab	Geometry & Positioning		Building material (concrete/steel)
			Edge angle
	Reference plan location		Top / Bottom
			Slab thickness
			Assigning to Floor layer

Table 7: Data required by SAP2000 and Etabs

Data	Sub-data				Attribute
Project details	Unit system				Measuring unit type (SI/ Metric)
	Project & engineering description				ID Contact information Description
Building Structure	Grid spacing				No of grid lines (x direction, y direction) Spacing of Grid (x direction, y direction)
	simple story data				No of stories Typical story height Bottom story height
Visualising (Geometry)	Drawing functions				To represent members (linear element) & surfaces (planar element)
Material	Defining properties	Material properties	General data		Material Name Material Type Directional symmetry type
			Material Weight and Mass		Weight/ unit volume Mass / unit volume
			Mechanical property data		Modulus of elasticity Compressive strength Poisson's ratio Coefficient of thermal expansion Shear modulus Yield Stress Tensile Strength
		Section properties	Frame section (linear element)	General data	Property name material
				Shape	Section shape
				Section dimensions	Depth & Width or Diameter
			Slab section (planar element)	General data	Property name Slab material Modelling type (Shell element/Membrane element/Plate element)
				Property data	Type of slab (Shell element/Membrane element/Plate element) Thickness of slab
			Wall section (planar element)	General data	Property name Slab material Modelling type(shell/membrane/plate)

		Property data:	Thickness of wall
Load data	Load patterns	Self-weight	Type Self-weight multiplier
		SIDL (superimposed dead load)	Type Self-weight multiplier
		LL (live load)	Type Self-weight multiplier
	Load assignment	Joint load	The load can be of type point
		Frame load	The load can be of type linear
		Shell load	The load can be of type planar (uniform load, area load)
		Load direction	Direction (Gravity or Local)
Load conditions	General data		Load combination name Combination type
	Define combinations of load cases		Load name Scale factor
Boundary conditions	Restraints		Restraints in global directions (rigid/ pin /roller)
	Springs		-
	Diaphragm		Joint Shell (rigid diaphragm)
Other data	Meshing Shell (Wall mesh/ Floor mesh)		To divide the floor into small areas

The architectural model provides the first wave of data required by the structural model that primarily includes geometrical and material information (Hu, Zhang and Deng, 2008). Although a structural model requires various information such as structural elements, mechanical connectivity, support conditions, mechanical properties and loadings, an architectural model can only provide structural elements, materials, and connectivity data while the other information needs to be added manually (Chen *et al.*, 2005a). Hence, the structure model requires information that the architect might not define since it is out of the architectural design scope. Several researchers pointed out the data needed and the differences between architectural and structural design. An architect is more concerned with the spatial arrangement of building elements such as shape, layout, the location of the geometry, member section profiles and material data, while a structural engineer focuses on the mechanical properties of elements, building behaviour and stability (Chen *et al.*, 2005b; Taylor *et al.*, 2009; Liu, Li and Zhang, 2010; Qin, Deng and Liu, 2011; Hu *et al.*, 2016). Furthermore, Wang, Yang and Zhang (2015) mentioned that although the same element might be represented in both the architectural and structural models, the detailed data of that element can be different. This difference is due to the unique use of that element in specific domain disciplines. The results identified in this section validate the previous statements since ArchiCAD is an architectural software tool built according to the architectural perspective to do a specific task.

On a similar topic, Wan, Chen and Tiong (2004) evaluated the IFC2X2 schema for the structural analysis field by looking into the information required by SAP2000's structural analysis software. They stated that the data needed by SAP2000 was geometry data, section data, material data, load data, and load combinations. A structural model can be represented as a simplified analysis model, which provides uncomplicated information (Hu, Zhang and Deng, 2008). Whereas a sophisticated analysis model, which is based on a finite element model (FEM), is used for complex analysis (Hu, Zhang and Deng, 2008). Moreover, to have a perfect structural model, it is necessary to consider the way two elements are connected. For instance, a structural element in a structural model is represented as a linear element or a planar element and the two elements are connected through the centroid (Chen *et al.*, 2005a); otherwise, it will result in instability. Whereas in the architectural model, the section is represented as a 3D-shaped section, and two elements can be connected face to face, edge to edge or centre to the centre since that will not affect the design. The finite element mesh (FEM) plays a vital role in the structural analysis model accuracy (Hassanien Serror *et al.*, 2008; Qin, Deng and Liu, 2011). For instance, meshing is based on dividing an element into small elements. The smaller the element size is, the more accurate results can be generated. However, this can increase the time of analysis. Exchanging mesh data is a difficult task and hard to preserve. Therefore, each structural analysis tool is designed to perform such advanced analysis; hence, this part will not be covered in this research. The research above supports the results in Tables 6 and 7, where data is more fixated on geometry and material in the architectural model. Whereas data related to mechanical properties (unit weight, Modulus of elasticity, Compressive strength, shear modulus), boundary conditions (fixed, pinned, and roller supports), meshes, loads (Self-weight, live load, superimposed dead load), and load combination can be categorised as specific data essential for the structural domain.

Furthermore, Yaman and Taş (2007) pointed out the information required in the cost estimation area. For instance, they stated that project details, including Project number, address, description, and project owners' details, are required in addition to building structure information such as site, building, building storeys, and spaces. Not forgetting to mention the total gross construction area of the project, the quantity of the Building Elements, Building Element Types, and product cost information, bearing in mind that the measurement units used in a project are important since it has relation to the unit price information selected by the stakeholders. However, despite the fact that the dataset for cost estimates is produced from the architectural model, the architectural model could provide only a few data such as space, element area, floor height, building parameter and gross area (Sherif, Jinkook and Chuck, 2011).

4.1.2. Similarities and differences between the exported MVDs

There are three types of data exchange that were implemented to improve interoperability in the AEC industry: Firstly, exchange using the same authoring tools. Secondly, exchange through an API. Thirdly, exchange using a common data schema (Aldegeily, Hu and Ph, 2018; Ramaji and Memari, 2018). Each of these workflows is briefly discussed in this section. The first type assisted in overcoming interoperability issues. However, the solutions that were provided are only for vendors' applications that belong to the same companies, for instance, Autodesk software tools packages. Such a type of data exchange could limit data exchange if stakeholders opt to use tools from different vendors. The second type is achieved by developing an API. For instance, CSIRevit was developed to link Revit to CSI tools such as ETABS, SAP2000, and SAFE to enhance collaboration between architectural designers and structural engineers (Aldegeily, Hu and Ph, 2018). This type of development might require access to the internal structure of software tools and excellent computing skills from the developers. The third type, which is the focus of this research, uses a unified file format such as IFC. Lai and Deng (2018) stated that using IFC as data exchange is viable since it can reduce the number of solutions developed. However, it is still facing some problems (Lai and Deng, 2018) and demands to be supported by additional tools or platforms (Chen et al., 2005a).

Since the IFC schema covers various domains, it is not convenient to implement the entire schema in software vendors. Consequently, a concept such as MVD has been used as a solution to enhance data exchange. According to the BuildingSMART database (2021b), several MVDs have been officially released. The MVDs that are discussed in this section are IFC2x3 Coordination View 2.0 (CV), IFC2x3 Structural Analysis View (SA), IFC4 Design Transfer View (DTV), IFC4 Reference View (RV), in addition to some other exported MVDs from ArchiCAD such as General Export, structural analysis, and CostX Export.

The architectural coordination view (CV) is comprehensively utilised in most BIM software tools (Hu *et al.*, 2016) and defines the elements of a building as volumetric objects (Ramaji and Memari, 2018). However, although only a few entities can be imported using this View since not all data are valid to ETABS and SAP2000 tools (Computers and Structures, 2013), the Coordination View could be utilised to create a basic structural model (Ramaji and Memari, 2016, 2018). On the other hand, the structural analysis view (SA), which defines a building in terms of nodes, elements, and loads (Computers and Structures, 2013), is used to transfer the structural analysis model to other structural analysis tools (Hu *et al.*, 2016). However, it is not supported by many software tools (Computers and Structures, 2013), and it only covers the data required by the structural domain (design and analysis) (Hu *et al.*, 2016). It can exchange data related to boundary conditions, loads, load combinations, connections, and other structural data.

Moreover, based on the IFC4 schema, BuildingSMART developed Design Transfer View (DTV) and IFC4 Reference View (RV). The Design transfer view was developed to share parametric elements for further editing and coordination. Whereas, Reference view, which is a subset of the Design transfer view (Ramaji and Memari, 2018), is used to share geometry representation for model referencing and clash detection, knowing that the model can be imported as a read-only model. Geometrical data can be necessary for coordination and clash detections. However, the analysis models necessitate additional information to establish a full model; therefore, this MVD will not provide a complete analysis model. Ramaji and Memari (2018) considered that the design transfer view could be assumed to replace the coordination view.

Furthermore, “CostX Export” is used for cost estimation purposes. Sherif, Jinkook and Chuck stated that “ although more cost estimating applications are moving toward IFC compatibility, IFC does not solely cover all components required to generate an estimate, as estimating requires not only quantity take-off data, but other types of associated databases. These carry labour, material and equipment unit costs, location parameters, market conditions, and other factors that require continuous adjustment and updating” (Sherif, Jinkook and Chuck, 2011). Jadid and Idrees (2007) stated that using the IFC schema in the cost estimating process can provide the stakeholder with product data that form a necessary aspect to calculate the cost of a project. However, it needs to be supported by additional methods and formats to enhance its performance.

4.1.2.1. Comparison using IFC file analyser

Based on the models that were created previously, IFC files representing those MVDs were exported, Figure 12. The first Five are exported from the ArchiCAD and were named as follows: “General Export”, “Structural Analysis”,” DTV”,’ RV’ and “CostX Export”, bearing in mind that the ArchiCAD software tool allows the export of several custom IFC files for different design tasks and end-users. However, these functionalities are specific to this tool. “Architecture Coordination View” (CV) and “Structural Analysis View” (SA) were exported from SAP2000 and Etabs. Following the discussion mentioned above, the exported MVDs were further imported into the IFC File analyser to identify IFC entities, relations, and properties in the exported Models and to show the similarities and differences between them. The final results are concluded in Table 8. Further details about the insides of those MVDs are as follows:

Table 8: Similarities and differences between some of the existing MVDs using IFC file analyser.

Data classification	Entities	ArchiCAD					Etabs & SAP2000	
		General Export	Structural analysis	CostX	DTV	RV	CV	SA
Building element & type	IfcColumn	√	√	√	√	√	√	-
	IfcColumnType	√	√	√	√	√	-	-
	IfcSlab	√	√	√	√	√	√	-
	IfcSlabType	√	√	√	√	√	-	-
	IfcWall	√	√	√	√	√	√	-
	IfcWallType	√	√	√	√	√	-	-
	IfcStructuralCurveMember	-	-	-	-	-	-	√
	IfcStructuralSurfaceMember	-	-	-	-	-	-	√
Project details & Spatial structure	IfcBuilding	√	√	√	√	√	√	√
	IfcBuildingStorey	√	√	√	√	√	√	√
	IfcOwnerHistory	√	√	√	√	√	√	√
	IfcProject	√	√	√	√	√	√	√
	IfcSite	√	√	√	√	√	√	√
	IfcActorRole	-	-	-	-	-	√	√
	IfcUnitAssignment	√	√	√	√	√	√	√
	IfcSIUnit	√	√	√	√	√	√	√
	IfcRelAggregates	√	√	√	√	√	√	√
	IfcRelContainedInSpatialStructure	√	√	√	√	√	√	√
Property Sets	IfcRelDefinesByProperties	√	√	√	√	√	-	-
	IfcRelDefinesByType	√	√	√	√	√	-	-
	IfcPropertySet	√	√	√	√	√	-	-
	IfcPropertySingleValue	√	√	√	√	√	√	√
	IfcMaterialProperties	-	-	-	√	√	√	√

		IfcElementQuantity	-	-	√	-	-	-	-
		IfcQuantityArea	-	-	√	-	-	-	-
Quantities		IfcQuantityCount	-	-	√	-	-	-	-
		IfcQuantityLength	-	-	√	-	-	-	-
		IfcQuantityVolume	-	-	√	-	-	-	-
		IfcMaterial	√	√	√	√	√	√	√
		IfcRelAssociatesMaterial	√	√	√	√	√	√	√
		IfcMaterialLayerSet	√	√	-	√	-	√	-
		IfcMaterialProfileSet	-	-	-	-	-	√	√
		IfcMaterialDefinitionRepresentation	√	√	√	√	√	-	-
		IfcStyledRepresentation	√	√	√	√	√	-	-
Material		IfcStyledItem	√	√	√	√	√	-	-
		IfcColourRgb	√	√	√	√	√	-	-
		IfcSurfaceStyle	√	√	√	√	√	-	-
		IfcMaterialList	-	-	√	-	√	-	-
		IfcMaterialLayer	√	√	-	√	-	√	-
		IfcMaterialLayerSetUsage	√	√	-	√	-	√	-
		IfcMaterialProfile	-	-	-	-	-	√	√
		IfcMaterialProfileSetUsage	-	-	-	-	-	√	√
		IfcArbitraryClosedProfileDef	√	√	-	√	-	√	-
		IfcExtrudedAreaSolid	√	√	-	√	-	√	-
	Extrusion	IfcRectangleProfileDef	√	√	-	√	-	√	√
		IfcCircleProfileDef	√	√	-	√	-	√	√
		IfcAxis2Placement3D	√	√	√	√	√	√	√
		IfcDirection	√	√	√	√	√	√	√
		IfcCartesianPoint	√	√	√			√	√
		IfcFacetedBrep	-	-	√	-	-	-	-
Geometric Representations		IfcFaceOuterBound	-	-	√	-	-	-	-
		IfcFaceBound	-	-	√	-	-	-	√
		IfcFace	-	-	√	-	-	-	-
		IfcClosedShell	-	-	√	-	-	-	-
	B -Rep	IfcPolyLoop	-	-	√	-	-	-	-
		IfcProductDefinitionShape	√	√	√	√	√	√	√
		IfcShapeRepresentation	√	√	√	√	√	√	-
		IfcLocalPlacement	√	√	√	√	√	√	√
		IfcGeometricRepresentationContext	√	√	√	√	√	√	√
		IfcRelConnectsPathElements	√	√	√			-	-
		IfcStructuralAnalysisModel	-	-	-	-	-	-	√
		IfcRelAssignsToGroup	-	-	-	-	-	-	√
		IfcRelConnectsStructuralActivity	-	-	-	-	-	-	√
		IfcStructuralPlanarAction	-	-	-	-	-	-	√
		IfcBoundaryNodeCondition	-	-	-	-	-	-	√
		IfcRelConnectsStructuralMember	-	-	-	-	-	-	√
Structural Information		IfcStructuralPointConnection	-	-	-	-	-	-	√
		IfcStructuralLoadPlanarForce	-	-	-	-	-	-	√
		IfcStructuralLoadCase	-	-	-	-	-	-	√
		IfcStructuralLoadGroup	-	-	-	-	-	-	√
		IfcRelAssignsToGroupByFactor	-	-	-			-	√
		IfcRelServicesBuildings	-	-	-	-	-	-	√
		IfcTopologyRepresentation	-	-	-			-	√

- **Building element & type.** Object type was shown in all the exported files from the ArchiCAD, while that was not the case with the IFC files exported from Etabs and SAP2000. The building members were represented as IfcColumn, IfcSlab and IfcWall in all MVDs except for the structural analysis view exported from Etabs and SAP2000. The building members were represented as IfcStructuralCurveMember for linear elements such as Column and IfcStructuralSurfaceMember for surface elements such as walls and Slabs, Figure 15. For cases where the IFC does not provide a particular modelling construct, the language contains a mechanism for modelling IfcProxy Elements (Steel, Drogemuller and Toth, 2012). However, this was not shown in any model. It is more likely to happen when an MVD is imported into another software and re-exported again, which might be because some entities cannot be recognised by the receiving tool. Hence the tool will convert them to the IfcBuildingElementProxy entity.

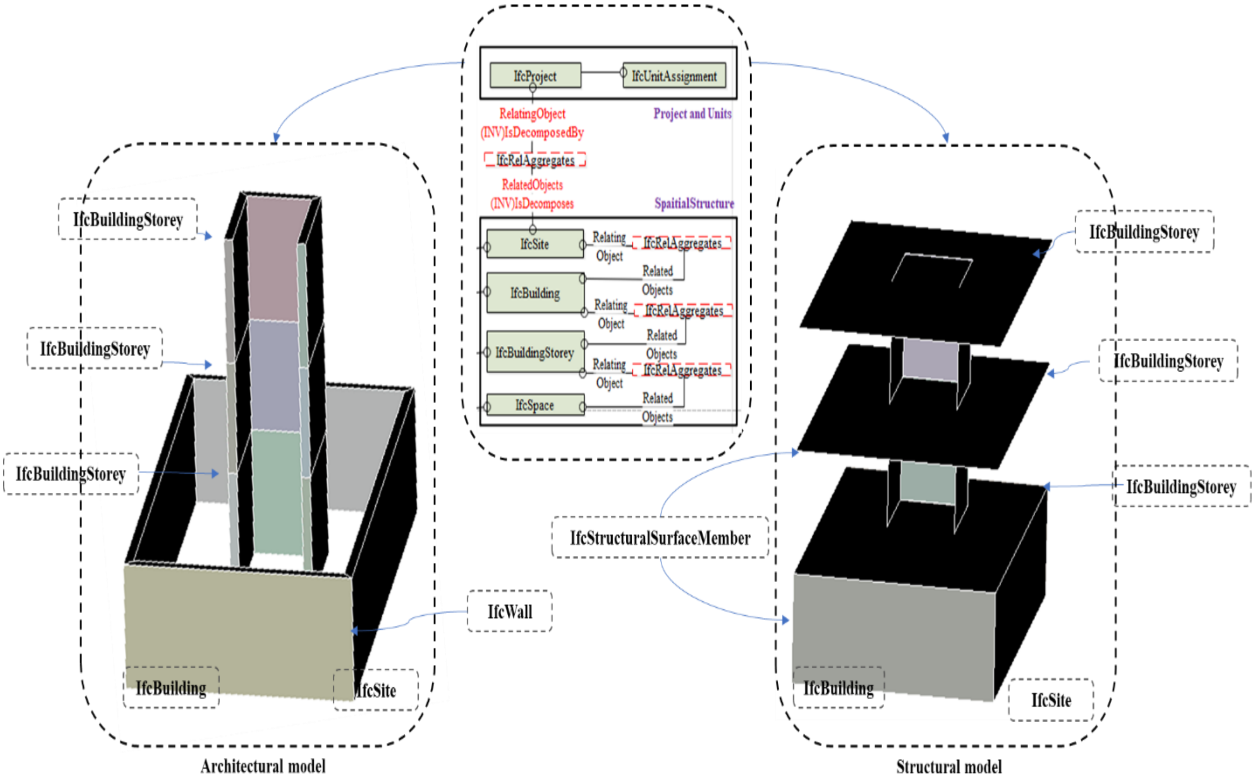


Figure 15: Architectural model versus Structural model in IFC format.

- **Project details and Spatial structure**. The IfcProject entity is used to contain project data and the units used in the project, which is represented by IfcUnitAssignment that specifies a set of units (BuildingSMART, 2021a). Data related to project ownership is represented in the IFC schema by the

IfcOwnerHistory entity and obtained through the IfcProject entity (BuildingSMART, 2021a). IfcActorRole, which assigns a role that an actor performs in a project (BuildingSMART, 2021a), was only used in the exported files from Etabs and SAP2000. Furthermore, the spatial structure is used to deliver the project structure to form a building. Entities contained by the spatial structure are IfcSite, IfcBuilding, IfcBuildingStorey and IfcSpace (BuildingSMART, 2021a). These entities are included under the IfcSpatialStructureElement relation entity in the IFC schema (BuildingSMART, 2021a). The IfcRelAggregates relation entity, Figure 15, a special type of IfcRelDecomposes, shows the relation among IfcProject, IfcSite, IfcBuilding, IfcBuildingStorey and IfcSpace (BuildingSMART, 2021a). Furthermore, the IfcRelContainedInSpatialStructure relation relates elements to spatial structure (BuildingSMART, 2021a). For instance, it is used to relate IfcElement, such as IfcColumn, to IfcBuildingStorey. It was noticed that all these entities were shown in all the exported MVDs.

- **Properties**. IfcPropertySet is used to hold properties within a property tree (BuildingSMART, 2021a). Building elements are linked to their properties following two paths: direct link using IfcRelDefinesByProperties and indirect link using IfcRelDefinesByType (Zhang and El-gohary, 2020; BuildingSMART, 2021a). These two entities are relation entities that relate an element to a property set (IfcPropertySet) and elements to an element type (IfcTypeObject) that has a property set (IfcPropertySet) (BuildingSMART, 2021a), respectively. For instance, by using the IfcRelDefinesByProperties relationship entity, the IfcWall entity can be related to an instance of IfcPropertySet. Whereas IfcRelDefinesByType allows for the assignment of one type of information, for instance, (IfcWallType) to a single or many elements (IfcWall) (BuildingSMART, 2021a). It was shown that these relation entities were not shown in the file exported from Etabs and SAP2000. Instead, IfcPropertySingleValue, a sub-entity of IfcPropertySet, defines a property object with a single numeric or descriptive value (BuildingSMART, 2021a) exported directly. Furthermore, IfcMaterialProperties is used to assign a set of material properties to associated material definitions.

- **Quantities.** IfcElementQuantity defines a “set of derived measures of an element’s physical property” (BuildingSMART, 2021a). To relate this entity to the elements, IfcRelDefinesByProperties is used. IfcElementQuantity is used to obtain properties such as length (IfcQuantityLength), area (IfcQuantityArea), volume (IfcQuantityVolume), and others. Since these properties are related to cost estimation, they were only shown in the CostX export.

- **Material.** IfcMaterial is the basic entity for material designation and definition (BuildingSMART, 2021a). Similar to IfcRelDefinesByType relation entity, IfcRelAssociatesMaterial relation is used to relate an instance of IfcMaterial to an element or element type (BuildingSMART, 2021a). A single material can be assigned directly or represented by other set entities such as IfcMaterialLayerSet, and IfcMaterialProfileSet (BuildingSMART, 2021a). IfcMaterialLayerSet was shown in General Export, Structural Analysis, and DTV exported from the ArchiCAD and in CV exported from Etabs and SAP2000. Whereas IfcMaterialProfileSet was only shown in the files exported from Etabs and SAP2000. IfcMaterialDefinitionRepresentation is used to provide presentation information associated with IfcMaterial (BuildingSMART, 2021a). It can apply different presentation styles for different representation contexts. However, this entity was not shown in the files exported from Etabs and SAP2000. Material colour is specified in IfcStyledItem. Nevertheless, this entity and its related entities were not shown in the Etabs and SAP2000. In contrast, they were shown in all other models. Furthermore, the IfcMaterialList, which is a list of the different materials that are used in an element (BuildingSMART, 2021a), was shown only in CostX and RV.

- **Geometric Representations.** The variety of geometric representations of structural elements in different tools might be a reason behind data exchange issues, and it can be more related to the software's internal mapping schema. The IFC entities used for the geometry representation are of three types: Extrusion (Swept Solid) (Chen *et al.*, 2005a), Boundary representation (B-rep) (Chen *et al.*, 2005a), which is widely used in computer graphics (Hu *et al.*, 2016), and Constructive solid geometry (CSG). The IfcExtrudedAreaSolid entity inherits entities such as Swept Area (IfcArbitraryClosedProfileDef, IfcRectangleProfileDef, IfcCircleProfileDef), position (IfcAxis2Placement3D), Extruded Direction (IfcDirection) and Depth (IfcPositiveLengthMeasure). Extrusion type was used in all exported files except CostX, RV and SA. However, in the SA, IfcRectangleProfileDef and IfcCircleProfileDef were shown. These two entities are a subtype of IfcProfileDef, which is used to define section profile. Instead of the IfcExtrudedAreaSolid entity, IfcFacetedBrep with IfcClosedShell and IfcFace entities were used in the cost model. The IfcFacetedBrep is used to represent planar surfaces only. Boundary representation is used in clash detection and volume calculation cases, which justify using this type in the cost MVD. It is also shown that extrusion type is used in Coordination View. Furthermore, an Element has two geometric representation attributes. Object placement is denoted by IfcObjectPlacement, and object representation is denoted by IfcProductRepresenation (Chen *et al.*, 2005a). The Position and dimensions of an element are determined with the IfcObjectPlacement, which is provided for an object with a shape representation (IfcProductDefinitionShape) (Chen *et al.*, 2005a). A subtype entity inherits all the attributes from its supertype. For instance, IfcProductDefinitionShape is a subtype of IfcProductRepresenation. Hence, all the Information in IfcProductRepresenation will be assigned to IfcProductDefinitionShape automatically. Furthermore, the object placement can have different types, such as an “absolute”, “relative”, or “constrained”.

- **Structural Information.** IfcStructuralAnalysisModel is used to assemble all information needed to represent a structural analysis model (BuildingSMART, 2021a). Thus, it was shown only in the SA MVD exported from the structural analysis software and was not shown in the Structural analysis MVD exported from ArchiCAD. It comprises a structural element, structural connection, structural activities, and others. The relationship entity IfcRelAssignsToGroup is used to relate the structural analysis model to the structural member’s entity (IfcStructuralMember). IfcStructuralMember is a supertype entity for IfcStructuralCurveMember (Linear elements) and IfcStructuralSurfaceMember (planar elements) (Wang, Yang and Zhang, 2015). The structural information comprises structural loads, boundary conditions, load cases (IfcStructuralLoadCase) and load combinations (IfcStructuralLoadGroup). For instance, the load is represented by an “IfcStructuralLoadPlanarForce” entity as part of an “IfcStructuralPlanarAction” instance. IfcStructuralPlanarAction entity is used to represent the load, and the IfcRelConnectsStructuralActivity relationship entity is used to relate structural elements such as IfcStructuralSurfaceMember to an activity (Wang, Yang and Zhang, 2015). Structural connection is represented by IfcStructuralConnection, divided into a point (IfcStructuralPointConnection), line and face connections(Wang, Yang and Zhang, 2015). The restraints of joints or release of frame elements are obtained from IfcBoundaryNodeCondition. As shown in Table 8, these entities are used only in the SA view exported from the Etabs and Sap2000. Whereas the structural analysis view exported from ArchiCAD did not show these entities. Hence, these data are specific data required in the structural domain and can be only provided by structural analysis tools. In summary, information associated with the structural behaviour of elements, connectivity, boundary conditions, mechanical properties of the material, and others is not included in the coordination view, DTV, and RV. These MVDs can only deliver the physical part of a model.

4.1.2.2. Comparison based on similarity rate between MVDs

In this section, the similarity rate equation below, adopted from Lee et al. (2011), was utilised to compare the exported MVDs to see how much information is maintained and shared between them. According to Lee et al. (2011), the similarity rate from File A to File B is defined as “the number of matching instances in File A divided by the total number of instances in File A”. For instance, the number of matching entities in “General Export” MVD and DTV is divided by the total number of entities in the “General Export” MVD. The comparison in this section is an extension of Section (4.1.2.1). Hence, only the main entities shown in Table 8 are used to calculate the similarity ratio.

$$Similarity\ rate\ (\%) = \frac{\text{Number of matching instances in File A}}{\text{Total number of instances in File A}} \times 100\%$$

In Table 9, one to one comparison, several similarity rates of 100 per cent were shown. Most of those percentages were recorded when comparing an MVD to itself. However, among all the results, only two cases had a similarity rate of 100 per cent, which was recorded between “General Export” MVD and “Structural Analysis” MVD. Those two MVDs are used for different purposes but belong to the same vendor. On the other hand, the “SA Etabs & SAP2000” has an almost 50% similarity rate compared to the “General Export” MVD, the “structural analysis” MVD, the “CostX ArchiCAD” MVD and the “DTV ArchiCAD” MVD except for the similarity rate from “SA Etabs & SAP2000” to “CV Etabs & SAP2000” was more than 60%. Moreover, if we compare the “General Export” to the “CV Etabs & SAP2000”, the similarity rate between them is high, but not 100 per cent, which in essence needs to be 100 per cent matching since they are representing the same MVD that is used for the same purpose. Furthermore, “RV ArchiCAD” has shown high similarity rate (above 90%) to several MVDs such as “General Export" (94.12 %),” DTV ArchiCAD” (97.1

%),"Structural Analysis" (94 %), and ”CostX ArchiCAD” (97 %). However, these were not the same results recorded with other MVDs such as “SA Etabs & SAP2000” (52.94%) and “CV Etabs & SAP2000” (64.7%).

Table 9: One to one comparison based on similarity rate equation

	General Export ArchiCAD	Structural Analysis ArchiCAD	CostX ArchiCAD	DTV ArchiCAD	RV ArchiCAD	CV Etabs & SAP2000	SA Etabs & SAP2000
General Export ArchiCAD	100	100	73.91	97.5	94.12	88.23	48.78
Structural. Analysis ArchiCAD	100	100	74	97.5	94	82.92	48.78
CostX ArchiCAD	82.93	83	100	80	97	61.76	46.34
DTV ArchiCAD	95.12	95.12	69.56	100	97.1	82.35	48.8
RV ArchiCAD	78	78	71.73	82.5	100	64.7	43.9
CV Etabs & SAP2000	73.17	70.73	45.65	70	64.7	100	60.97
SA Etabs & SAP2000	48.78	48.78	41.3	50	52.94	73.5	100

On the other hand, “one to many” comparisons were made by comparing one MVD among the other selected MVDs to give an overall evaluation, Figure 16. The MVDs with the highest similarity rate to other MVDs were “RV ArchiCAD” (58.82%) and “CV Etabs & SAP2000” (58.82%), followed by “DTV ArchiCAD” with a percentage of 50%. The above result supports the statement made by Ramaji and Memari (2018), where the RV can be a subset of the DTV and other MVDs. Hence, the similarity rate of “RV ArchiCAD” to “DTV ArchiCAD” was high. Furthermore, “CV Etabs & SAP2000” also showed a high similarity rate. Although the implementation of the “CV Etabs & SAP2000” is specific for Etabs and SAP2000”, it showed a high similarity rate similar to “RV ArchiCAD”.

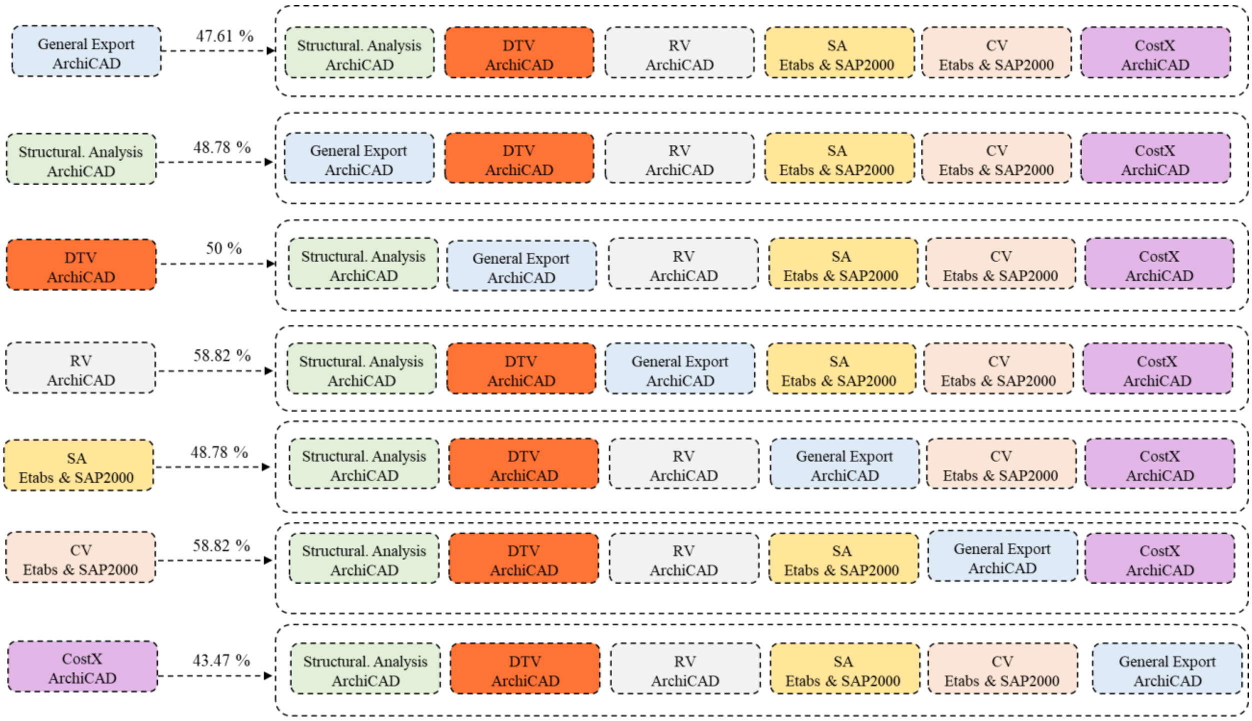


Figure 16: One to many comparisons based on similarity rate equation.

To sum up, the analysis that was carried out showed that some MVDs could be represented as a subset of other MVDs, which point out that although they were developed independently and not interconnected, they share some common information that is not exclusive to a specific domain. That common information can be standardised to form a unified data sets that can form the foundation for data exchange solutions. Moreover, some of these MVDs are built and restricted to a specific domain, which justifies why some cases, such as the case of "CostX ArchiCAD" to "SA Etabs & SAP2000", showed the lowest percentage (41.3%). Hence, defining a "single truth of information" shared throughout the building lifecycle can help reduce duplication of developing efforts and overcome inconsistency while developing data exchange solutions. It can help the developer to use the same standardised mapped minimum data set when they create their idea and extend this mapped data set to fit their specific use case, which can eliminate the time wasted on rebuilding this common information again while developing a new subset, and also reduces the number of solutions developed. The following section will explain this unified data set concept further, and some of this common information will be represented in the following section.

4.2. Correlation between architectural, structural, and cost models

Lee defined semantic intersection as “a set of information items in different data sets that are functionally dependent” (Lee, 2011). This definition also aligns with the concept of “dependency modelling”, which states that to advance to the next design stage, one model will mainly rely on data shared with another model or several models since these models are interconnected and share some commonalities. Therefore, to better understand how different models can work together, this thesis adopted Lee's approach (2011).

In this research, the example considered to elaborate on the intersections between data sets was based on the data collected in stage 1 (Section 4.1.1), the exported MVDs in stage 2 (Section 4.1.2) and the concepts adopted from Lee (2011). The architectural model is considered as the source model, whereas the structural and cost models represent the models delivered to the end-users.

- **One to One data exchange.** This type of data sharing represents the commonality shared between two models or MVDs, associated with two distinct disciplines, and it is denoted by CsharedDS $_{1\text{-}1}$ (commonly shared data set), Figure 17.

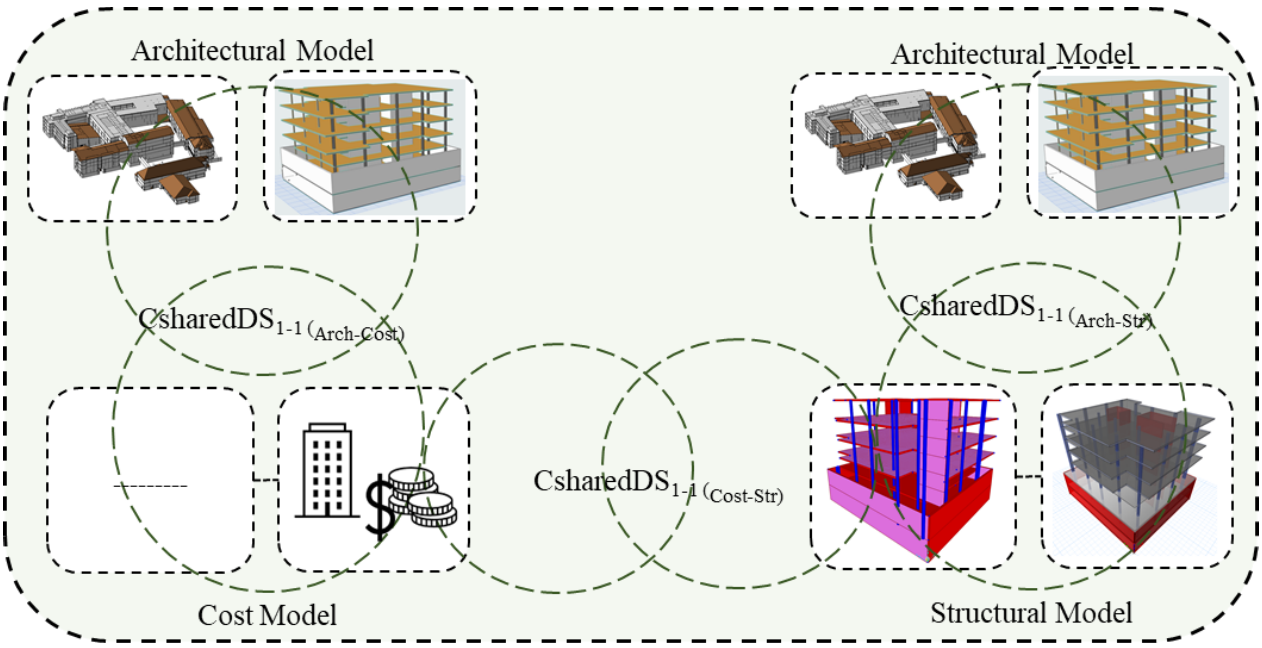


Figure 17: One to one data exchange between two different disciplines.

Architectural data set (ADS) = { project details, unit system, building, building storey, building site, building space, connectivity data, building elements, building element types, geometry data, member section profiles, material data, layout}

Structural data set (SDS) = {project details, unit system, building, building storey, building site, building space, structural elements, Structural Element types mechanical connectivity/ boundary conditions (fixed, pinned, and roller supports), loads (Self-weight, live load, superimposed dead load), load combination, meshes, mechanical properties (unit weight, Modulus of elasticity, Compressive strength, shear modulus), Section properties, layout }

Cost data set (CDS) = {project details, unit system, building, building storey, building site, building space, gross area, element area, building elements, building element types, geometry data, product cost information, the quantity of the building elements}

Regular intersection (∩) was defined as a “ set of information items in different data sets that share the same item” (Lee, 2011). Taking the CsharedDS $_{1\text{-}1}$ between the models selected in this research as an example, some of the data shared among those disciplines/ MVDs can be represented as follows:

CsharedDS $_{1\text{-}1\ (\text{Arch-Str})}$ = ADS ∩ SDS = { project details, unit system, building, building storey, Site, space, structural elements, geometry data }

CsharedDS $_{1\text{-}1\ (\text{Arch-Cost})}$ = ADS ∩ CDS = { project details, unit system, building, building storey, Site, space, building elements, geometry data }

CsharedDS $_{1\text{-}1\ (\text{Str-Cost})}$ = SDS ∩ CDS = { Project details, Unit system, Building, building storey, Site, space, structural elements, geometry data }

The same information can be represented in the IFC format as follows:

ADS ∩ SDS = { IfcProject, IfcUnitAssignment, IfcBuilding, IfcBuildingStorey, IfcSite, IfcSpace}

ADS ∩ CDS = { IfcProject, IfcUnitAssignment, IfcBuilding, IfcBuildingStorey, IfcSite, IfcSpace}

SDS ∩ CDS = { IfcProject, IfcUnitAssignment, IfcBuilding, IfcBuildingStorey, IfcSite, IfcSpace}

As Lee (Lee, 2011) mentioned, “two different systems do not always use the same terms to mean the same thing or have the same internal data structure.” (Lee, 2011). Therefore, if two information items have two different definitions but are used for the same intention, they are also exchangeable, and this is called semantic intersection ∩* or exchangeable synonym data. For instance, taking the connectivity data as an example, connectivity is represented as “connectivity data” in the architectural data set, whereas “mechanical connectivity” represents connectivity in a structural data set. This semantic intersection can be represented as :

ASD ∩* SDS = { <connectivity data : mechanical connectivity> }

However, there is always a set of information that cannot be exchanged, which can be direction sensitive or not defined since it is out of the sender's design scope. Technology such as ontology is utilised in this research to provide that additional information. For instance, in the case of cost estimation, the gross area is important. However, in some cases, this information is not included in the exported model. Hence, this information can be obtained by calculating the appropriate area from the available dimensions for the appropriate section using SWRL rules. However, it can be challenging to reverse this process. Hence, it is a one-way data exchange. Such data exchange is called “the driving and driven entities” (Lee, 2011).

Moreover, in some cases, some entities can hold the same definitions or names. However, these entities are not used for the same purpose. Consequently, they cannot be exchanged and need to be excluded. Furthermore, building elements are required by all three data sets. However, to be more accurate, building elements can be divided into structural and non-structural elements. The structural elements are the load-bearing elements required for structural design, whereas the structural and non-structural elements need to be considered in a cost model to calculate the total cost of each item and the entire project. Hence the structural elements required by the structural model are a subset of the building elements sent by the architectural model.

- **Many to many data exchanges.** This data set consists of a common data set shared within three or more disciplines, denoted by CsharedDS $_{\text{Many- Many}}$. For instance, the architectural model will be linked to the structural and cost estimation models (Figure 18).

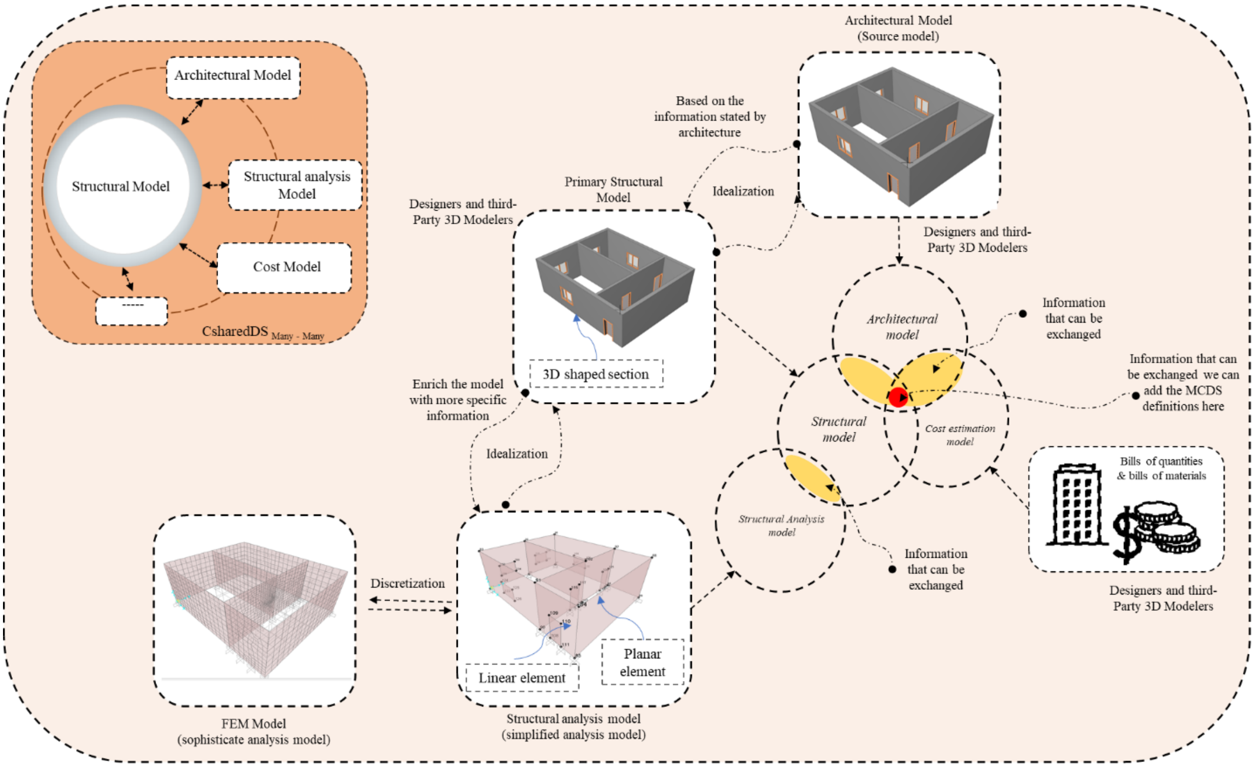


Figure 18: Correlation between architectural, structural, and cost models.

Some of the data shared among those disciplines can be represented as follows:

CsharedDS $_{\text{Many- Many (Arch-Str-Cost)}}$ = ASD ∩* SDS ∩* CDS = { project details, Unit system, building, building storey, Site, space, geometry data }

This data set comprises the minimum common data set (MCDS), or what is called the “single truth of the information”, Figure 19. This data set will be shared throughout the entire building lifecycle, where models can be transformed at the minimum commonality level. However, some of the information can be domain-specific. Therefore, on top of this MCDS, there will be several common specific data sets (CSDS) or what is represented as “partial truth of information”, which represent the common specific data for a specific domain. For instance, taking the structural domain as an example, this data set represents the common data shared among all structural software tools (Figure 19). Lee (2009) defined it as the smallest nonempty complete subset of a schema that corresponds to a given concept. However, in some cases, some tools will require more specific data, which is limited to precise tools. An example of such data can be represented as shown below and is donated by “¬” symbol, which means this information will be presented in the structural data set, but will not be included in the architectural data set or cost data set:

¬ ADS = { Loads , Load combination, Boundary conditions }

¬ CDS = { Loads , Load combination, Boundary conditions }

An example of this data set representation in IFC format is as follows:

¬ Architectural data set = { IfcStructuralLoadPlanarForce , IfcStructuralLoadGroup, IfcBoundaryNodeCondition }

More content relating to the information exchange requirements collected from a student MSc project completed at Cardiff University and a sample of the proposed MCDS data set is shown in appendix A.

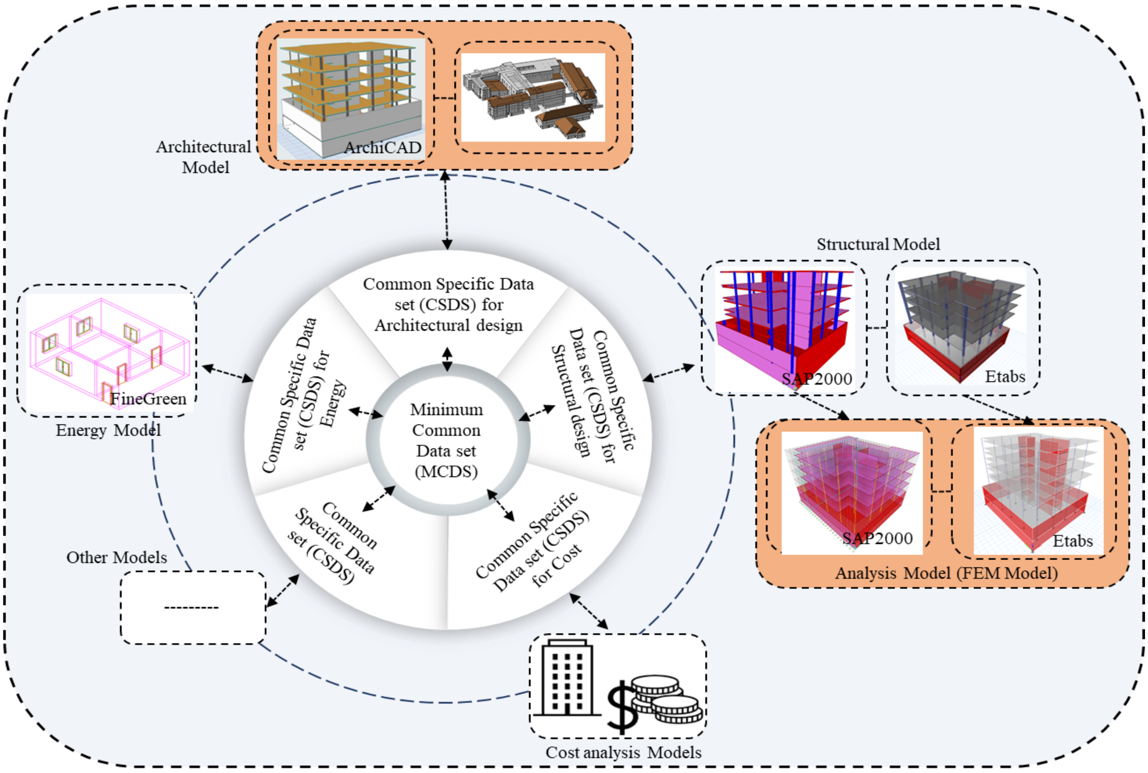


Figure 19: Many to many data exchange.

Chapter 5. Automatic data exchanging framework implementation

This chapter outlines the technical contribution of this research. Therefore, based on the analysis and review that were carried out in this thesis, a data exchanging framework that combines both a data exchange method and a multi-objective knowledge base is proposed to eliminate inefficiencies in data sharing and improve the decision-making in the early design stage by helping stakeholders in exchanging data and making decisions associated with sustainability and cost aspects while considering design conditions. The chapter is divided into two main parts: extraction as required method (Section 5.1) and multi-objective Knowledge base development (Section 5.2).

In Section 5.1, the “extraction as required” method is defined and implemented, which include several stages: (1) identifying all the required data, which was discussed earlier in Section 4.1 and 4.2; (2) mapping the required data to a machine-readable format (Section 5.1.1), in this research IFC4 schema was selected; (3) building or reusing existing libraries to develop an “extraction as required” tool (Section 5.1.2). A simple tool was developed to demonstrate the proposed method. Following Section 5.1, Section 5.2 discusses the development and implementation of the multi-objective knowledge base. Moreover, the automatic data acquisition method proposed in this research to align the developed ontology with the “extraction as required” method is also discussed in an effort to extract the critical information from an IFC file and merge them with the data presented in the developed ontology automatically. This combined framework is unlike previous research approaches where data and instances are entered manually one by one. This framework presents a more direct way to work with IFC-based BIM models in order to evaluate various aspects such as sustainability and cost while considering design criteria.

5.1. Extraction as required method

Being able to utilise a unified data set to support generating other design models could help realise significant time reductions in model generating since this unified data set can serve as a starting point for other disciplines’ end-users. Consequently, only the necessary information or a subset of information needs to be extracted according to the end users' requirements. The workflow of this section is shown in Figure 20.

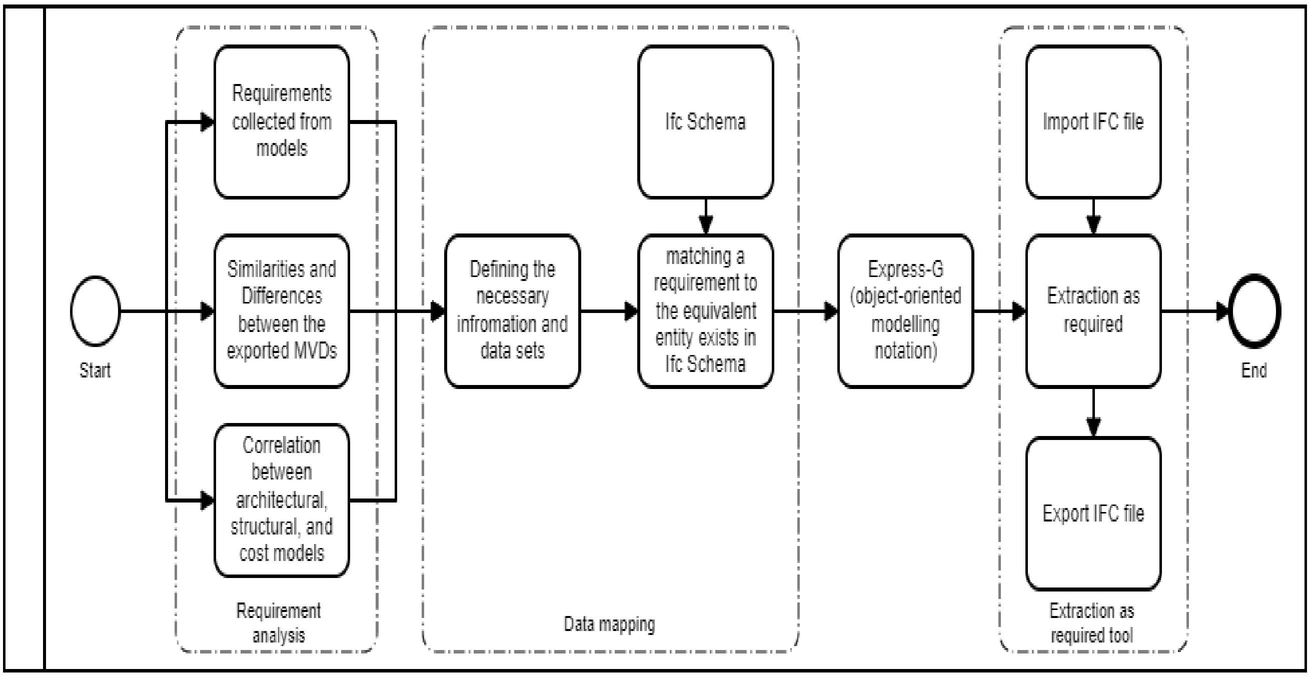


Figure 20: Workflow of the development process of the "extraction as required" method

5.1.1. Mapping requirements to a machine-readable format

Two semantics should be considered in any model exchange: (1) defining what information is needed for the exchange model to satisfy the user's need; (2) checking whether the selected schema for the model exchange provides the necessary information to satisfy the user intentions (Venugopal *et al.*, 2012). Therefore, according to the previous analysis in chapter 4, which includes requirements collected from models, Similarities and Differences between the exported MVDs, and the correlation between architectural, structural, and cost models, and by considering the data presented in appendix A, the required information has been classified into common information shared among several disciplines (MCDS) and common specific information related to a specific domain (CSDS). Bearing in mind that several iterations need to be done to refine this information.

Following this step, an object-oriented modelling notation approach based on Express-G was utilised for mapping the defined requirements to the IFC schema. Mapping is the process of matching a requirement to the equivalent entity that exists in a given selected schema. This process can differ from one developer to another. However, defining the MCDS can help the developer use the same mapped data set when they create their idea and extend this mapped data set to fit their specific use case, eliminating the time wasted on rebuilding this common information again. In this section, the structural and cost information have been selected to represent the information for a specific discipline. It is necessary to understand the types of information elements and their relationships represented in the IFC schema. Consequently, three main points should be compared to realize the relations between IFC entities and the defined requirements. First, is the defined data already included in the available version of IFC or need to be added? Secondly, are there any common definitions between different domains? Thirdly, are there any entities used for the same purpose but have different name conventions?

The Express-G data modelling language has several notations such as entity, attributes, Enumeration, Select, Cross-references, and others (Ag, 2003). An entity, which is represented by a box, is a class that contains several objects. The objects contained in a class share the same characteristics (Ag, 2003). Each entity has an attribute that describes the characteristics of an object, and it is represented as a data type (Ag, 2003). In order to connect entities to their attributes, a line is used to represent the connectivity. A solid line shows that the attribute is compulsory and cannot be neglected (Ag, 2003).

Figure 21 illustrates many entities that are related to different models. Some of these entities are common among several models. This common information was surrounded by a green dotted box. For instance, project data and the units are common information required by all disciplines. IfcProject entity is used to contain project data and the units used in the project. Data related to project ownership is represented in the IFC schema by the IfcOwnerHistory entity and obtained through the IfcProject entity. Moreover, the spatial structure is used to deliver the project structure to form a building and plays a significant part in constructing the hierarchical composition in an IFC file. Entities contained by the spatial structure are IfcSite, IfcBuilding, IfcBuildingStorey and IfcSpace (BuildingSMART, 2021a). These entities are included under the IfcSpatialStructureElement relation entity in the IFC schema (BuildingSMART, 2021a). Simultaneously, the IfcRelAggregates relation entity shows the relation among IfcProject, IfcSite, IfcBuilding, IfcBuildingStorey and IfcSpace (BuildingSMART, 2021a). Whereas the IfcRelContainedInSpatialStructure relation relates elements to spatial structure (BuildingSMART, 2021a). For instance, it is used to relate IfcElement, such as IfcColumn, to IfcBuildingStorey. Consequently, all the entities mentioned above are commonly shared by all data sets except for IfcSpace, which can depend on the use case taken for. Hence, in Figure 21, to indicate it is an optional entity, it was represented by a yellow box.

Furthermore, the Position and dimensions of an element are determined with the IfcObjectPlacement, which is provided for an object with a shape representation (IfcProductDefinitionShape) (Chen *et al.*, 2005a). A subtype entity inherits all the attributes from its supertype. For instance, IfcProductDefinitionShape is a subtype of IfcProductRepresenation. Hence, all the Information in IfcProductRepresenation will be assigned to IfcProductDefinitionShape automatically. The shape information is necessary for visualisation and also can be used to infer new information such as area and volume by taking the dimensions of an element as input. Consequently, all the entities mentioned above are commonly shared by all data sets since those dimensions will be further used later in this research to provide embodied carbon content and cost in a given element. However, an entity such as IfcStyledItem, which is used to represent the colours in a model, is marked as optional and can be specified depending on the developer.

In contrast, other entities are more related to a specific domain, which in this study, these entities represent some of the structural and cost estimation information. For instance, IfcPropertySet is used to hold properties within a property tree Building elements are linked to their properties following two paths: direct link using IfcRelDefinesByProperties and indirect link using IfcRelDefinesByType (Zhang and El-gohary, 2020; BuildingSMART, 2021a). These two entities are relation entities that relate an element to a property set (IfcPropertySet) and elements to an element type (IfcTypeObject) that has a property set (IfcPropertySet) (BuildingSMART, 2021a), respectively. By using the IfcRelDefinesByProperties relationship entity, the IfcWall entity can be related to an instance of IfcPropertySet. On the other hand, in the cost model, IfcElementQuantity is used. IfcElementQuantity is used to obtain properties such as length (IfcQuantityLength), area (IfcQuantityArea), volume (IfcQuantityVolume), and others. To relate this entity to the elements, IfcRelDefinesByProperties is used. These entities are marked as discipline-specific information and marked with red dotted boxes since they can hold values related to a specific scenario. Furthermore, IfcElementQuantity is not always provided in an IFC file. Consequently, values for the Length, Area, and Volume might not be presented in the imported model. In this research, those values will be calculated by applying SWRL rules in case they do not exist in the imported file.

Moreover, the IfcRelAssociatesMaterial relation is used to relate an instance of IfcMaterial to an element or element type (BuildingSMART, 2021a). A single material can be assigned directly or represented by other set entities such as IfcMaterialLayerSet, and IfcMaterialProfileSet (BuildingSMART, 2021a). IfcMaterialDefinitionRepresentation is used to provide presentation information associated with IfcMaterial (BuildingSMART, 2021a). It can apply different presentation styles for different representation contexts. These entities are marked as discipline-specific information.

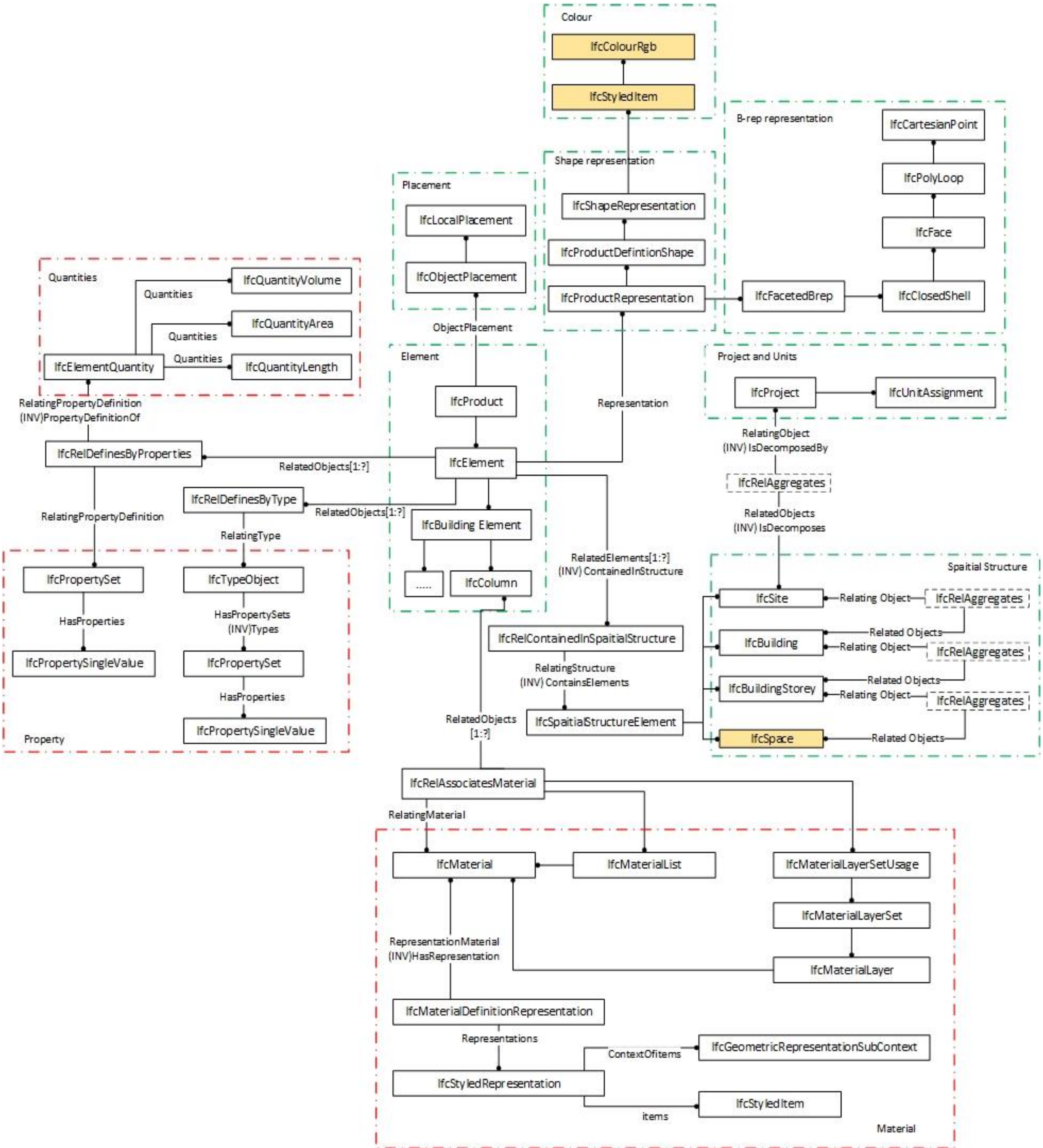


Figure 21: Data sets representation in IFC schema using EXPRESS–G.

5.1.2. Extraction as required method using IfcOpenShell library

After mapping the required data to the IFC4 schema, a data exchange method is designed in this section. Consequently, by analysing the existing research in this area, the initial functionalities required by any data extraction tool have been identified, Table 10: (1) reading and analysing the IFC schema or what is known as an IFC parser functionality; (2) information processing functionality, which can be divided into several sub functionalities such as extracting groups of data sets (one to one, one to many and many to many), extracting only a specific type of data out of a group of data sets such as element extraction, and extracting property set, which includes extracting the properties that are related to a specific element or a group of elements.

Table 10: Basic functionalities required by data extraction tool.

Functionality	Sub functionalities
Reading and parsing the IFC schema	
Information processing	Extraction as required (data sets)
	Element extraction
	Property set extraction

- **Reading and parsing the IFC schema.** Ramaji and Memari stated that "parsing an IFC file has two steps: 1) reading instances from the IFC file and 2) assigning meaning to the read values" (Ramaji and Memari, 2020). Qin, Deng and Liu (2011) stated that the process of automating the data exchange between IFC models comprises two aspects: the IFC parser, which is utilised to read the IFC physical file, and the IFC model schema that is used to create the equivalent objects defined in the IFC file in a machine comprehensible format. The proposed method can be developed by either using existing libraries or creating a tool from scratch. However, since the developed tool is a proof of concept and due to the time limit of this research, using existing libraries is more appropriate to demonstrate the proposed concept. As shown in appendix B, several efforts have focused on developing open-source libraries to help software developers work with the IFC files. However, this research aims not to discuss the functionality of these tools and libraries but instead to select the suitable library to demonstrate the intended concept. Hence, in this section, the IFCOpenShell library was selected. This library was chosen since it is a python library, which is the programming language used in this research, and due to the uninterrupted progress of this library development, which gives the developers the opportunities to add more functionalities with time and update the developed tool as new IFC schema versions will be released.

Bearing in mind that python version python 3.7 was used. Hence the above library needs to be consistent with this version. This type of functionality will help to read the imported IFC-based BIM model in order to extract the required information and export them again in IFC format. The IFC4 schema was used in this study.

- **Information processing.** As an initial effort to test and validate the proposed data exchange method, a simple data extraction tool was developed to stress the technical feasibility of the proposed architecture. Python language was used to develop a Graphical user interface (GUI) using Tkinter, which is a standard GUI library for Python that helps to create fast GUI applications. Tkinter has several advantages, such as it provides a layered design and is compatible across all operating systems, including Windows, macOS, and Linux. Figure 22 illustrates the extraction workflow of the tool from uploading the BIM IFC-based model through the data processing to the final IFC file output. The data extraction process was divided into three parts:

  (1) Extraction as required, including extraction of a single data set or multiple data sets. This extraction will include several objects according to their types and relations entities. For instance, after the IFC file is imported and read against the IFC schema using the IFCOpenShell library, different classes, such as Structuraldataset () and Costdataset (), were created based on the previously defined requirements to extract various data sets. In another other, these data sets are used to extract a partial model from the imported model. Each of those classes included a command-line called ".write()" that is used to save the extracted data set into a new IFC file locally in the same location where the tool is stored and the name of the output file follows the extraction purpose.

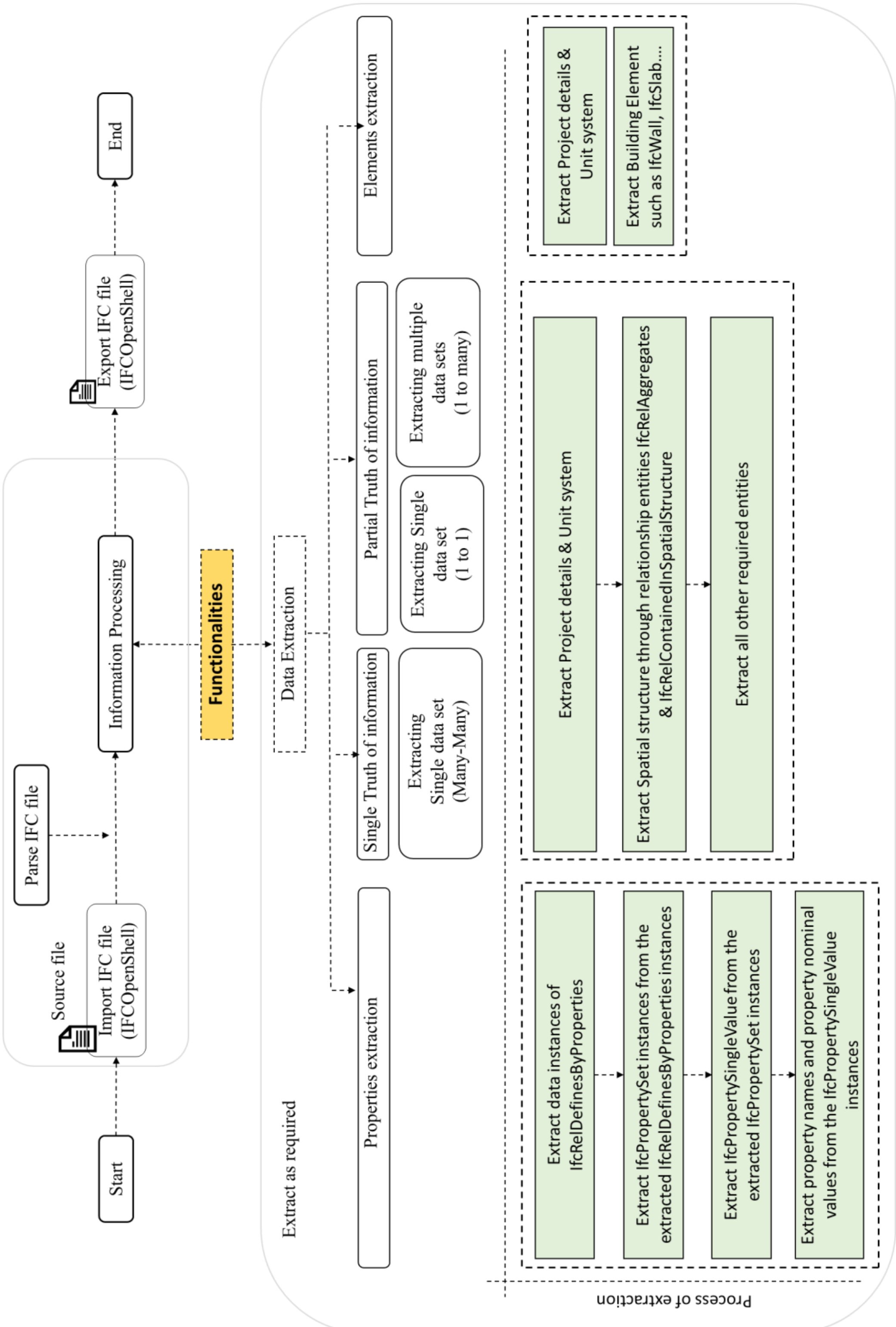


Figure 22: The workflow of the data extraction tool.

In order to achieve partial model extraction, the relationship entity was included as an approach to extract data sets. Using this approach will extract a group of instances since these entities are defined and used in the IFC schema to specify the relations across different entities in an IFC model. For instance, to extract data related to Project details & Spatial structure, the IFC relation entities IfcRelAggregates and IfcRelContainedInSpatialStructure were used. IfcRelAggregates entity can help to get the building structure, including the building location (IfcSite), the building (IfcBuilding), the building storey (IfcBuildingStorey) and spaces (IfcSpace). Figure 23 illustrates an example of using the extraction tool to extract IfcRelAggregates and its related data instances. The data instances with the identifier (ID) numbers #1033759, #1033763 and #1033767 are instances of IfcRelAggregates. Many instances of an IFC entity can appear in one model file. These instances include the data instances of IfcProject (#108) and IfcSite (#1030319), IfcBuilding (#123) and IfcBuildingStorey (#148 and #166). Whereas IfcRelContainedInSpatialStructure entity can help to get the building storey (IfcBuildingStorey) and their Building elements. However, to avoid missing any of these entities if the IFC file was modelled differently, each has also been added separately. This approach was used only in the partial model extraction function and cannot be used for element extraction since it will extract data that might not be needed.

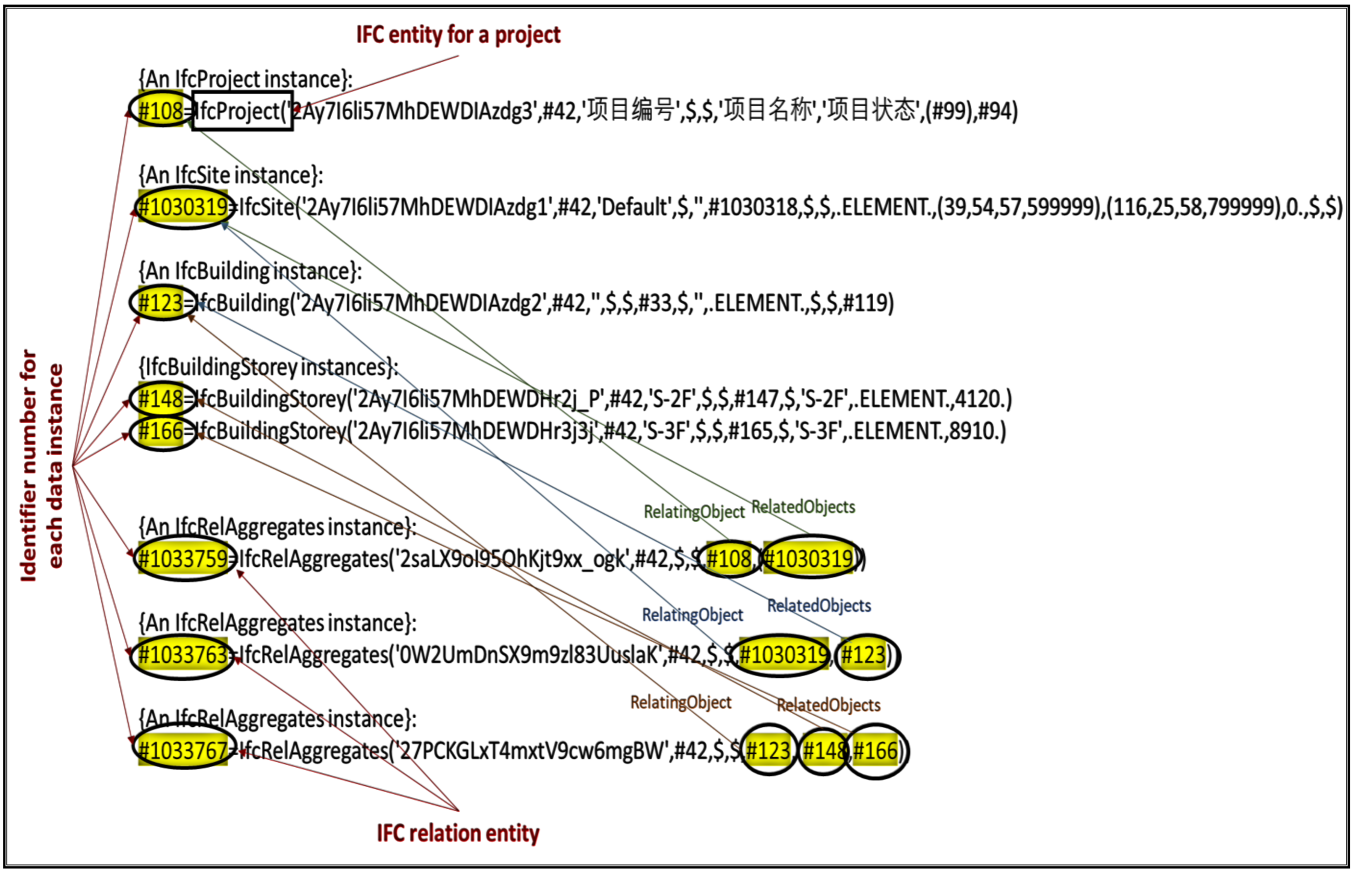


Figure 23: An example for using the developed tool to extract IfcRelAggregates entities.

Furthermore, the building members are represented as IfcColumn, IfcSlab and IfcWall. However, this representation is different in the structural analysis model. The building members are represented as IfcStructuralCurveMember for linear elements such as Column and IfcStructuralSurfaceMember for surface elements such as Wall and Slab. Consequently, the requirements of the structural model and the structural analysis model were combined to form one model, which was based on Ramaji and Memari's work (2016), where they stated that following such a process of extraction will help in linking the physical design model to its analytical model. Hence, an entity such as IfcStructuralAnalysisModel, which is used to assemble all information needed to represent a structural analysis model (BuildingSMART, 2021a), was included. It comprises a structural element, structural connection, structural activities, and others. The relationship entity IfcRelAssignsToGroup is used to relate the structural analysis model to the structural member's entity (IfcStructuralMember). IfcStructuralMember is a supertype entity for IfcStructuralCurveMember (Linear elements) and IfcStructuralSurfaceMember (planar elements) (Wang, Yang and Zhang, 2015).

The structural member comprises structural loads, boundary conditions, load cases (IfcStructuralLoadCase) and load combinations (IfcStructuralLoadGroup). For instance, the load is represented by an "IfcStructuralLoadPlanarForce" entity as part of an "IfcStructuralPlanarAction" instance. IfcStructuralPlanarAction entity is used to represent the load, and the IfcRelConnectsStructuralActivity relationship entity is used to relate structural elements such as IfcStructuralSurfaceMember to an activity (Wang, Yang and Zhang, 2015). Structural connection is represented by IfcStructuralConnection, divided into a point (IfcStructuralPointConnection), line and face connections(Wang, Yang and Zhang, 2015). The restraints of joints or release of frame elements are obtained from IfcBoundaryNodeCondition. These data are specific data required in the structural domain and can be only provided by structural analysis tools. However, the extraction process depends mainly on the input model. For instance, if the model has only the structural model information, only the structural model will be extracted, and the same will be applied to the structural analysis model.

(2) Element extraction includes extraction of a certain type of building elements such as beams, columns, or slabs from the entire IFC file. This sub-functionality can follow two ways of extraction either extracting only the specific instance or eliminating unnecessary instances. The latter was used. Although it is possible to

extract an element directly, extracting an element immediately without its spatial structure can result in inconsistency in the IFC file hierarchy. Consequently, in this extraction step, only the required element with its spatial structure and project details are retained.

(3) Property set extraction, which is used to extract properties data sets and quantity sets that can be used as input data for other tools and design needs. Building elements are linked to their properties following two paths: direct link using IfcRelDefinesByProperties and indirect link using IfcRelDefinesByType (Zhang and El-gohary, 2020; BuildingSMART, 2021a). The proposed method uses IfcElement to find the IfcRelDefinesByProperties, Figure 24. The property sets are found using the path: IfcRelDefinesByProperties => IfcPropertySet. This iteration will reach IfcPropertySingleValue, which includes name, description, nominal value, and unit of elements. The relationship entity can be extracted using the inverse (INV) attribute IsdefinedBy. Whereas the related entity can be extracted using the direct attribute RelatingPropertyDefinition. On the other hand, the algorithm uses the same workflow to extract quantities. However, to extract quantities, quantities are found using the path: IfcRelDefinesByProperties => IfcElementQuantity, which includes area, volume, and length, Figure 25.

```python
#  iterating through IFC entities
def process(self):
    products = self.ifc_file.by_type('IfcElement')
    for product in products:
        if product.IsDefinedBy:
            definitions = product.IsDefinedBy
            for definition in definitions:
                if 'IfcRelDefinesByProperties' == definition.is_a():
                    property_definition = definition.RelatingPropertyDefinition
                    if 'IfcPropertySet' == property_definition.is_a():
                        for property in property_definition.HasProperties:
                            if 'IfcPropertySingleValue' == property.is_a():
                                print('Name: ' + str(property.Name))
                                self.ifc_file2.add(property)
                                w = Label(top, text='Name: ' + str(property.Name))
                                w.pack(side=LEFT,anchor=NE,padx=5,pady=10)
                                print('Description: ' + str(property.Description))
                                w = Label(top, text='Description: ' + str(property.Name))
                                w.pack(side=LEFT,anchor=NE,padx=5,pady=10)
                                print('Nominal Value: ' + str(property.NominalValue.wrappedValue))
                                w = Label(top, text='NominalValue: ' + str(property.NominalValue.wrappedValue))
                                w.pack(side=LEFT,anchor=NE,padx=5,pady=10)
                                print('Unit: ' + str(property.Unit))
                                w = Label(top, text='Unit: ' + str(property.Unit))
                                w.pack(side=LEFT,anchor=NE,padx=5,pady=10)
```

Figure 24: A sample of property set extraction using IfcOpenShell.

```python
#  iterating through IFC entities
def process(self):
    products = self.ifc_file.by_type('IfcElement')
    for product in products:
        if product.IsDefinedBy:
            definitions = product.IsDefinedBy
            for definition in definitions:
                if 'IfcRelDefinesByProperties' == definition.is_a():
                    property_definition = definition.RelatingPropertyDefinition
                    if 'IfcElementQuantity' == property_definition.is_a():
                        for quantity in property_definition.Quantities:
                            if 'IfcQuantityArea' == quantity.is_a():
                                print('Area value: ' + str(quantity.AreaValue))
                                w = Label(top, text='Area value: ' + str(quantity.AreaValue))
                                w.pack()
                            if 'IfcQuantityVolume' == quantity.is_a():
                                print('Volume value: ' + str(quantity.VolumeValue))
                                w = Label(top, text='Volume value: ' + str(quantity.VolumeValue))
                                w.pack()
                            if 'IfcQuantityLength' == quantity.is_a():
                                print('Length value: ' + str(quantity.LengthValue))
                                w = Label(top, text='Length value: ' + str(quantity.LengthValue))
                                w.pack()
```

Figure 25: A sample of quantity extraction using IfcOpenShell.

This approach was used as an initial step to test the developed tool. Further validating will be conducted in Chapter 6. The developed data exchange method plays a significant role in simplifying the process of obtaining related data from a BIM model where users can use subsets, critical information or specific elements for their design or analysis instead of working with a complex model. The tool developed uses the IFC format, which is a neutral format. Consequently, it can provide a flexible input that can merge easily with other technologies and data. In order to visualise the extracted data sets, the BIM vision tool, which is a free IFC model viewer, was used in this section to check the consistency, Figure 26. More content related to the tool developed is shown in appendix C.

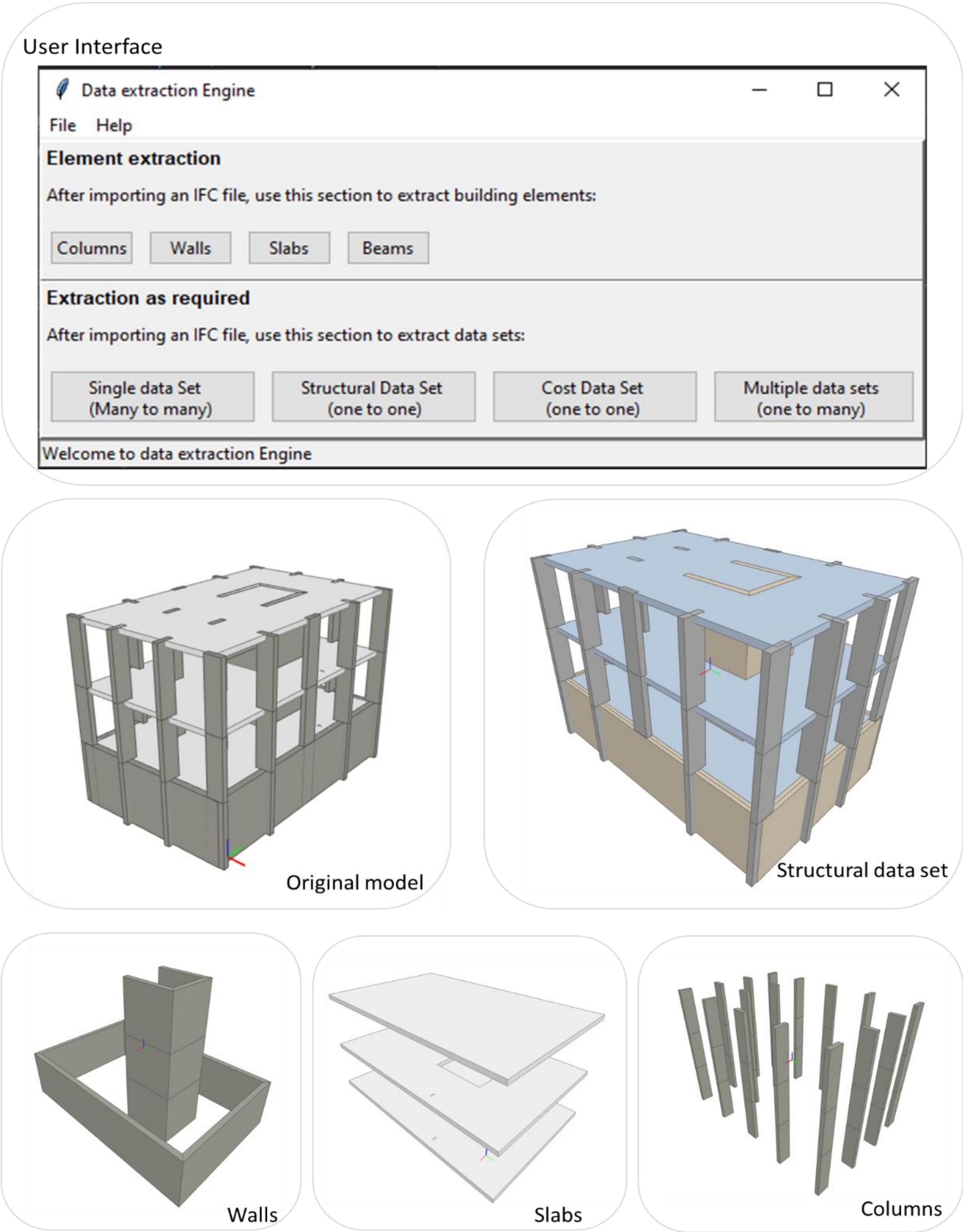


Figure 26: GUI interface based on Tkinter and the extracted information.

5.2. Multi-objective knowledge base development

5.2.1. System architecture

The ontology development process can be processed in two phases: Firstly, ontology developers need to identify and create a comprehensive review of all concepts related to the targeted problem. In this research, this was shown in Chapter 2, where calculation methods related to design criteria (load capacity and fire resistance), material prices, embodied carbon values and labours cost were discussed and obtained. For instance, Embodied CO2 values related to sustainability and cost values were collected from the ICE database and the available literature. The above values were also reviewed against the available articles in these areas.

Secondly, the data and knowledge collected need to be converted into a knowledge base. Therefore, Protégé is employed as the ontology management system to model, edit and work with the ontology, Figure 27. The collected data will be stored in the knowledge base in the form of classes, properties, relations, instances, and values. The knowledge base will be further supported by new facts based on the stored information using SWRL rules. SWRL rules can be accessed using SWRL Tab, which is offered by the protégé. Moreover, end users can query information using the SQWRL tab to execute SQWRL queries. Several queries have been added based on the stored information and the developed SWRL rules. Furthermore, the protégé provides some reasoning plugins such as Pellet, which will be utilised as a reasoning tool to check the consistency of the developed knowledge base before and after applying the data acquisition method proposed in this research.

The output of the ontology can be delivered in OWL/XML or RDF format. The main advantages that exist in the OWL language are better expressiveness and the ability to emerge new ontologies to the current existing application (Wang, Boukamp and Elghamrawy, 2011). Two ways can be used to provide semantics using OWL language, direct semantics and RDF-based semantics (Venugopal, Charles M. Eastman and Teizer, 2015c). RDF, which is described as a “standard developed for describing any web resource in a machine-understandable way while exchanging information” (Venugopal, Charles M. Eastman and Teizer, 2015c), is adopted as the modelling language in OWL to define the knowledge base. The main advantage of RDF over basic XML is that RDF can define a basic set of terms with specifiable meaning in the form of schema that can be used to model a hierarchy of classes and properties with elements and restrictions (Venugopal, Charles M. Eastman and Teizer, 2015c). Consequently, the Rdflib library, which is based on RDF, will be used later in this study to align the developed ontology with the data exchange method.

Using such a format can offer a flexible way of operating where the proposed ontology can be updated or modified in the future.

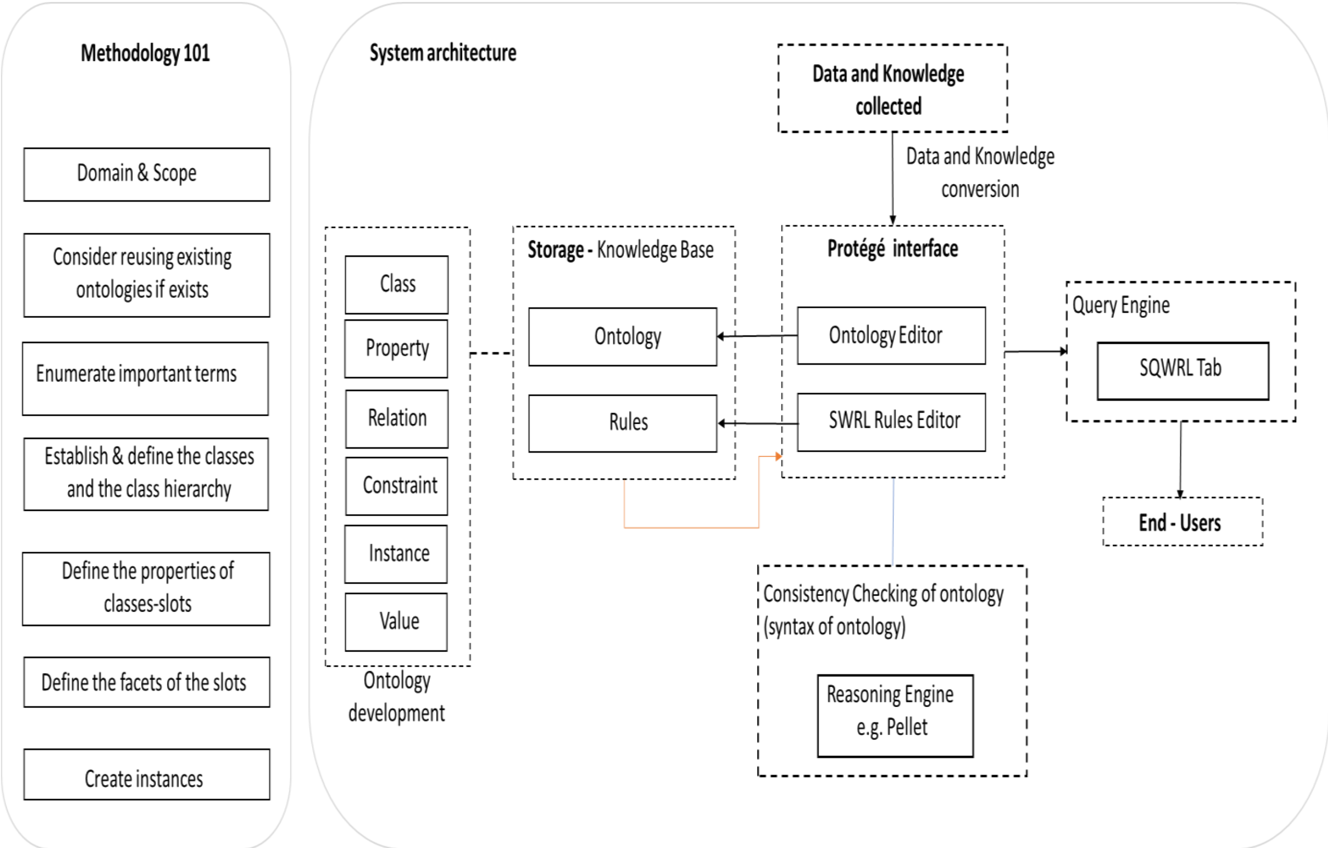


Figure 27: The ontology development process

5.2.2. Ontology development and automatic data acquisition method

Several methodologies have been proposed to construct ontologies, such as "Uschold and King", "Methodology of Grüninger and Fox", "METHONTOLOGY", "Common KADS and KACTUS", "Ontology 101", and others (Iqbal et al., 2013). Each of those ontologies has advantages and disadvantages. After reviewing several methodologies, ontology development 101 (Noy and McGuinness, 2001) was selected for this study since it provides a simple guideline on implementing an ontology that can be easily understood by non-experienced developers and users. Consequently, by choosing this method, several steps need to be followed, Figure 27:

- **Domain & Scope.** This step plays a critical part in the ontology development phase since ontology development can cover broad topics and domains. Therefore, to define the domain and scope of the ontology required in this research, some competency questions can be stated, which can help limit the scope of the ontology and provide clarity about the final outcome. Some of these questions with the corresponding answers are as follows:

Q: What will this ontology be used for?

A: To provide a holistic decision-making knowledge base that can assist engineers who lack knowledge that is associated with sustainability and cost in comparing different design choices while considering design conditions based on the existing data in an IFC-based BIM model in order to develop an ideal design in the early design stage.

Q: Who will use the ontology?

A: The ontology will be used by engineers who lack knowledge that is associated with sustainability and cost perspectives.

Q: What is the implementation language?

A: Ontology will be implemented in RDF/ OWL format.

- **Consider reusing existing ontologies.** Using existing ontology can help in developing ontologies that are compatible with each other. Moreover, by reusing existing ontologies, the developed ontology can be semantically valid by default since those ontologies were developed by other experts in those fields and have already been validated through various approaches. Only the additional concepts need further validation. Several ontologies have been developed for various domains, Table 11. However, they were developed separately. Some of these ontologies can contribute to the proposed ontology. For instance, the structure of the proposed ontology follows the semantic structure of the IfcOWL ontology, Figure 28, which was created based on the IFC schema.

Table 11: Existing ontologies in the AEC industry

Name	Purpose
IfcOWL	It makes the entire IFC EXPRESS schema available in OWL format.
Free Class OWL ontology (FC)	Construction material

QUDT (Quantities, Units, Dimensions, and Types Ontology)	Quantities and units of measurement
Free Class OWL ontology (FC)	Construction material
Good Relations ontology (GR)	Provides a conceptual model for general concepts such
OWL-S ontology	Provides computer-interpretable descriptions of web services
Organization Ontology	Organization ontology is intended to provide a generic, reusable core ontology that can be extended or specialized for use in particular situations
BOT (Building Topology Ontology)	Minimal ontology for describing the core topological concepts of a building.
Teddy	Describes the building elements for structural analysis software

- **Enumerate important terms and define the class hierarchy.** As discussed earlier, the key concepts were obtained from different sources in the literature and databases. All the collected concepts and data related to the proposed ontology need to be structured in a hierarchy structure where all the relations and attributes need to be indicated. Chong Johnson and Chong Johnson Lim (2015) classified the Ontology development process into two approaches: top-down and bottom-up. In this research, a top-down approach is utilised to create a taxonomy, which is used to organise and connect concepts. Bearing in mind that it should be rationally defined.

  According to Corcho and Fernandez-Lopez (2003), a basic ontology need to include several elements: (1) classes, which act as a blueprint that reflects the concepts considered; (2) object properties, which represent the relations between concepts; (3) Data-type properties, which represent the relations between concepts and attributes and can be characterised as a string, float, Boolean or integer. A Unified Modelling Language (UML) was used to model all those concepts, object properties, and data type properties, Figure 29. The general classes, including their sub-classes, were added first. For instance, main classes such as “characteristics” is broken down into “Material definition” class that was divided further into “Material” class that included different concrete types such as HPC and NSC. Other main super-classes include “Element”, “Labor”, “Location”, ”Project”, ”ResourceSupplier”, etc. Further information about those main classes and their subclasses are shown in Figure 30.

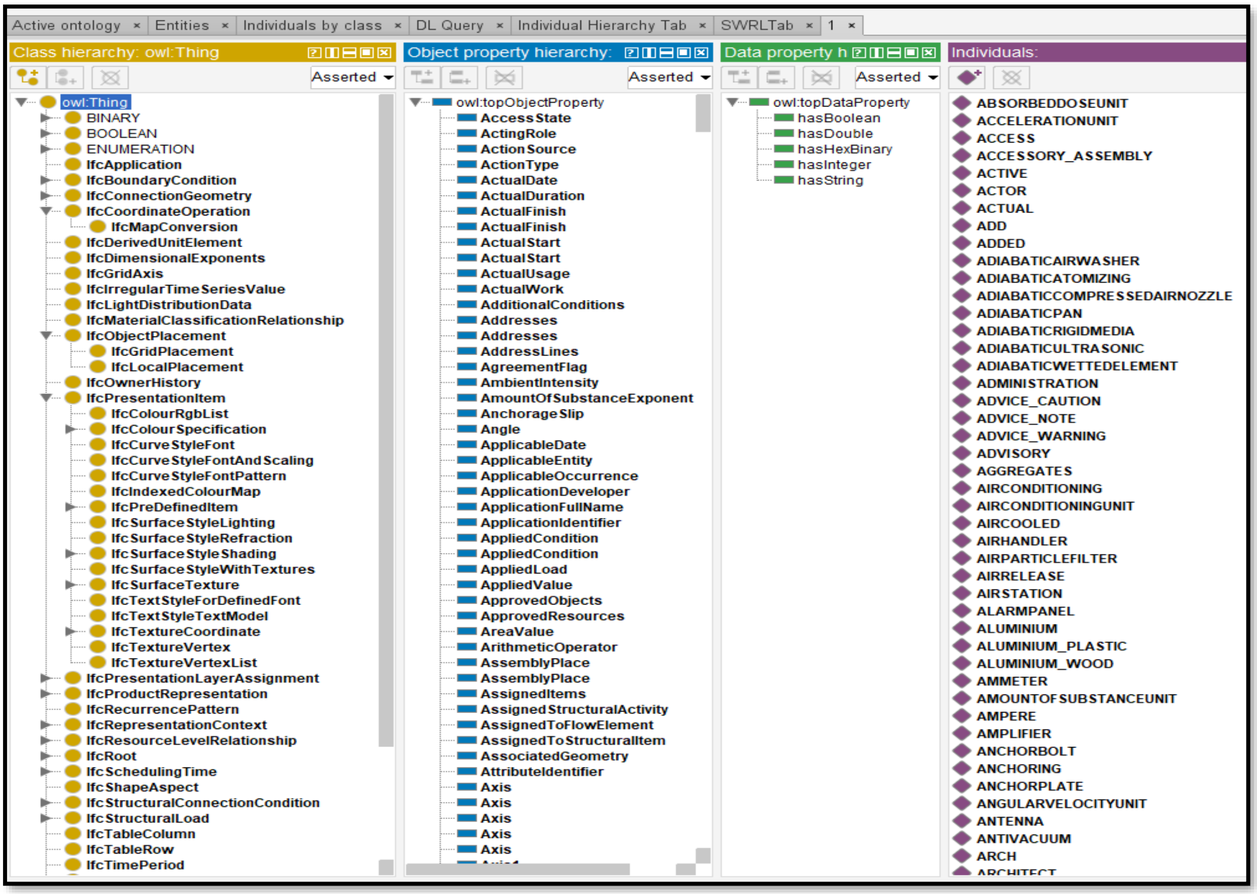


Figure 28: The IfcOWL representation in Protégé.

- **Define the properties of classes.** After defining the class hierarchy of all concepts, new properties should be attached for further development. Three different types of properties can be defined in an ontology: “Object property”, “Data-type property”, and “Annotation property”. “Object property” can define the relationship between two classes. For instance, “Object property” such as “isLocatedAt” can be defined by providing the first class, such as “Building”, as the domain, whereas the second class, such as “Siteinfo”, can be defined as a range. Figures 30 and 31 show the relations between various classes using “Object property” and all the defined “Object property” before applying the data acquisition method, respectively. On the other hand, “Data-type property” is used to define the attributes of instances of classes. For instance, a C25 concrete has several attributes such as compressive strength (hasfckC25), Cost (hasCostC25), density (hasDensity) and embodied CO2 (hasEmbodiedCO2eC25). All the defined Data-type properties are shown in Figure 31. The final type of property is “Annotation property”, which helps in adding comments or explanations in text format to any class of property. In the proposed ontology, the “Object properties” and “data-type properties” were added in two steps: the first step is done before applying the data acquisition method proposed, which includes static information. These properties include the information collected from manuals, papers, databases, and standards. The second step is done through the automatic data acquisition method proposed in this research using the Rdflib library and IfcOpenShell, where additional properties were added based on the data collected from the IFC file.

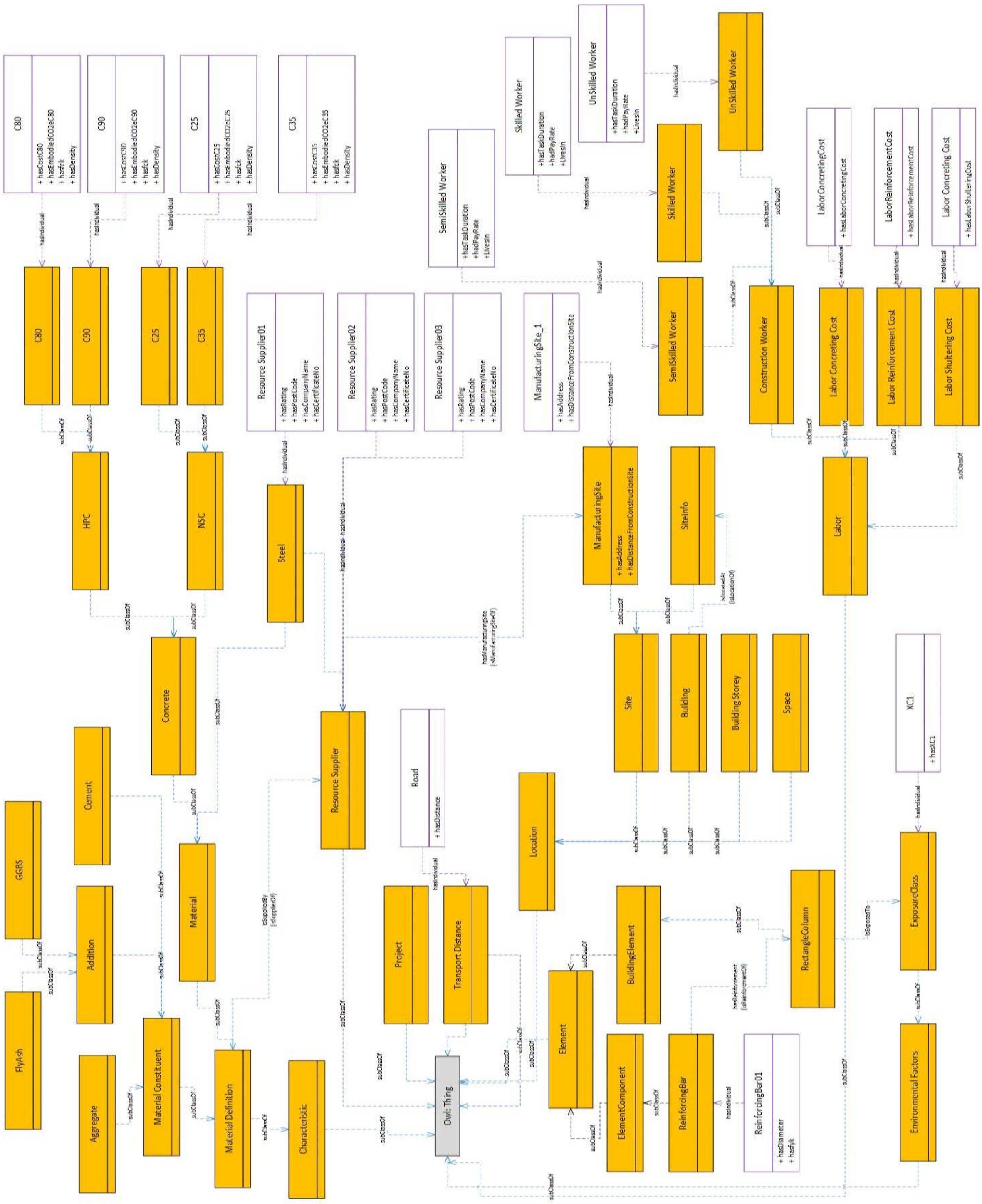


Figure 29: UML class diagram of the proposed ontology before applying the data acquisition method.

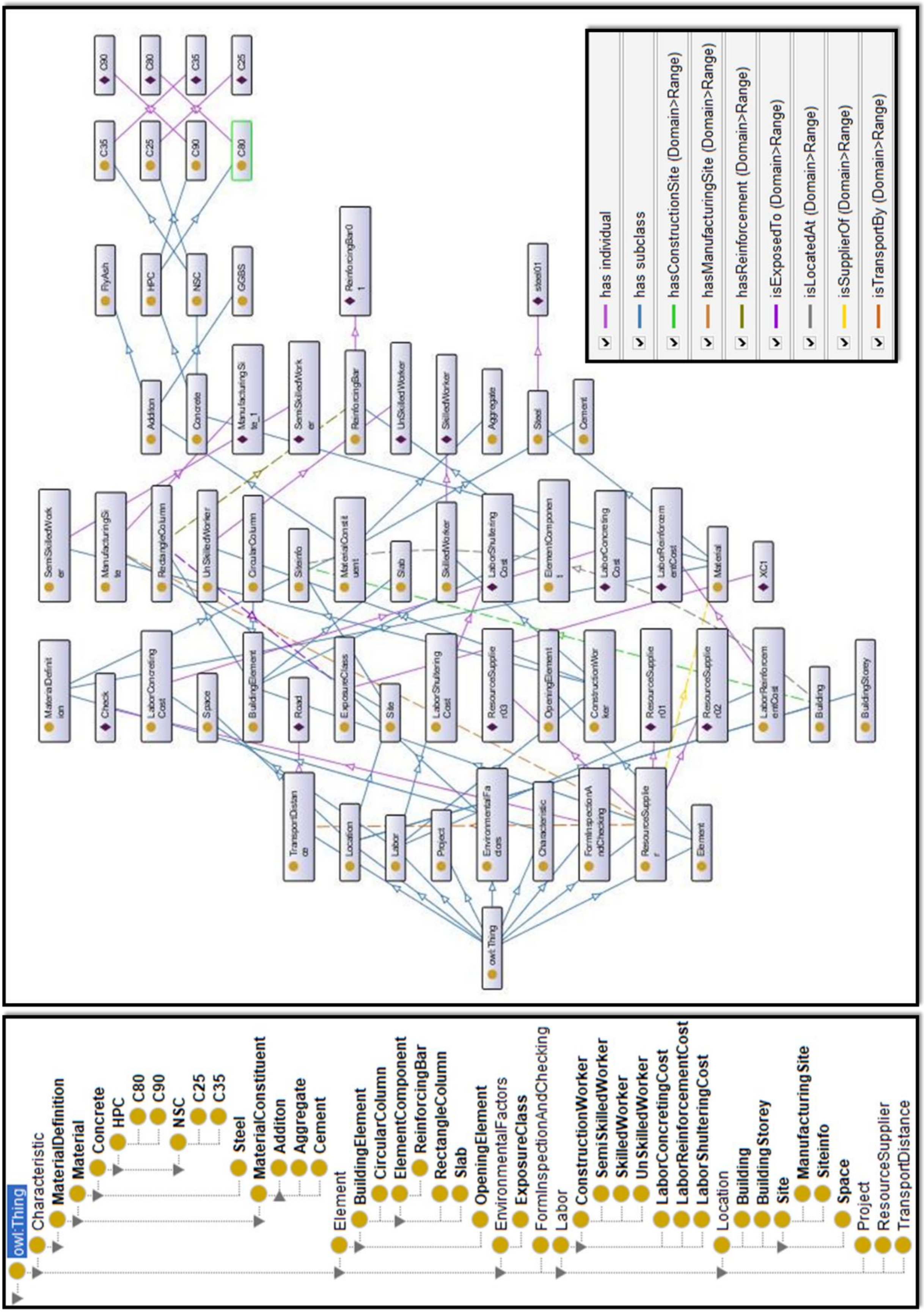


Figure 30: A High-level overview of the proposed ontology before applying the data acquisition method.

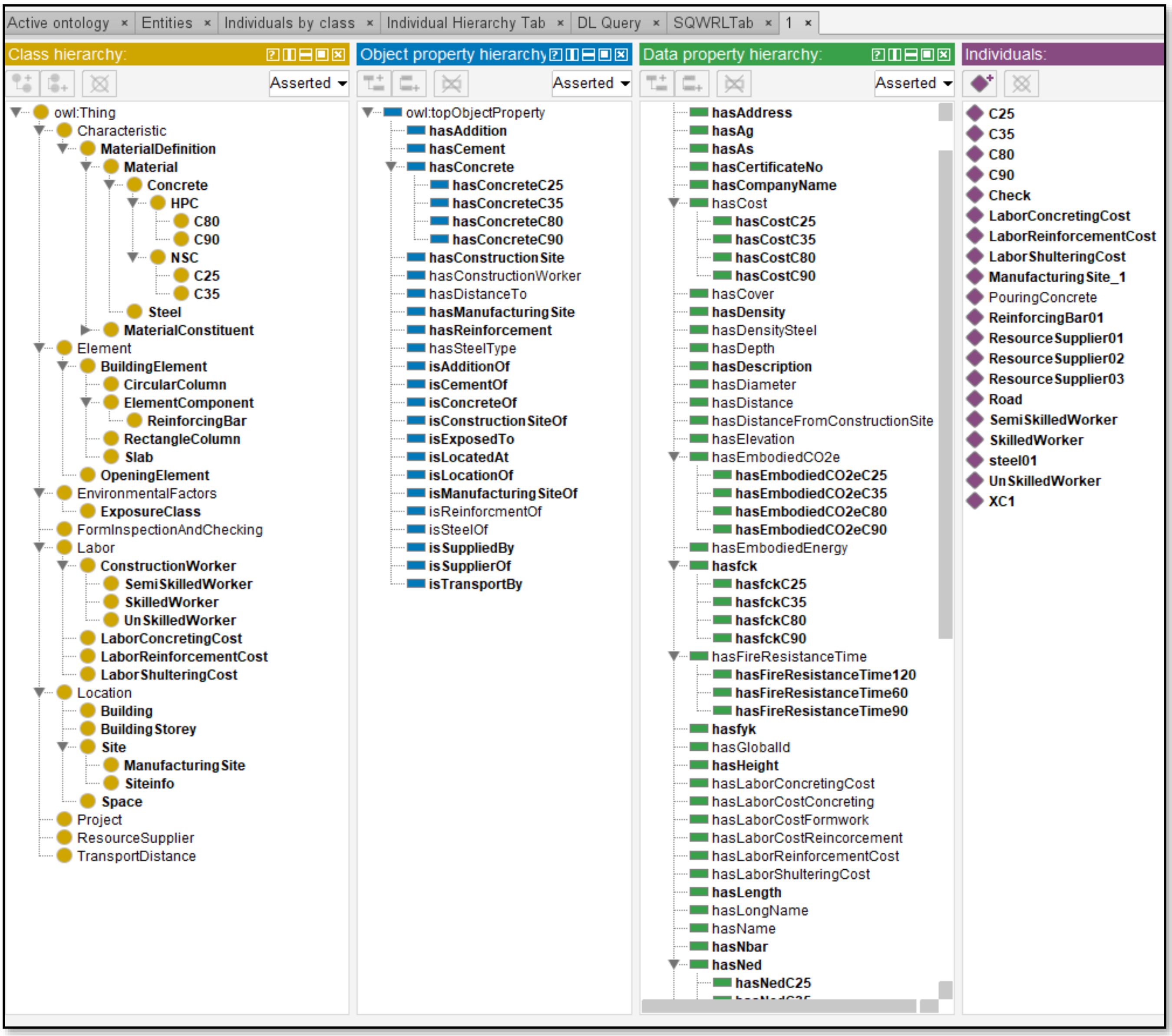


Figure 31: The proposed ontology representation in Protégé before applying the data acquisition method.

- **Define the facets and create instances.** Most of the ontologies were produced manually to work as a knowledge base. However, instances play a significant role in any ontology. In work done by Zhang et al. (2018), where they developed a multi-objective knowledge base, instances were added manually one by one, which can be time-consuming and require users to know how to add the needed instances and values to generate some results. This way of working provides a static knowledge base, which does not give the engineers the chance to link automatically to the actual project information to review several design choices. Zhang et al. (2013) mentioned that processing the IFC file against the ontology is the most crucial step in developing a tool. Consequently, in the proposed ontology, the instances were classified as:

(1) "Static knowledge", where the information is collected from manuals, papers, databases, and standards. The values collected are fixed and might change after a period of time. Consequently, it is necessary to revise and update the ontology to improve its accuracy. These instances were added by selecting the specific class and then adding all the related instances, including its object and data-type properties. For example, in Figure 32, the instance "C25" should have the "hasfckC25" property, which provides the strength of the concrete material used.

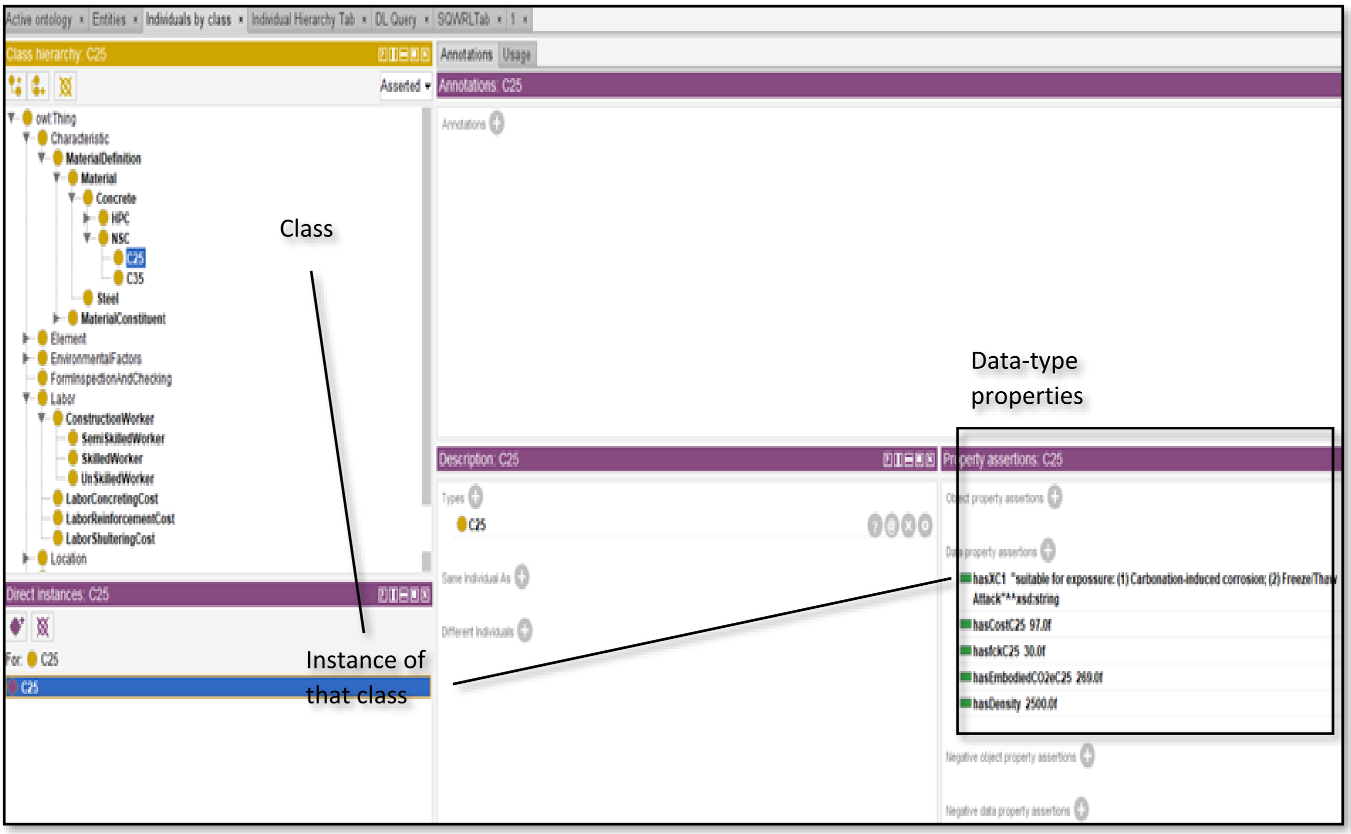


Figure 32: An example of an instance and its corresponding data-type properties before applying the data acquisition method.

(2) "Dynamic knowledge", which represents the instances and information collected from an IFC file. These instances and information will change according to the given project (IFC file). Therefore, they will be added automatically by taking advantage of the "extraction as required" method and the data acquisition method developed in this research, which will eliminate the manual input and help connect with project data by utilising the data that exists in the IFC-based BIM model.

Ontology alignment is normally performed by comparing and mapping the concepts based on the ontology structures and the linguistic similarity between concepts (Cheng et al., 2008). The developed automatic data acquisition method, as shown in appendix D, is based on the Rdflib and IfcOpenShell libraries. By using the Rdflib library, the developed method takes the original ontology, including its classes, properties, instances, and rules, as an input and then generates a new file in RDF format. In the meantime, IfcOpenShell is utilised to go through the IFC structure to extract the required values from the imported IFC file and assign them to the correct Uniform Resource Identifier (URI) using the Rdflib library. For instance, the "RectangleColumn" class in the original ontology, Figure 33, showed no instances before applying the data acquisition method. By iterating through the "IfcColumn" entity using IfcOpenShell, as shown in Figure 34, several instances were added under the "RectangleColumn" class, Figure 35. These instances were extracted and aligned through the IFC entity type "IfcRectangleProfileDef" and by using the URI below:

RectangleColumn =
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-ontology-48#RectangleColumn")

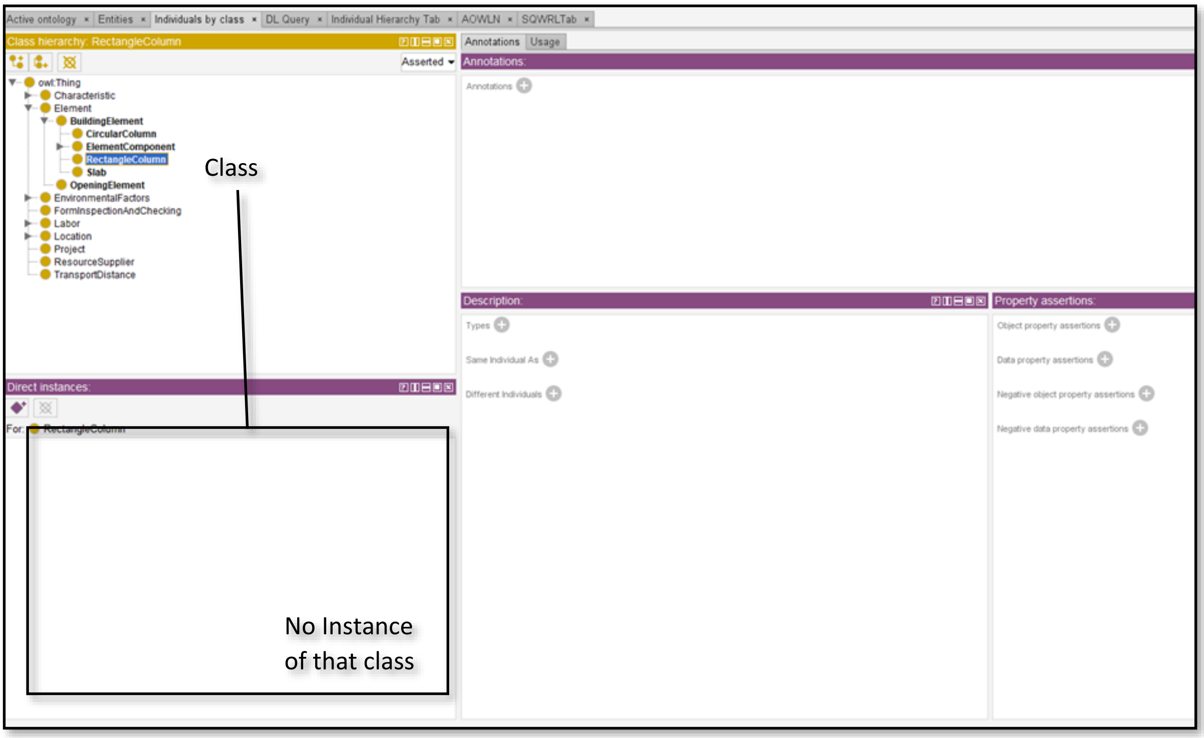


Figure 33: Instances of a rectangular column class in Protégé before applying the data acquisition method.

```
IfcColumn = ifc_file.by_type('IfcColumn')
for IfcProductDefinitionShape in IfcColumn:
    if IfcProductDefinitionShape.Representation:
        IfcProductDefinitionShape = IfcProductDefinitionShape.Representation
        if IfcProductDefinitionShape.Representations:
            IfcShapeRepresentation = IfcProductDefinitionShape.Representations[0]
            if IfcShapeRepresentation.Items:
                IfcMappedItem = IfcShapeRepresentation.Items[0]
                if IfcMappedItem.MappingSource:
                    IfcRepresentationMap = IfcMappedItem.MappingSource
                    if IfcRepresentationMap.MappedRepresentation:
                        IfcShapeRepresentation = IfcRepresentationMap.MappedRepresentation
                        if IfcShapeRepresentation.Items:
                            IfcExtrudedAreaSolid = IfcShapeRepresentation.Items[0]
                            if IfcExtrudedAreaSolid.SweptArea:
                                SweptArea = IfcExtrudedAreaSolid.SweptArea
                                BuildingElementlist.append(SweptArea)
```

Figure 34: Iterating through IfcColumn entity using IfcOpenShell.

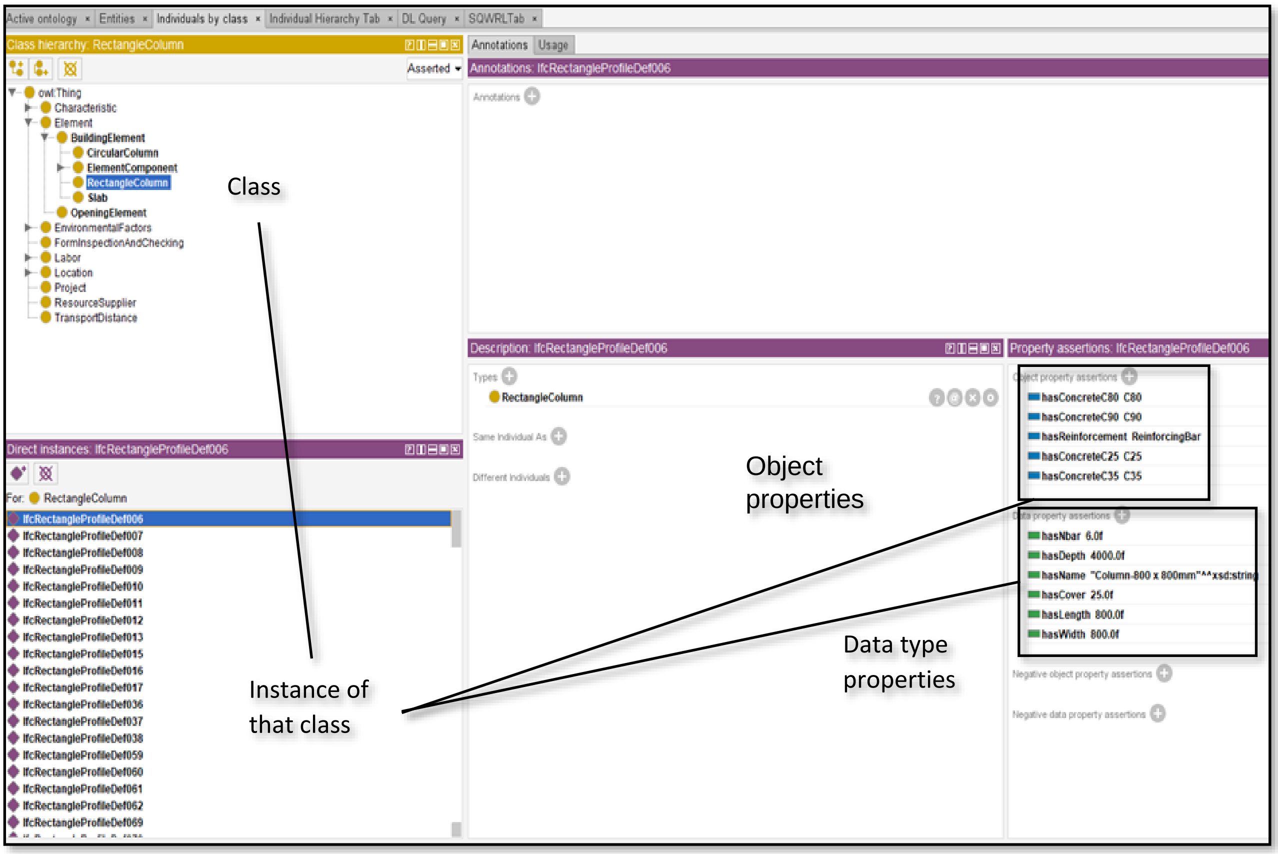


Figure 35: Instances of a rectangular column class in Protégé after applying the data acquisition method.

Furthermore, each instance consists of several "Object properties" and "Data-type properties". For example, the name of column has been assigned automatically to each "IfcRectangleProfileDef" instance using the "hasName" data-type property and by extracting the profile name information from the IFC file, Figures 35 and 36. Other data-type properties values such as length (hasLength) and width (hasWidth) were also added following the same process. Bearing in mind that all the calculation is made with a depth, which represents the column's height, equal to 4m. This information was aligned by using the URIs and the triples below:

```
g.add((bEI, URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-ontology-48#hasName"), Literal(list['ProfileName'], datatype=XSD.string)))

g.add((bEI, URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-ontology-48#hasWidth"), Literal(list['YDim'], datatype=XSD.float)))

g.add((bEI, URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-ontology-48#hasLength"), Literal(list['XDim'], datatype=XSD.float)))
```

"Object properties" related to the materials used, such as "hasConcrete25", were also automatically added to each "IfcRectangleProfileDef" instance. Further information about the data acquisition method can be found in appendix D. Figure 37 shows the proposed ontology after applying the data acquisition method.

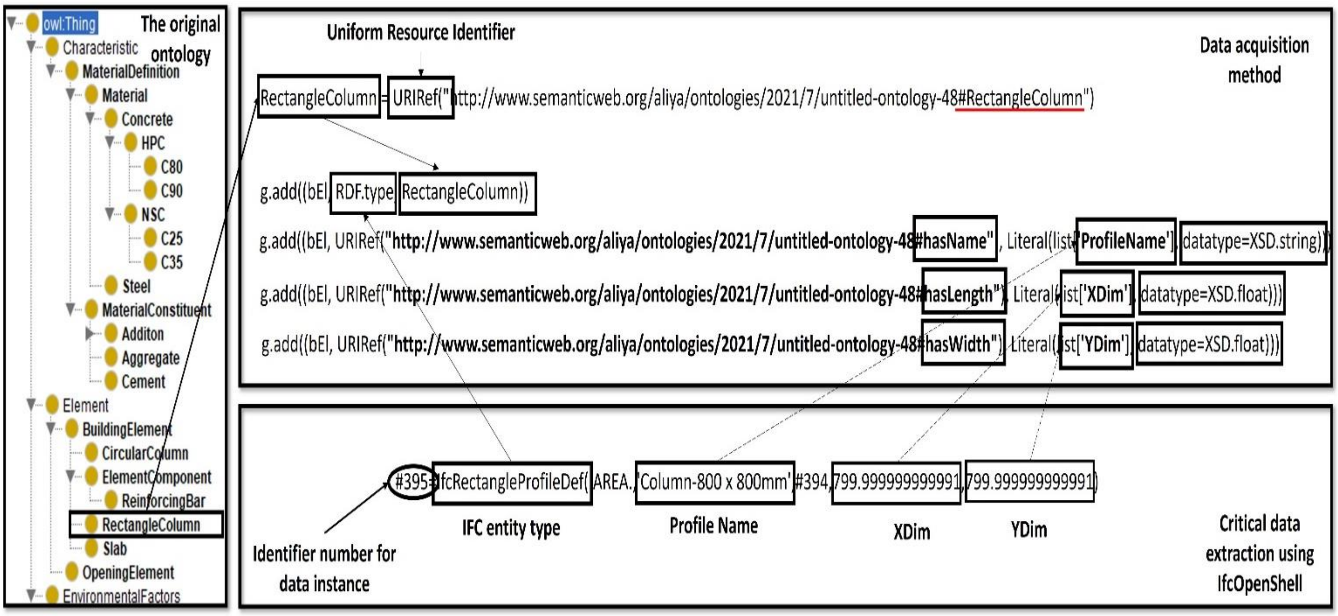


Figure 36: An example for using the automatic data acquisition method.

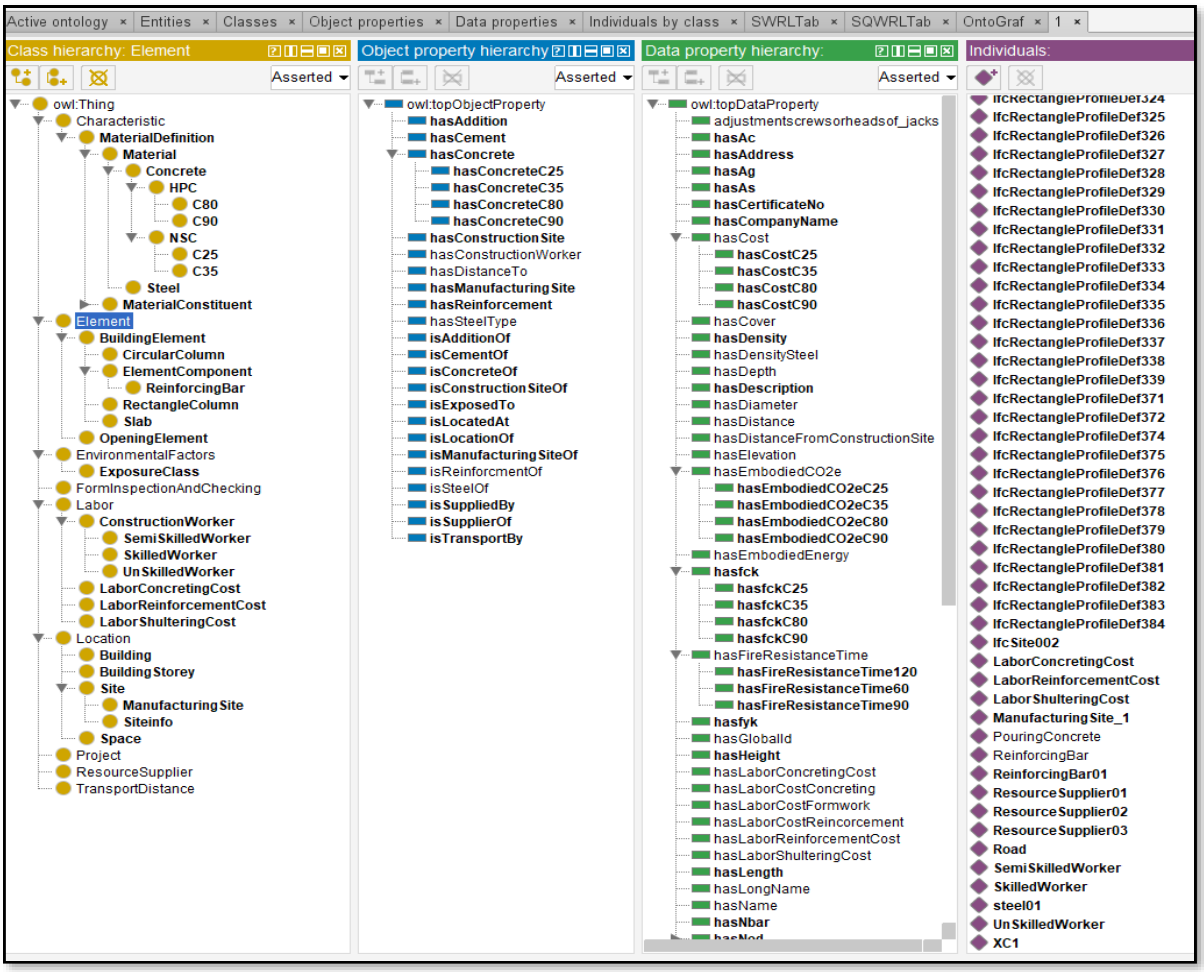


Figure 37: The proposed ontology representation in Protégé before applying the data acquisition method.

- **Defining SWRL rules and SQWRL queries.** The reasoning of SWRL rules can help in generating new facts based on the existing information in an IFC file, especially if that sort of information is not included in the exported BIM model. Several rules were defined in the ontology proposed, appendix E. Those rules were built based on the analysis and requirements defined in Chapter 2 and Chapter 4. The complexity of those rules varies from rules that consider only one aspect, which represents a single decision, to rules that consider several aspects and conditions together to provide a multi-objective knowledge base.

The SWRL provides several methods such as class atom, individual property atom, and data valued property atom. A detailed discussion about each atom can be found in Ren, Ding and Li (2019). For instance, the symbol '^' can be used to connect various classes and individuals' atoms. Whereas a mark question "? " represents the variable in each atom. The symbol "→ " can be used to connect antecedent. For example, In the case of cost estimation, the area is a critical component and can play a main part in making some decisions. To calculate the net area of a rectangular column (?ColAc), the gross area (?ColAg) needs to be subtracted from the area of steel bars (?CAs). The syntax swrlb: subtract was used to model this function. The data needed to calculate the appropriate area was extracted automatically from the IFC file based on the available dimensions for the appropriate section and the steel bars used. The data was added to the ontology as data type properties using the automatic data acquisition method developed in this research. For instance, in order to calculate the gross area (?ColAg) of a rectangular column, the width(?Cb) and length (?Ch) of the cross-section need to be multiplied. The syntax 'swrlb: multiply' is utilised to multiply these two values. The gross cross-section area, the area of the reinforcement and the net area of a rectangular column, can be represented in SWRL rules as follows:

Example of SWRL rule

Rule1-3: Net cross-section area (Ac) of a rectangular Column

RectangleColumn(?Column) ^ hasAg(?Column, ?ColAg) ^ hasAs(?Column, ?CAs) ^ swrlb:subtract(?ColAc, ?ColAg, ?CAs) -> hasAc(?Column, ?ColAc)

Rule1-1: Gross cross-section area (Ag) of rectangular Column

RectangleColumn(?Column) ^ hasWidth(?Column, ?Cb) ^ hasLength(?Column, ?Ch) ^ swrlb:multiply(?ColAg, ?Cb, ?Ch) -> hasAg(?Column, ?ColAg)

Rule1-2:Area of longitudinal reinforcement (As)

RectangleColumn(?Column) ^ hasNbar(?Column, ?CNbar) ^ ReinforcingBar(?RB) ^ hasDiameter(?RB, ?Diameter) ^ swrlb:multiply(?CAs, ?CNbar, ?Diameter, ?Diameter, 3.14, 0.25) -> hasAs(?Column, ?CAs)

Furthermore, in order for the end-user to find the needed information, SQWRL was used to query information from the developed ontology. For instance, to query the cross-section area of a rectangular Columns following the above SWRL rules, the syntax 'sqwrl: select' function is used, and the following query is built:

Example of an SQWRL query: Cross-section area of a rectangular Column

RectangleColumn(?Column) ^ hasWidth(?Column, ?Width) ^ hasLength(?Column, ?Length) ^ hasAg(?Column, ?ColAg) ^ hasAs(?Column, ?ColAs) ^ hasAc(?Column, ?ColAc) -> sqwrl:select(?Column, ?Width, ?Length, ?ColAg, ?ColAs, ?ColAc)

A list of all the developed SWRL rules and SQWRL queries can be found in Appendix E. Appendix F illustrates the names and descriptions of the atoms used in the reasoning rules and the proposed ontology.

5.3. Summary

The chapter discussed and outlined the technical development of the proposed framework. First, the data extraction method was implemented based on the proposed concepts such as MCDS and CSDS. The tool developed uses the IFC format, which is a neutral format. Hence, it can provide a flexible input that can merge easily with other technologies and data. The developed method plays a significant role in simplifying the process of obtaining related data from a BIM model where users can use only the critical information instead of working with a complex model.

Furthermore, a multi-objective knowledge base was developed to assist engineers who lack knowledge that is associated with sustainability and cost in comparing different design choices while considering design conditions based on the existing data in an IFC-based BIM model. The knowledge base was supported by a data acquisition method to fetch data automatically from an IFC file. The proposed approach is unique compared to previous research. Most of the previous ontologies require human intervention to add the necessary data, which can be time-consuming and require users to know how to add the needed instances and values. Hence, the ontology was built around two sources of data: (1) Static data, where the information is collected from manuals, papers, databases, and standards; (2) Dynamic data, which represents the information collected from an IFC file, which can be changed according to the project. Moreover, the proposed ontology was supported by SWRL rules in order to generate new facts based on the existing information, especially if that sort of information is not included in the exported BIM model. The complexity of those rules varies from rules that consider only one aspect to rules that consider several aspects and conditions together to provide a multi-objective decision., which gives the engineers a chance to compare different design choices in the early stages of a project.

Chapter 6. Framework validation and scenario-based testing

In order to eliminate inefficiency in data sharing and enhance the decision-making in the early design stage, this research implements different functionalities to represent the automatic data exchanging framework. The developed framework consists of several functionalities such as data repository, data exchange method, visualisation, access rights administration, and multi-objective knowledge base. After conducting the technical developments in chapter 5, the intended framework is proposed to be validated using scenario-based case testing to show whether it provides the necessary information through the proposed data exchange method and, at the same time, help end-users to compare different design choices related to sustainability and cost while considering design conditions based on the existing data in an IFC-based BIM model.

The first part (Section 6.1) outlines the IFC-based BIM model, which was used for this thesis's validation, and the scenario-based case testing objectives. The second part (Section 6.2) presents the scenario-based case testing, which was carried out on an airport BIM model. The scenario-based case testing was conducted to prove that the proposed framework is functional and reliable for data exchange and holistic decision-making.

6.1. IFC-based BIM model and scenario-based case testing objectives

6.1.1. IFC-based BIM model description

To avoid using a biased uncomplete BIM model, an airport BIM model located in Nanjing, a city in Jiangsu, China, which was developed by the ECADI (East China Architectural Design & Research Institute), was utilised in this thesis. The BIM model was developed based on experts' points of view using the Revit software tool. The selected BIM model, Figures 38 and 39, contains various information such as project information and site information. It comprises two building storeys that include various building elements, including their properties. It consists of substructures such as concrete beams, concrete columns, and concrete slabs; superstructures such as concrete beams, and upper floor structures such as in situ columns, upper slab sections and roof columns.

The information, which is related to sustainability and cost estimation, was not included in the selected model. The proposed framework will provide that additional information, which is out of the sender's scope, automatically using the built-in SWRL rules. Consequently, this step will be part of the framework testing and validation process. The selected BIM model was developed in the Revit software tool. Hence, in order to convert the model from RVT format, which is the native format used in Revit, to the IFC format, the embedded functionality in Revit software was utilised, and the IFC4 Design Transfer View (DTV) was selected to export the IFC file. The model showed a total of 664280 entities, of which 3415 building element entities, beams (2962), columns (424) and slabs(29).

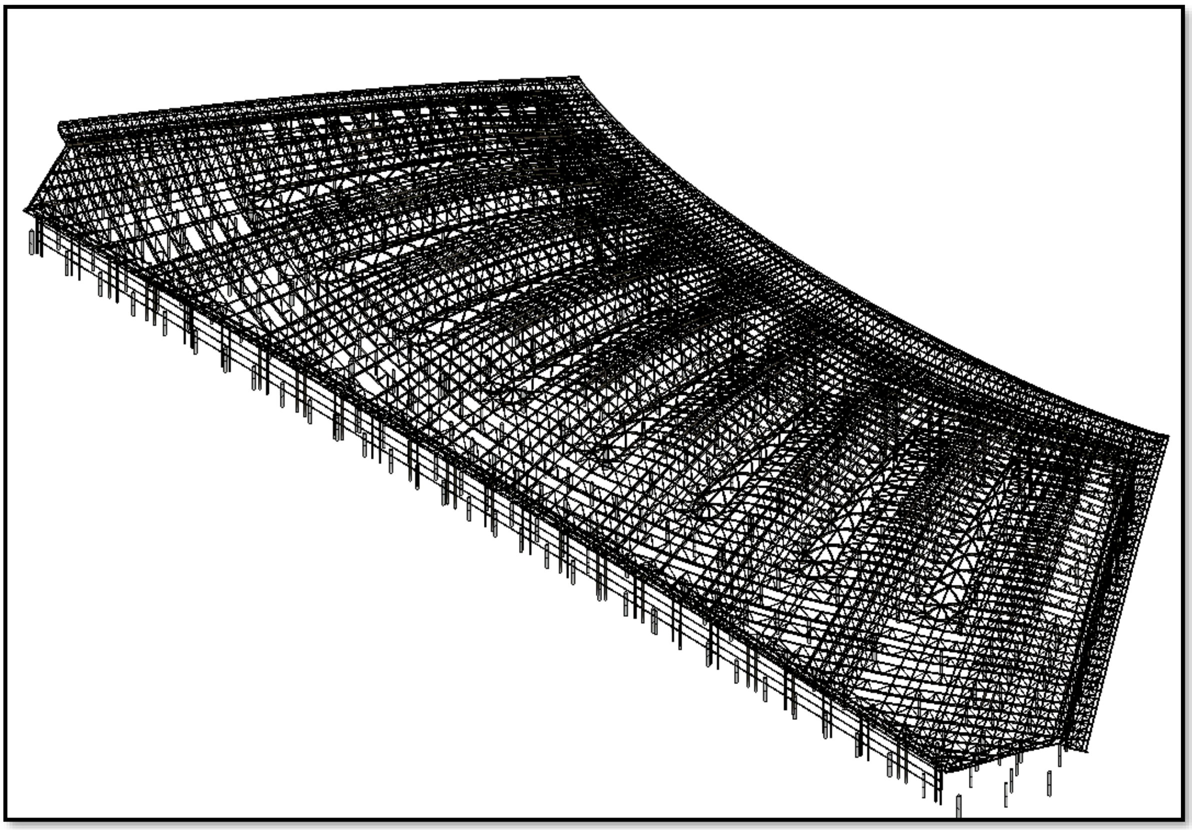


Figure 38: The airport BIM model developed using Revit software tool.

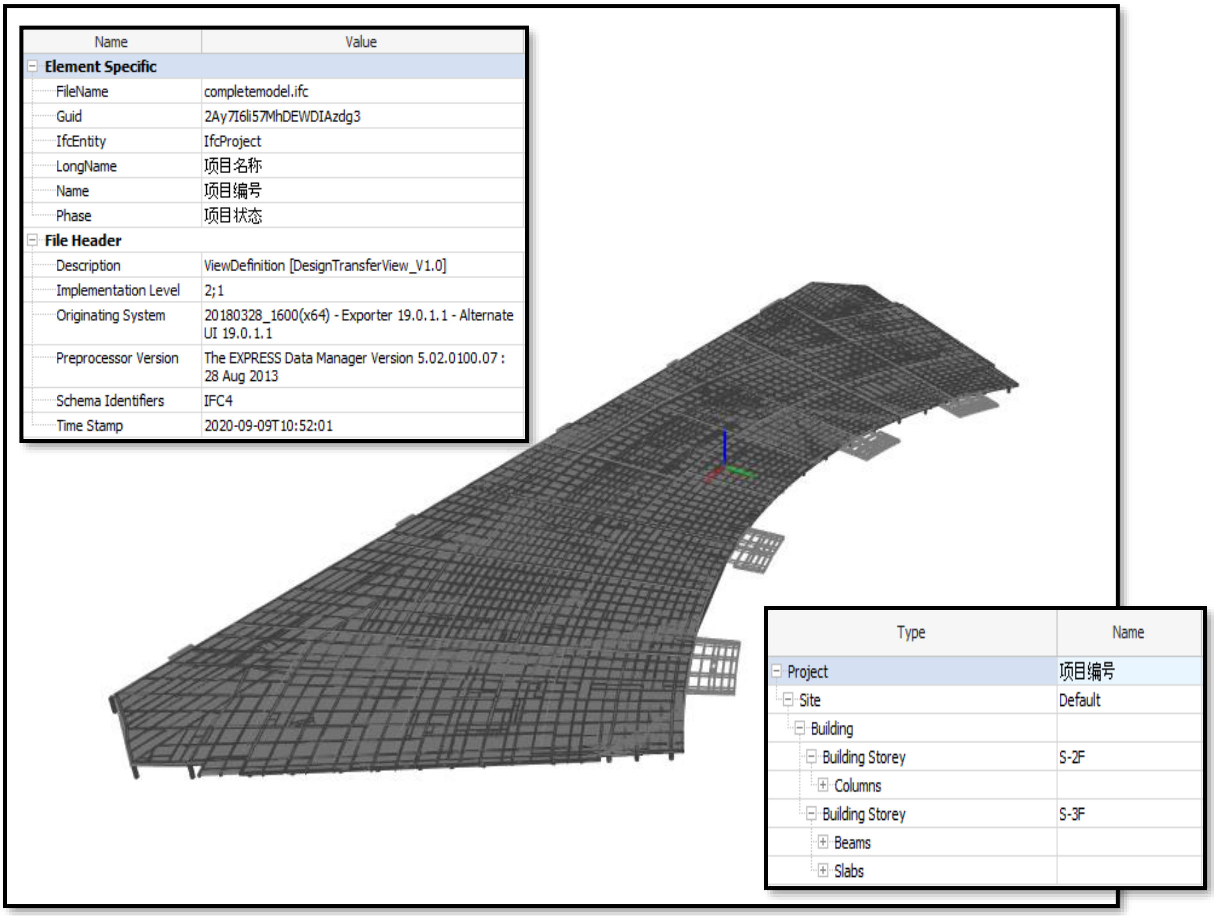


Figure 39: IFC-based Airport BIM model used for the case study.

6.1.2. The objectives of the scenario-based case testing

To assess the viability of using the proposed framework, several objectives were defined to break down the research question Q6:

(1) The ability of the framework to enhance data exchange by delivering different data sets from a complex BIM model by focussing only on the critical information is considered vital. Thus, the data exchange method was tested by applying it to the airport model to extract different data sets in order to check consistency and ensure no data loss is noted. During development, the data exchange method was constantly tested and improved. The scenario-based case testing is used to validate this assumption.

(2) The ability of the proposed knowledge base to work with other systems and formats is important. Thus, this step focuses on validating the knowledge base structure to ensure it is syntactically and semantically correct and no

inconsistencies exist. This validation plays a significant part in the knowledge base development since, in this research, a data acquisition method was proposed to align the IFC file with the ontology. Hence, ensuring that the used terms/concepts are uniform and consistent throughout the ontology development is important. Thus, this validation is done in two stages: before and after applying the data acquisition method to make sure the ontology is still semantically and syntactically correct.

(3) The efficiency of the framework to provide a multi-objective knowledge base that considers single to multi-objective decisions and its ability to work with different sources of information such as standards and databases is considered vital. Thus, the built-in SWRL rules were tested by processing several queries in order to show the reliability of the framework in providing multi-objective decisions that consider various conditions and also to test their ability to generate new facts or information that is out of the sender's scope by utilising the existing data in the IFC-based airport model.

6.2. Framework validation using an Airport IFC-BIM model

6.2.1. Data exchange method

In this section, the validation was mainly focused on the data exchange method, whereas other functionalities, except for the ontology part, were left for future work. A prototype web platform was implemented to demonstrate the data exchange method. The platform involves three authorised users: a structural engineer, a cost engineer and a client, and one unauthorised user, Figure 40. Access authority is assigned to the relevant team member. After passing the authentication and authorisation check by the server associated with the project’s user role, the participants will log into the web-based platform. People, who have no contribution to the design, will be directed to a shared dashboard, Figure 41. The Dashboard page shows all the information provided by different stakeholders. However, people cannot change or modify any content on this page. Using this approach can show only the information required for the logged-in user.

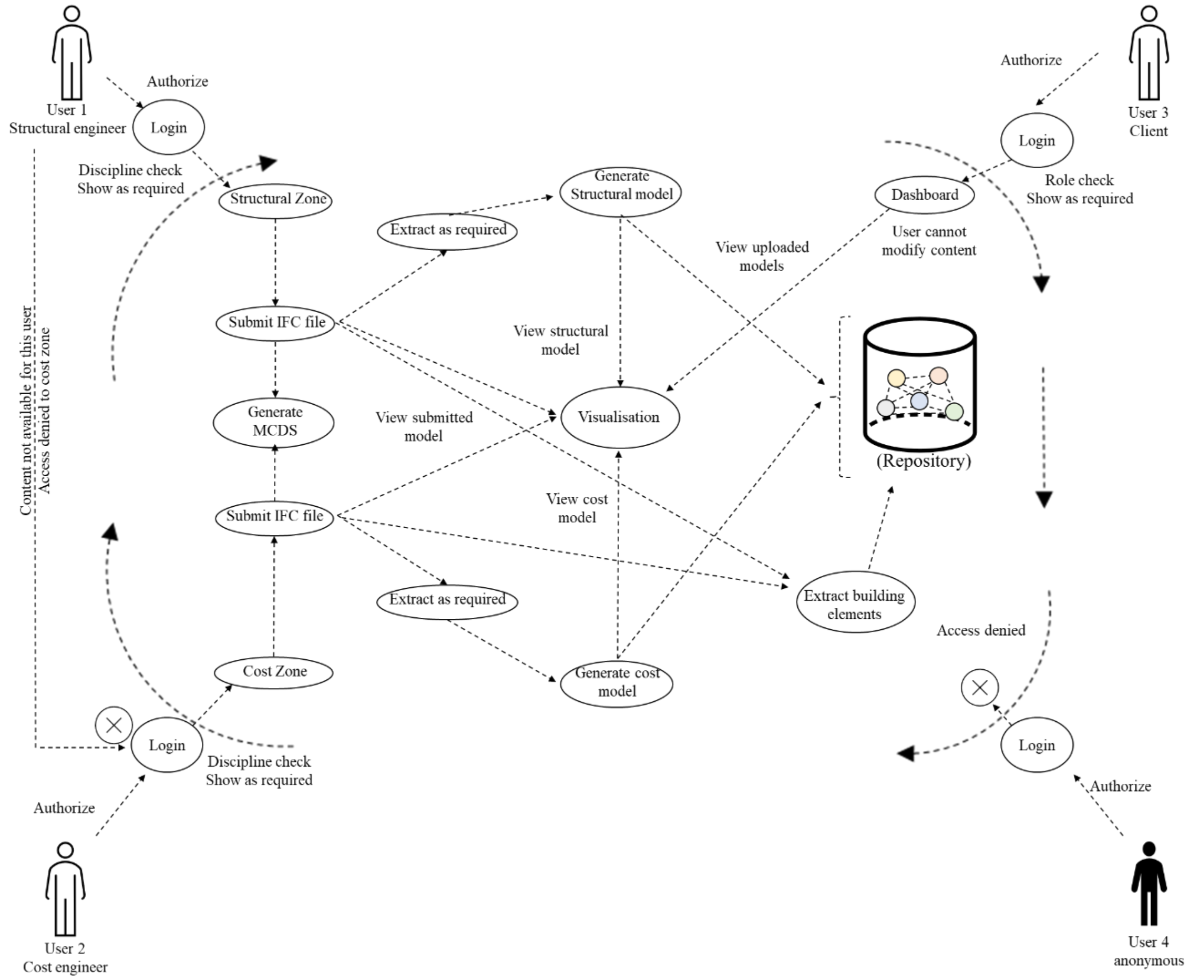


Figure 40: Use case diagram for the proposed platform.

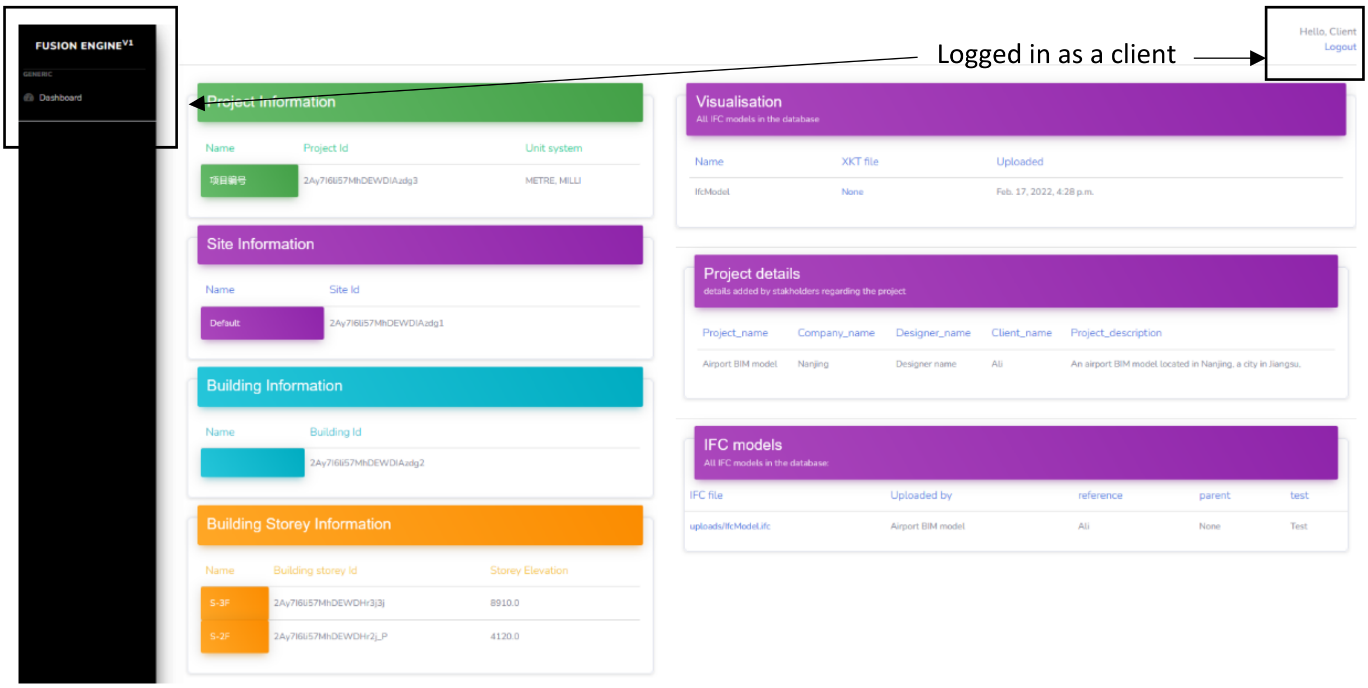


Figure 41: The dashboard page shows the information provided by different stakeholders.

Once the appropriate engineer is directed to the associated pages, the engineer can upload the IFC- based BIM model. The users will have various options: users can extract information as required, including various SCDS such as structural data set, cost data set or extract only the MCDS. The user can also extract a group of building elements. Moreover, visualisation plays an important part in the design of a building. Consequently, the platform will provide the users with a visualisation option via a web browser using Xeokit viewer, which is an open-source 3D graphics SDK built to view BIM models in the browser, Figure 42.

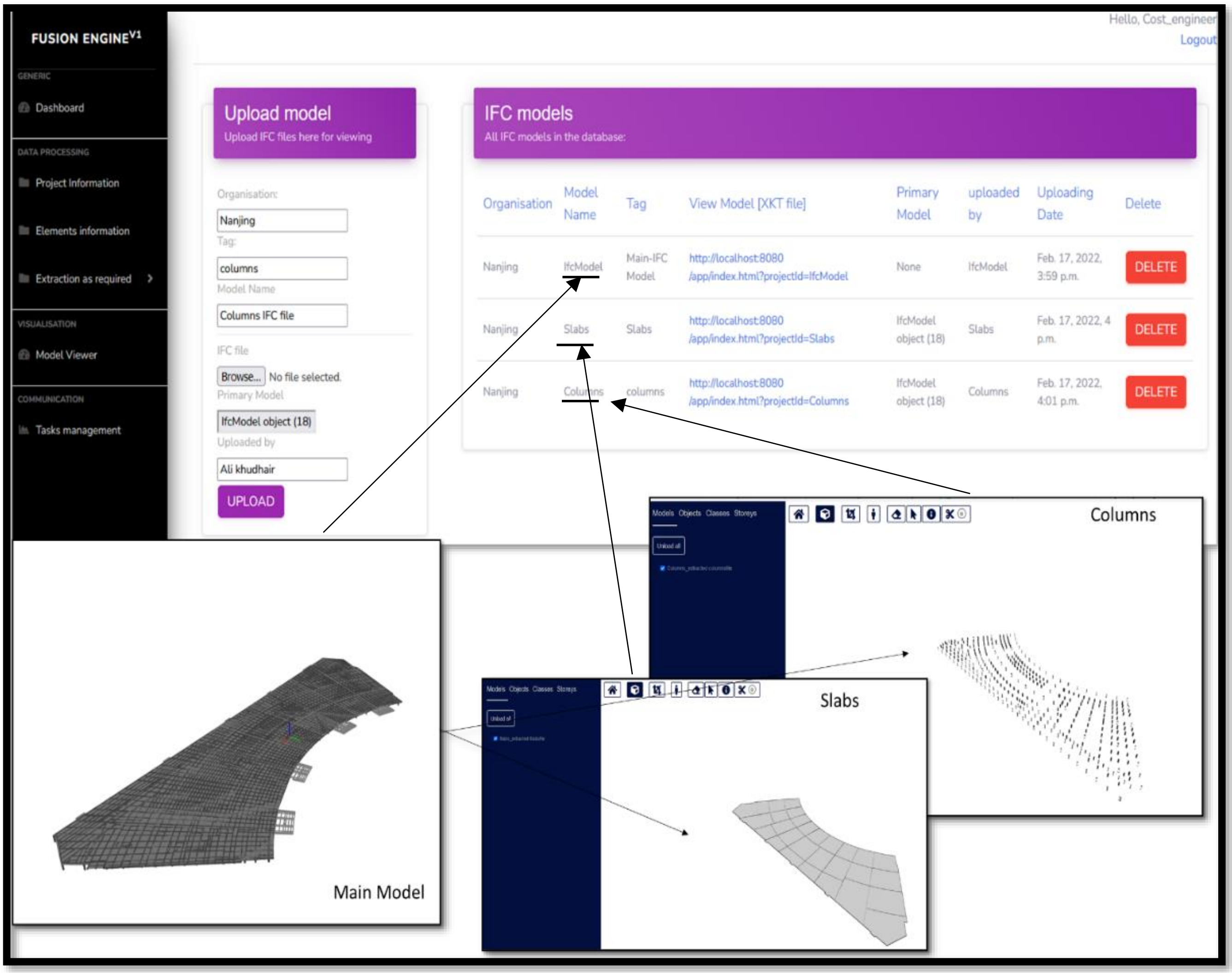


Figure 42: Visualisation of the exported Slabs and Columns using Xeokit viewer.

To easily implement the proposed data exchange method in an isolated environment that can be easily deployed on any operating system, Docker is used to host the different web-based services. Docker is an alternative to virtual machines, which are much less resource-intensive (Chung et al., 2016) since containers share a common kernel, reducing the total overheads in multi-container systems. Services can also be distributed amongst a cluster of nodes through the use of Docker Swarm, which allows for the potential development of easily deployable and high-performance cloud computing systems (Ismail et al., 2016).

This research made use of several container images in its architecture, Figure 43, such as: (1) the conda/miniconda3 (continuumio/miniconda3 - Docker Hub, 2021) image was used as a base for the Django frontend and backend, which is an open-source framework for web development based on python. (2) the SQLite database, which was used to store information supporting the Django application. (3) a dockerised version of Xeokit viewer to visualise IFC models and parse the model tree, and IFC Convert was wrapped into a microservice container using the Python Hug library. These images were deployed as a single stack using a single docker-compose file, Figure 44. This file configured the network settings automatically, allowing for requests between the services on the virtual Docker network. Ports were also exposed to the host machine, allowing connections to the web services via a browser. Volumes were bind-mounted to local directories on the host machine so that IFC and other data could be easily inspected.

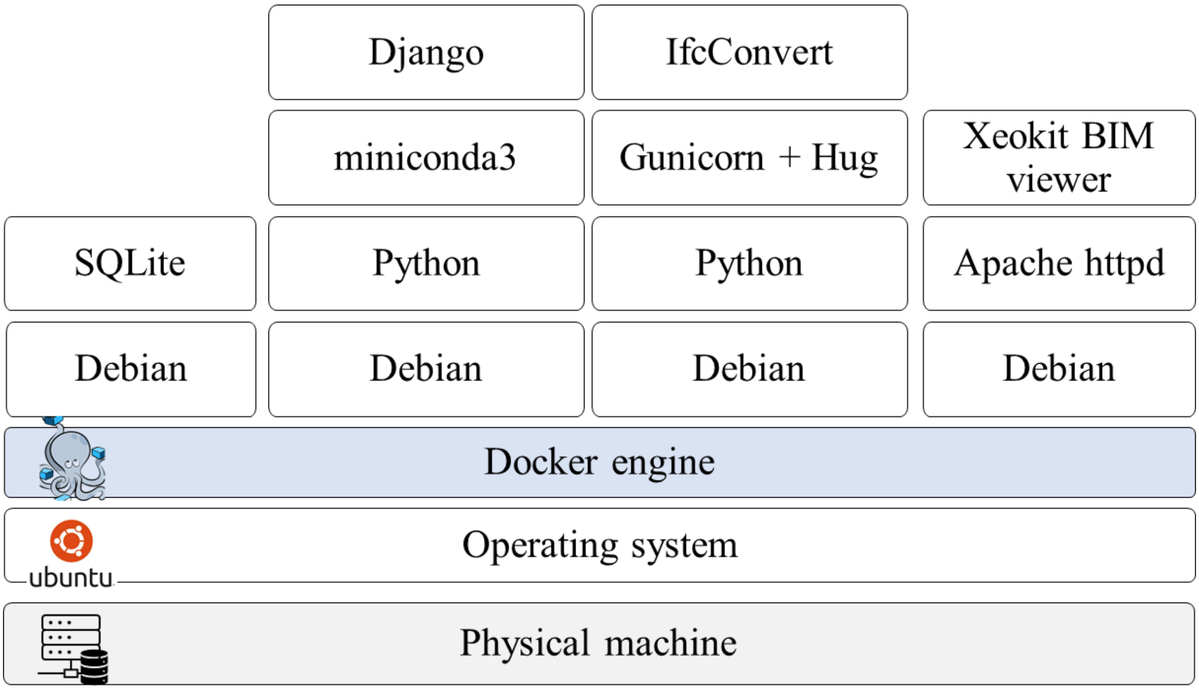


Figure 43: Stack diagram for Docker environment used in this study.

```yaml
version: '3'

services:
  web:
    build: .
    command: python manage.py runserver 0.0.0.0:8000
    volumes:
      - .:/code
    ports:
      - "8082:8000"
    depends_on:
      - db
      - ifc-converter

  xeokit-viewer:
    image: "bowerta/xeokit-bim-docker"
    ports:
      - "8080:80"
    volumes:
      - "./media/ifc/output:/usr/local/apache2/htdocs/app/data/projects"


  ifc-converter:
    image: registry.gitlab.com/cardiffbimse/smartharbour/ifc-converter-microservice:v1.1
    volumes:
      - "./media/ifc:/ifc"
    ports:
      - "8081:8080"
```

Figure 44: Docker-compose file used in this study.

Won et al. (2013) mentioned that an extraction tool is semantically successful if it can preserve the same semantic relationships before and after extraction without any data loss. Consequently, following the evaluation of the extraction tool in chapter 5, the validation process in this section will be carried out by comparing the models before and after data processing in order to test the framework's ability to deliver different data sets from a complex BIM model. The validation is done by checking whether the necessary information has been extracted correctly and without data loss. However, the tool currently supports the extraction of basic elements to elaborate on the process. Therefore, only elements such as beams, columns and slabs will be investigated.

The original model showed a total of 664280 entities, of which 3415 building element entities, beams (2962), columns (424) and slabs(29), Table 12. After uploading the IFC file to the proposed prototype of the platform, Figure 45, several files were extracted, such as the Slabs-IFC file, Beams-IFC file, Columns-IFC file, SCDS for structure design, SCDS for cost estimation and MCDS using the defined data sets. The exported files were imported into the IFC analyser for analysis. It was shown that the number of entities was reduced in the newly generated files while maintaining the same number of building elements. The number of elements extracted showed the accuracy of the data processing, and no data loss was noted. Thus, the data extraction method helped simplify the data exchange process by extracting only the necessary information, which will be further utilised in Section 6.2.3.

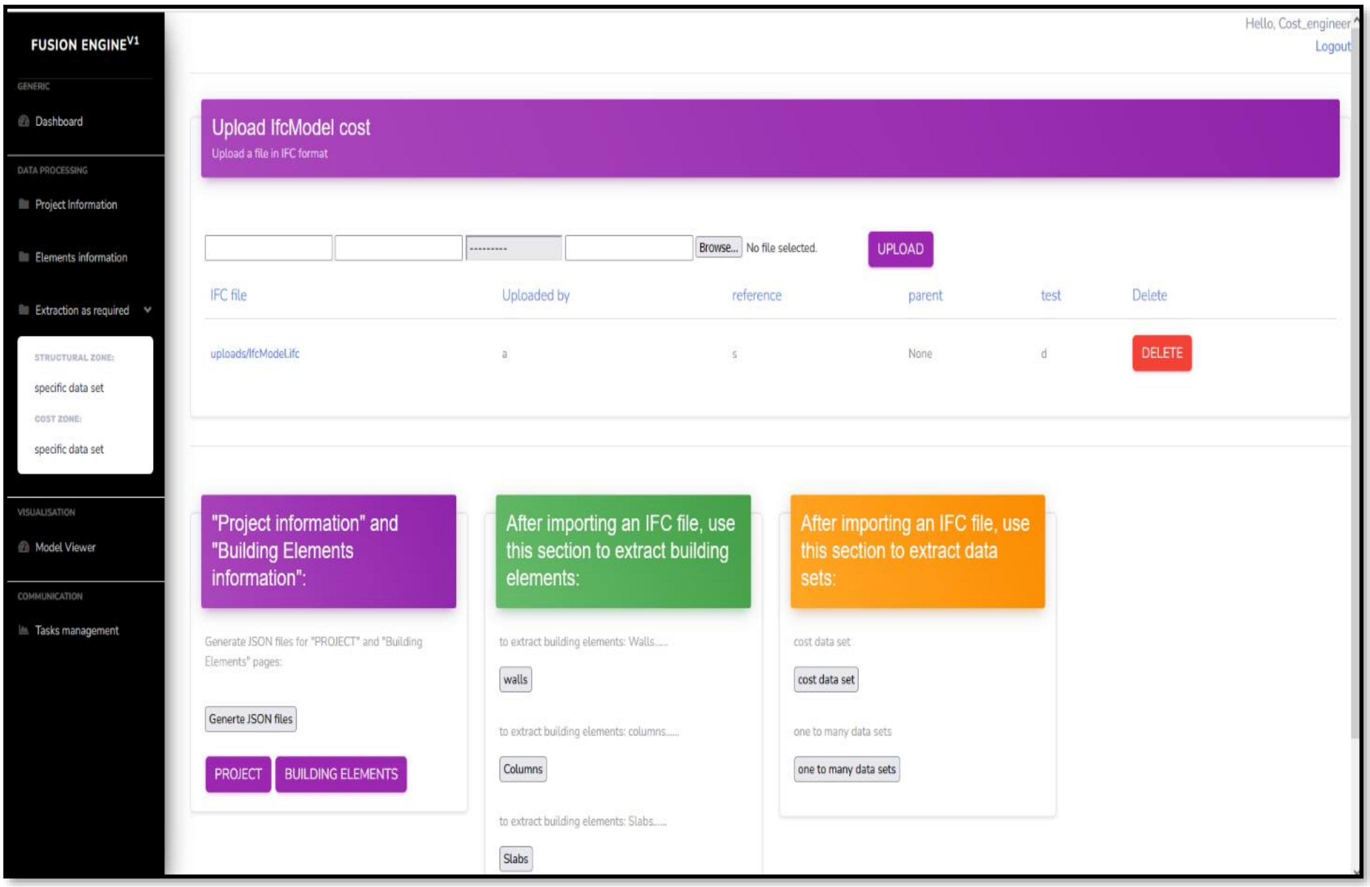


Figure 45: The proposed prototype of the platform.

Table 12: Total number of entities before and after data processing.

File Name	Total entities	Building element	Columns	Column Type	Beams	Beam Type	Slabs	Slab Type
Before extraction								
Original IFC file	664280	3415	424	389	2962	2958	29	0
After extraction								
Columns-IFC file	658331	424	424	389	0	0	0	0
Beams-IFC file	663438	2962	0	0	2962	2958	0	0
Slabs-IFC file	657547	29	0	0	0	0	29	0
SCDS-str	659722	3415	424	389	2962	2958	29	0
SCDS-cost	664279	3415	424	389	2962	2958	29	0
MCDS	640872	3415	424	0	2962	0	29	0

6.2.2. Reasoning through protégé plugins

As an initial attempt to validate the proposed ontology, the ontology reasoner pellet, a plugin function included in the protégé, was utilised to check that the developed ontology is syntactically correct. This check was carried out in two stages: (1) before applying the data acquisition method, Figure 46; (2) after aligning the IFC data with the proposed ontology, Figure 47. Based on the reasoner, the knowledge base structure was syntactically correct, and no inconsistency was recorded in both stages.

Moreover, using existing resources and ontology structures helped validate the proposed ontology semantically since using the structure and terms of previously validated ontology such as IfcOWL helped keep the consistency in the proposed ontology, which is not the case with ontologies that were developed from scratch. Ontologies that were developed from scratch require further consultations by domain experts to provide their validity. Whereas ontologies based on previous ontologies still require domain experts' opinions. However, it is more needed to adjust the proposed ontology toward the industry requirements. Furthermore, the developed SWRL rules were validated, as shown in Figure 48, using the SWRL rule Tab provided by the protégé. After the ontology was semantically and syntactically validated before and after the ontology alignment process, it was further validated through a case-based scenario in Section 6.2.3 to confirm if the ontology meets the needed requirements and to test its capability in producing multi-objective decisions.

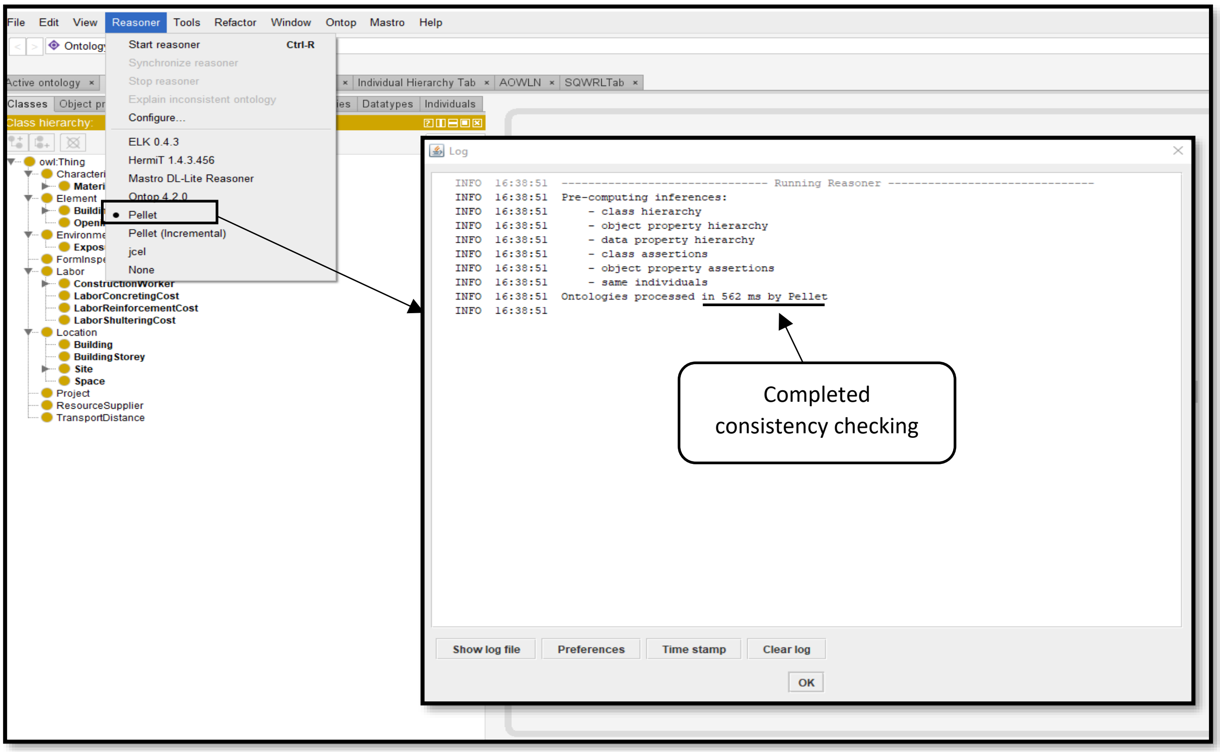


Figure 46: Reasoning process using Pellet plugin before applying the data acquisition method.

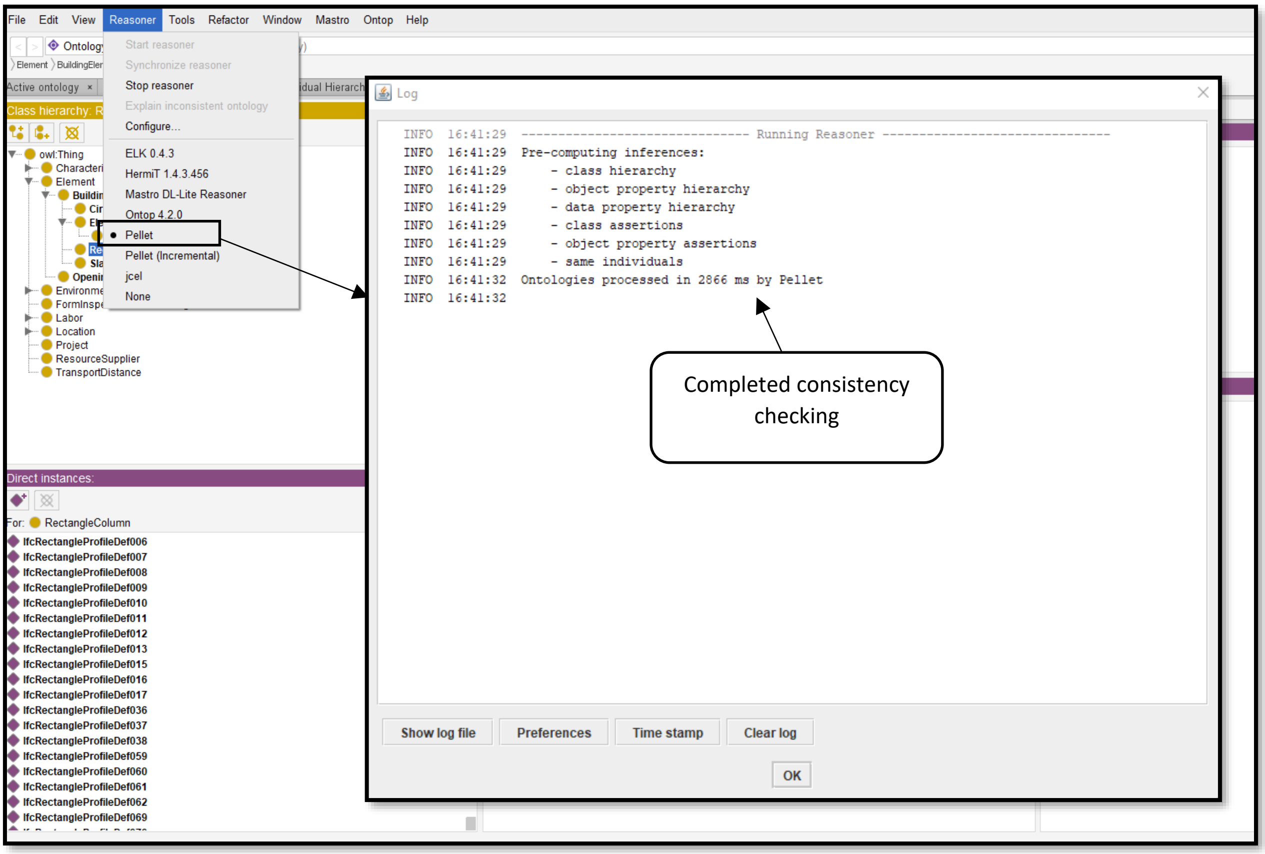


Figure 47: Reasoning process using Pellet plugin after applying the data acquisition method.

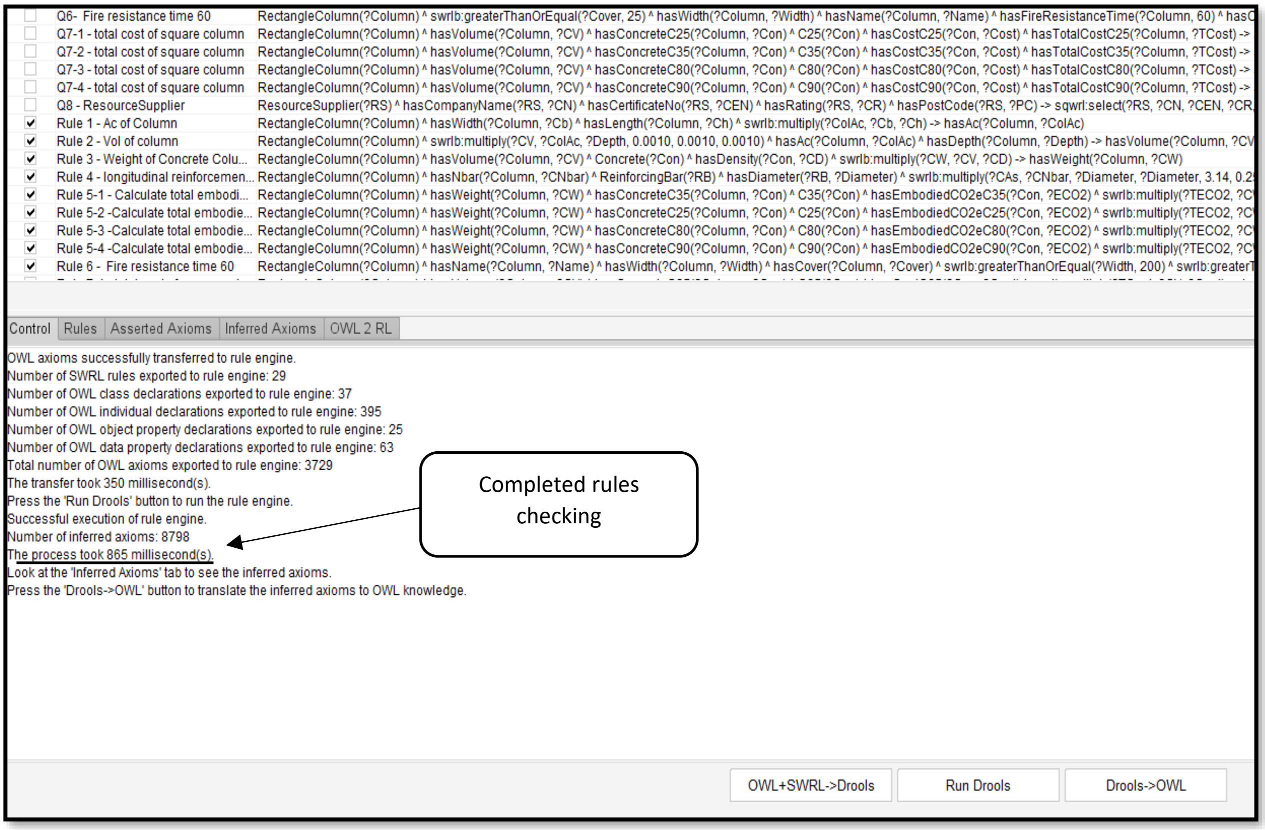


Figure 48: Validation of the SWRL rules using protege.

6.2.3. Holistic decision-making knowledge base validation

As mentioned earlier, the proposed data acquisition method can help in merging the data existing in the original knowledge base, which includes static knowledge related to design conditions, sustainability, and cost, with the data extracted from the IFC file to provide a rich multi-objective knowledge base that can help decision-makers to produce an idea design at an early stage of the design. Thus, this part of the framework validation focuses on the holistic knowledge base development and its ability to work with different sources of information in order to generate multi-objective decisions.

Due to the limited time and resources for this research, it is difficult to explore all types of building structures. Columns are one of the most critical components in building design and play a strong part in the stability of a structure. Consequently, the reasoning and query functions in the proposed ontology focused mainly on rectangular columns, and a thorough knowledge base covering various building elements will be developed in the future. The height of the concrete columns considered is equal to 4 meters. Normally in a knowledge base, the column dimensions are assumed in order to provide the required results. However, the proposed data acquisition method eliminates the manual input and provides the actual dimensions from an IFC-based BIM model, which reduces the assumption made. Thus, the ontology can compute various aspects for a building element and automatically output the query results, unlike other ontologies where building element information needs to be added manually. This approach helps to connect the real data in a given project to the decision-making knowledge base in order to investigate various design alternatives.

The data exchange method developed in this research was used to improve the efficiency of reasoning and querying since this procedure can help in reducing the processing time by only showing only the critical information instead of the entire model. Consequently, following the data sets extracted in the previous section (Section 6.2.1), the MCDS was utilised as input for the automatic data acquisition method since only the minimum required data will be utilised. Several questions were stated to test and validate the developed knowledge base. For instance:

Q-a: Using the MCDS proposed, are the exiting rectangular columns structural feasible considering the load capacity?

Q-b: By using the MCDS proposed, what is the total embodied carbon content in each rectangular column used in that IFC file for different concrete materials while considering the load capacity criteria?

Q-c: Using the MCDS proposed, what is the cost of material used for each rectangular column?

Q-d: By using the MCDS proposed, what is the cost of the total labour for each rectangular column by considering the cost of concreting, reinforcement, and formwork?

Q-e: Can the proposed framework help review the IFC-based BIM model in parallel with all the factors mentioned above while considering design conditions such as load capacity and fire resistance requirements to make decisions in the early design stage?

In order to find answers to those questions, several SWRL rules and SQWRL queries were constructed, appendix E. After running the reasoning process, several types of columns were used in this projected such as “Column-800 x 800mm”, “Column-1000 x 1000mm”, “Column-700 x 1000mm”, “Column-600 x 600mm”, “Column-800 x 1200mm”, “Column-500 x 800mm”, “Column-500 x 1000mm” and “Column-300 x 750mm”. After screening the results, only one element of each type of column was considered in order to demonstrate the results.

6.2.3.1. Checking building elements against axial load capacity

The ultimate axial load capacity of a concrete column depends on the strength of the concrete used and the strength of the reinforcement. Therefore, the ultimate axial load capacity of the extracted columns was calculated based on various types of concrete. At the same time, the yield strength of the reinforced bar was set to 415 N/mm2, and six reinforcing bars with diameter= 20 mm were used. By taking C25 as an example, the ultimate load capacity of a rectangular column with C25 can be represented in SWRL rules and SQWRL as follows:

Some of the SWRL rules and SQWRL queries used to answer Q-a

Rule2-1: Ultimate axial load of a rectangular column (C25)

RectangleColumn(?Column) ^ hasAs(?Column, ?CAs) ^ hasAc(?Column, ?ColAc) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasfckC25(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) -> hasNedC25(?Column, ?CNed)

Q2-1: Ultimate axial load of a rectangular column (C25)

RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasAs(?Column, ?CAs) ^ hasAc(?Column, ?ColAc) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasfckC25(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) ^ hasNedC25(?Column, ?CNed) -> sqwrl:select(?Column, ?Name, ?ColAc, ?CAs, ?Confck, ?SBfyk, ?CNed)

Figure 49 shows the ultimate axial load capacity for different sections of rectangular columns with various types of concrete. This ultimate axial load that was calculated using the SWRL rules proposed was compared automatically to the axial load applied to check whether the existing columns in the IFC file met the design criteria. In this research, the axial load applied to the rectangular column is assumed to be equal to 12000KN. It was shown that when C25 and C35 were used, the ultimate load capacity of some columns, such as “Column-600 x 600mm”, “Column-500 x 800mm”, and “Column-300 x 750mm” did not have enough strength to support the loads transferred. Consequently, if any of those types of concrete are selected, some sections require adjustment by either using a higher concrete strength or need to change the section dimensions in order to resist the load applied. On the other hand, it was shown that as the strength of the concrete increases, the chance of reducing the section dimensions is feasible. Thus, the proposed rules helped in providing a single objective decision.

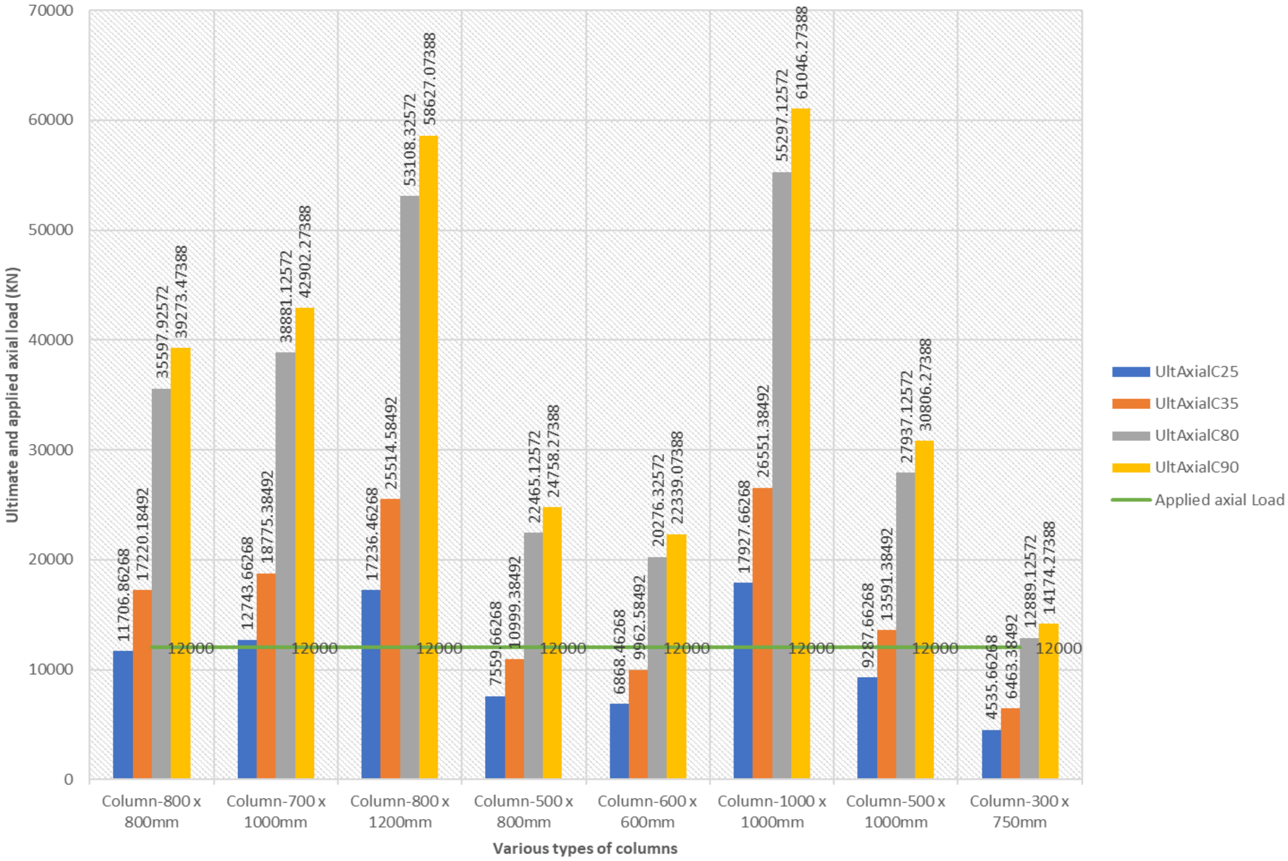


Figure 49: Ultimate load capacity for rectangular columns with various concrete strength

This applied axial load limit has been modelling in the ontology using the syntax swrlb:greaterThanOrEqual(?CNed, 12000000) and has been stored in a variable called "meetDesignCondition". This variable will be used to model other SWRL rules to make sure that the design condition (axial load capacity) is met while reviewing other aspects to move from a single to multi-objective decision-making. By taking C25 and C90 as an example, this condition can be represented in SQWRL rules and SQWRL queries as follows:

Rule3-1: ultimate axial load meet design condition (C25)

RectangleColumn(?Column) ^ hasAc(?Column, ?ColAc) ^ hasAs(?Column, ?CAs) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasfckC25(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) ^ swrlb:greaterThanOrEqual(?CNed, 12000000) -> meetDesignConditionC25(?Column, "Yes")

Rule3-4: ultimate axial load meet design condition (C90)

RectangleColumn(?Column) ^ hasAc(?Column, ?ColAc) ^ hasAs(?Column, ?CAs) ^ hasConcreteC90(?Column, ?Con) ^ C90(?Con) ^ hasfckC90(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) ^

```
swrlb:greaterThanOrEqual(?CNed, 12000000) ->
meetDesignConditionC90(?Column, "Yes")
```

6.2.3.2. Assessing building elements against sustainability aspects while considering axial load capacity

In this research, the embodied carbon content is taken as the indicator to study the environmental impact of the extracted columns. As mentioned earlier, it is necessary to treat the considered factors all at the same time in order to provide a holistic decision that considers various aspects. Consequently, the embodied carbon content and the rest of the aspects will be reviewed while considering only the Columns that satisfied the load capacity applied. The variable “meetDesignCondition” was used to apply the design condition factor. Some of the rules and queries related to sustainability are modelled in the proposed ontology as below:

Some of the SWRL rules and SQWRL queries used to answer Q-b

Rule7-1: total embodied CO2eC25 while considering design condition

```
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^
hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasEmbodiedCO2eC25(?Con,
?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) ->
hasTotalEmbodiedCO2eC25(?Column, ?TECO2)
```

Rule7-2: total embodied CO2eC35 while considering design condition

```
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^
hasConcreteC35(?Column, ?Con) ^ C35(?Con) ^ hasEmbodiedCO2eC35(?Con,
?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) ->
hasTotalEmbodiedCO2eC35(?Column, ?TECO2)
```

Rule7-3: total embodied CO2eC80 while considering design condition

```
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^
hasConcreteC80(?Column, ?Con) ^ C80(?Con) ^ hasEmbodiedCO2eC80(?Con,
?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) ->
hasTotalEmbodiedCO2eC80(?Column, ?TECO2)
```

Rule7-4 - total embodied CO2eC90 while considering design condition

```
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^
hasConcreteC90(?Column, ?Con) ^ C90(?Con) ^ hasEmbodiedCO2eC90(?Con,
?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) ->
hasTotalEmbodiedCO2eC90(?Column, ?TECO2)
```

Q7-1: Total embodied CO2e -C25 while considering design condition

```
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasVolume(?Column,
?CV) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^
hasEmbodiedCO2eC25(?Con, ?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) ^
hasTotalEmbodiedCO2eC25(?Column, ?TECO2) ^
meetDesignConditionC25(?Column, "Yes") ^ hasNedC25(?Column, ?CNed) ->
sqwrl:select(?Column, ?Name, ?CV, ?Con, ?CNed, ?ECO2, ?TECO2)
```

Q7-2: Total embodied CO2e -C35 while considering design condition

```
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasVolume(?Column,
?CV) ^ hasConcreteC35(?Column, ?Con) ^ C35(?Con) ^
hasEmbodiedCO2eC35(?Con, ?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) ^
hasTotalEmbodiedCO2eC35(?Column, ?TECO2) ^
meetDesignConditionC35(?Column, "Yes") ^ hasNedC35(?Column, ?CNed) ->
sqwrl:select(?Column, ?Name, ?CV, ?Con, ?CNed, ?ECO2, ?TECO2)
```

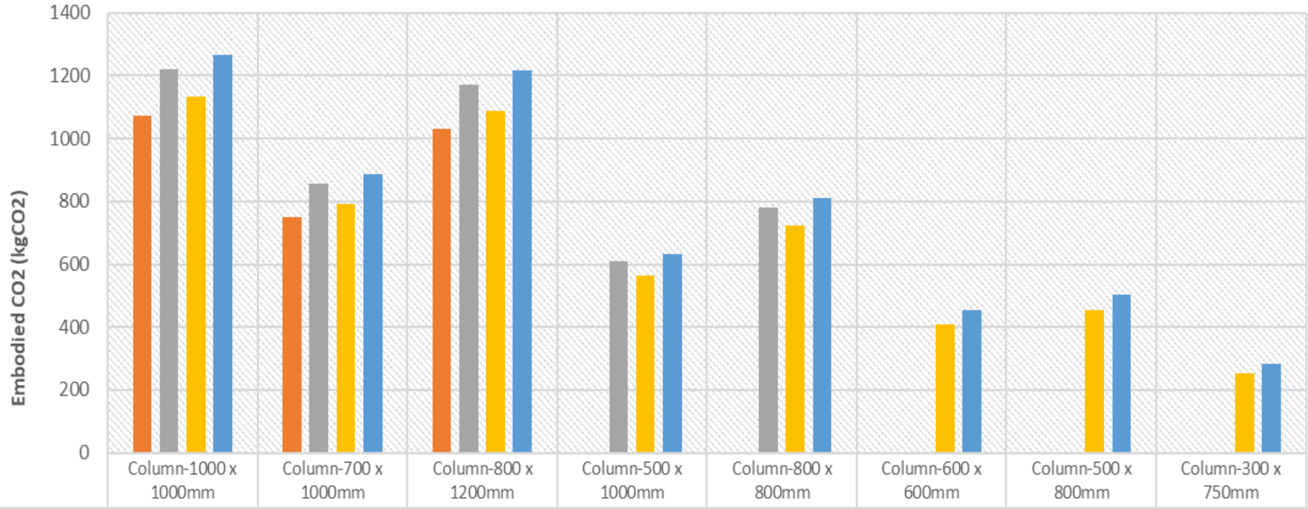
Q7-3: Total embodied CO2e -C80 **while considering design condition**

```
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasVolume(?Column,
?CV) ^ hasConcreteC80(?Column, ?Con) ^ C80(?Con) ^
hasEmbodiedCO2eC80(?Con, ?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) ^
hasTotalEmbodiedCO2eC80(?Column, ?TECO2) ^
meetDesignConditionC80(?Column, "Yes") ^ hasNedC80(?Column, ?CNed) ->
sqwrl:select(?Column, ?Name, ?CV, ?Con, ?CNed, ?ECO2, ?TECO2)
```

Q7-4: Total embodied CO2e -C90 **while considering design condition**

```
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasVolume(?Column,
?CV) ^ hasConcreteC90(?Column, ?Con) ^ C90(?Con) ^
hasEmbodiedCO2eC90(?Con, ?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) ^
hasTotalEmbodiedCO2eC90(?Column, ?TECO2) ^
meetDesignConditionC90(?Column, "Yes") ^ hasNedC90(?Column, ?CNed) ->
sqwrl:select(?Column, ?Name, ?CV, ?Con, ?CNed, ?ECO2, ?TECO2)
```

In Figure 50, the total embodied CO2 for each column type was calculated by taking various concert types such as C25, C35, C80 and C90 while considering design criteria. Thus, users can investigate the effect of various concrete materials on the selected building elements. It was shown that not all the columns passed the design condition (the applied axial load limit). For instance, when C25 was selected only three Columns passed the check, which are “Column-1000 x 1000mm”, “Column-700 x 1000mm”, “Column-800 x 1200mm”. Hence, the proposed ontology calculated the embodied CO2 only for those columns. Moreover, in Figure 50, the column with the lowest total embodied CO2 was “Column-300 x 750mm” when using C80, while the column with the highest total embodied CO2 was“Column-1000 x 1000mm” recorded with concrete C90. Taking “Column-300 x 750mm” as an example, using this type of column with C80 instead of C90 can decrease the total embodied CO2 by 11.6%. Furthermore, normally, the column with a minimum embodied carbon content is considered the most sustainable design solution. However, another design consideration is required mainly related to design safety.



	Column-1000 x 1000mm	Column-700 x 1000mm	Column-800 x 1200mm	Column-500 x 1000mm	Column-800 x 800mm	Column-600 x 600mm	Column-500 x 800mm	Column-300 x 750mm
TEmbodied25	1073.972816	751.172816	1030.932816					
TEmbodied35	1221.693984	854.493984	1172.733984	609.693984	781.053984			
TEmbodied80	1133.859776	793.059776	1088.419776	565.859776	724.899776	406.819776	452.259776	253.459776
TEmbodied90	1265.611088	885.211088	1214.891088	631.611088	809.131088	454.091088	504.811088	282.911088

Figure 50: Comparison of total embodied CO2 of different concrete in respect to the selected columns while considering design condition (load capacity)

In terms of the economic aspect of sustainability, the cost of the material used in a column is considered. The cost of a material associated with a certain column can be calculated through the following rules and queries:

Some of the SWRL rules used to answer Q-c

Rule9-1: total cost of square columnC25

RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasCostC25(?Con, ?Cost) ^ swrlb:multiply(?TCost, ?CV, ?Cost) -> hasTotalCostC25(?Column, ?TCost)

Rule9-2: total cost of square columnC35

RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ hasConcreteC35(?Column, ?Con) ^ C35(?Con) ^ hasCostC35(?Con, ?Cost) ^ swrlb:multiply(?TCost, ?CV, ?Cost) -> hasTotalCostC35(?Column, ?TCost)

Some of the SQWRL queries used to answer Q-c

Q9-1: total cost of columnC25

RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasVolume(?Column, ?CV) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasCostC25(?Con, ?Cost)

^ hasTotalCostC25(?Column, ?TCost) ^ meetDesignConditionC25(?Column, "Yes") ^ hasNedC25(?Column, ?CNed) -> sqwrl:select(?Column, ?Name, ?CV, ?Con, ?CNed, ?Cost, ?TCost)

Q9-1: total cost of columnC35

RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasVolume(?Column, ?CV) ^ hasConcreteC35(?Column, ?Con) ^ C35(?Con) ^ hasCostC35(?Con, ?Cost) ^ hasTotalCostC35(?Column, ?TCost) ^ meetDesignConditionC35(?Column, "Yes") ^ hasNedC35(?Column, ?CNed) -> sqwrl:select(?Column, ?Name, ?CV, ?Con, ?CNed, ?Cost, ?TCost)

The cost of material used can be calculated according to the different types of elements and concrete materials used. For instance, a column with dimensions of 300 x 750 mm and a depth of 4m with a concrete material of type C90 can cost 232 $US, Figure 51. Using “Column-300 x 750mm” with C80 instead of C90 can decrease the cost of the material used by 7.4%. Moreover, the column with the lowest total cost was “Column-300 x 750mm” with concrete C80, while the column with the highest total cost was “Column-1000 x 1000mm” when using concrete C90. Although using high-strength concrete can help in reducing the section dimension of a column, it cannot be applied to all scenarios. In some cases, reducing the section dimension of a column is not possible due to other factors mainly related to design fire safety, such as fire resistance.

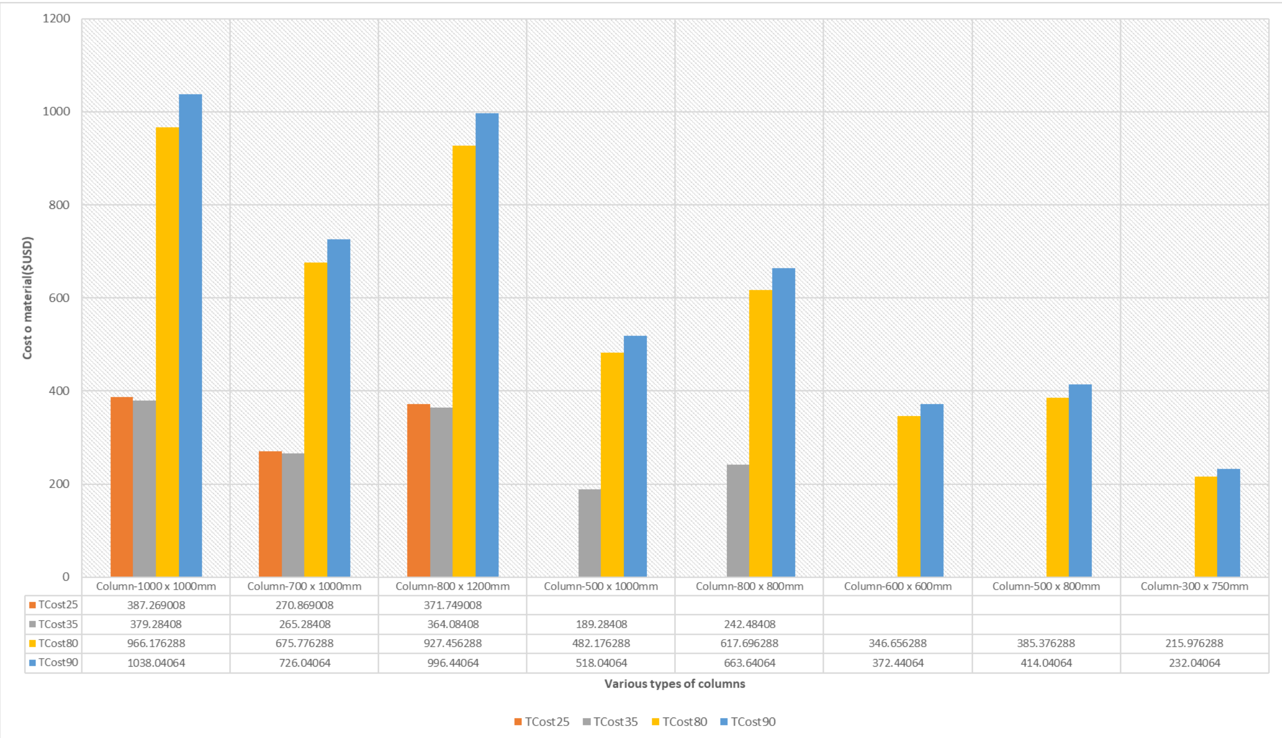


Figure 51: Comparison of the material cost of different concrete in respect to the selected columns while considering design conditions

6.2.3.3. Considering the cost of the labour based on dimensions and reinforcement used

Following the equations defined in chapter 2, and in order to consider the total labours cost of a rectangular column earlier in the design stage, several SWRL rules and SQWRL queries have been implemented in the proposed ontology. For instance, to calculate the labours cost of column concreting, reinforcement, and shuttering work, the following SWRL rules were added:

Some of the SWRL rules used to answer Q-d

Rule11-1: Labour Cost rectangular column Concreting

RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ LaborConcretingCost(?LaborConCost) ^ hasLaborConcretingCost(?LaborConCost, ?hasLaborConCost) ^ swrlb:multiply(?LCC, ?CV, ?hasLaborConCost) -> hasLaborCostConcreting(?Column, ?LCC)

Rule11-2: Labour Cost rectangular column Reinforcement

RectangleColumn(?Column) ^ hasWeightSteel(?Column, ?WS) ^ LaborReinforcementCost(?LaborReinfCost) ^ hasLaborReinforcementCost(?LaborReinfCost, ?hasLaborReinfCost) ^ swrlb:multiply(?LCR, ?WS, ?hasLaborReinfCost) -> hasLaborCostReincorcement(?Column, ?LCR)

Rule11-3: Labour Cost rectangular column formwork

RectangleColumn(?Column) ^ hasTotalAreaShulteringWork(?Column, ?TASHW) ^ LaborShulteringCost(?LaborShultCost) ^ hasLaborShulteringCost(?LaborShultCost, ?hasLaborShultCost) ^ swrlb:multiply(?LCF, ?TASHW, ?hasLaborShultCost) -> hasLaborCostFormwork(?Column, ?LCF)

The summation of all the above three variables will result in the total labours cost of the column, which was represented in the ontology as follows:

Some of the SWRL rules and SQWRL queries used to answer Q-d

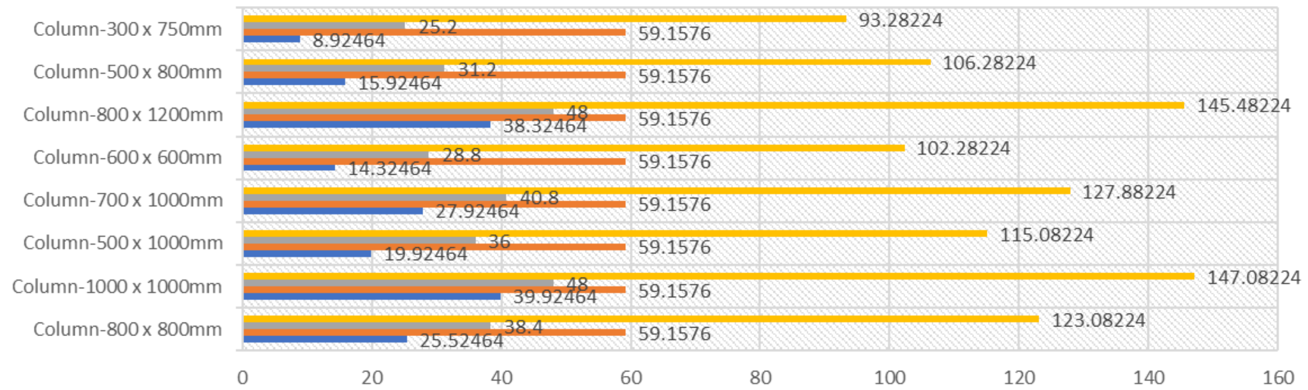
Rule11-4: Total Labour Cost rectangular column

RectangleColumn(?Column) ^ hasLaborCostConcreting(?Column, ?LCC) ^ hasLaborCostReincorcement(?Column, ?LCR) ^ hasLaborCostFormwork(?Column, ?LCF) ^ swrlb:add(?TotalLCC, ?LCC, ?LCR, ?LCF) -> hasTotalLaborCostColumn(?Column, ?TotalLCC)

Q11-4: Total Labour Cost rectangular column

```
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^
hasLaborCostConcreting(?Column, ?LCC) ^
hasLaborCostReincorcement(?Column, ?LCR) ^ hasLaborCostFormwork(?Column,
?LCF) ^ swrlb:add(?TotalLCC, ?LCC, ?LCR, ?LCF) ^
hasTotalLaborCostColumn(?Column, ?TotalLCC) -> sqwrl:select(?Column, ?Name,
?LCC, ?LCR, ?LCF, ?TotalLCC)
```

The syntax swrlb:add(?TotalLCC, ?LCC, ?LCR, ?LCF) was used to model the total cost variable. The output of executing Query 11-4 is presented in Figure 52, where the total labour cost of rectangular columns is calculated based on the labour cost of column concreting, reinforcement, and shuttering work.



	Column-800 x 800mm	Column-1000 x 1000mm	Column-500 x 1000mm	Column-700 x 1000mm	Column-600 x 600mm	Column-800 x 1200mm	Column-500 x 800mm	Column-300 x 750mm
Total LC	123.08224	147.08224	115.08224	127.88224	102.28224	145.48224	106.28224	93.28224
LC formwork	38.4	48	36	40.8	28.8	48	31.2	25.2
LC reinforcement	59.1576	59.1576	59.1576	59.1576	59.1576	59.1576	59.1576	59.1576
LC concreting	25.52464	39.92464	19.92464	27.92464	14.32464	38.32464	15.92464	8.92464

Total LC ■ LC formwork ■ LC reinforcement ■ LC concreting

Figure 52: The total labour cost in respect to the selected columns.

6.2.3.4. Multi-objective knowledge base considering various aspects

It is shown that the previous sections stated the possibility of combining different resources and conditions such as standards and databases to produce collective decisions. Consequently, in an effort to move from a single objective decision making to a multi-objective decision that considers various factors together, which also satisfies design regulations, the above-mentioned aspects were put all together with several restrictions related to design criteria. For instance, taking C25/30 as an example and by selecting this type of material, an engineer can look into several factors together, such as embodied carbon content, exposure condition, cost of the material used and the total labour cost, while considering design conditions such as load capacity and fire resistance.

Fire resistance is another condition that needs to be considered while designing a column, and this factor is usually collected from tables or statements, as mentioned in Chapter 2. Three different conditions of fire resistance have been modelled in the proposed ontology. These conditions rely on two factors: the minimum width of the selected column and the minimum concrete cover (distance between the surface of the concrete and the reinforcement). The cover was set to 25 cm in this study, whereas the dimensions were extracted automatically from the IFC file. In the proposed ontology, this has been modelling as follows:

Some of the SWRL rules used to answer Q-e
Rule4-1: Fire resistance time 60
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasWidth(?Column, ?Width) ^ hasCover(?Column, ?Cover) ^ swrlb:greaterThanOrEqual(?Width, 200) ^ swrlb:lessThan(?Width, 300) ^ swrlb:greaterThanOrEqual(?Cover, 25) -> hasFireResistanceTime60(?Column, "R60")
Rule4-2: Fire resistance time 90
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasWidth(?Column, ?Width) ^ hasCover(?Column, ?Cover) ^ swrlb:greaterThanOrEqual(?Width, 300) ^ swrlb:lessThan(?Width, 350) ^ swrlb:greaterThanOrEqual(?Cover, 25) -> hasFireResistanceTime90(?Column, "R90")
Rule4-3: Fire resistance time 120
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasWidth(?Column, ?Width) ^ hasCover(?Column, ?Cover) ^ swrlb:greaterThanOrEqual(?Width, 350) ^ swrlb:greaterThanOrEqual(?Cover, 25) -> hasFireResistanceTime120(?Column, "R120")

Some of the SQWRL queries used to answer Q-e
Q4-1: Fire resistance time 60
swrlb:lessThan(?Width, 300) ^ RectangleColumn(?Column) ^ swrlb:greaterThanOrEqual(?Cover, 25) ^ hasFireResistanceTime60(?Column, "R60") ^ hasWidth(?Column, ?Width) ^ hasName(?Column, ?Name) ^ hasCover(?Column, ?Cover) ^ swrlb:greaterThanOrEqual(?Width, 200) -> sqwrl:select(?Column, ?Width, ?Cover, "R60")
Q4-2: Fire resistance time 90

```
hasFireResistanceTime90(?Column, "R90") ^ RectangleColumn(?Column) ^
swrlb:greaterThanOrEqual(?Width, 300) ^ swrlb:lessThan(?Width, 350) ^
swrlb:greaterThanOrEqual(?Cover, 25) ^ hasWidth(?Column, ?Width) ^
hasName(?Column, ?Name) ^ hasCover(?Column, ?Cover) ->
sqwrl:select(?Column, ?Width, ?Cover, "R90")
```

Q4-3: Fire resistance time 120

```
hasFireResistanceTime120(?Column, "R120") ^ RectangleColumn(?Column) ^
swrlb:greaterThanOrEqual(?Width, 350) ^ swrlb:greaterThanOrEqual(?Cover, 25) ^
hasWidth(?Column, ?Width) ^ hasName(?Column, ?Name) ^ hasCover(?Column,
?Cover) -> sqwrl:select(?Column, ?Width, ?Cover, "R120")
```

These different conditions have been implemented using the syntaxes “swrlb:greaterThanOrEqual” and “swrlb:lessThan”. It was concluded that the increase of column dimensions and the concrete cover could result in good fire resistance. However, selecting the column dimension also affects the ultimate load capacity, as discussed before and other factors such as sustainability. Hence, those conditions need to be considered at the same time while designing a structure. Those factors were all put into one SQWRL query to provide a multi-objective decision. Furthermore, different exposure conditions were added according to the strength of the concrete. Since various concrete strengths are set as a constraint in this ontology, these conditions will be shown as a recommendation while reviewing a certain type of concrete. For instance, when selecting C25/30 as concrete material, the query will show that this concrete type is suitable for exposures such as carbonation-induced corrosion and freeze/thaw attack.

Taking C25 and C90 as an example, the following SQWRL queries have been modelled:

Some of the SQWRL queries used to answer Q-e

Holistic design of rectangular column considering multiple aspects with C25 and R120

```
C25(?Con) ^ hasfckC25(?Con, ?Confck) ^ hasTotalCostC25(?Column, ?TCost) ^
hasName(?Column, ?Name) ^ hasTotalLaborCostColumn(?Column,
?TotalLabCostColumn) ^ hasFireResistanceTime120(?Column, "R120") ^
RectangleColumn(?Column) ^ hasConcreteC25(?Column, ?Con) ^
hasTotalEmbodiedCO2eC25(?Column, ?TECO2) ^ hasWidth(?Column, ?Width) ^
hasLength(?Column, ?Length) ^ meetDesignConditionC25(?Column, "Yes") ^
hasCover(?Column, ?Cover) ^ hasXC1(?Con, ?expossure) ->
sqwrl:select(?Column, ?Name, ?Width, ?Length, ?Cover, "R120", ?expossure,
?Confck, ?TECO2, ?TCost, ?TotalLabCostColumn)
```

Holistic design of rectangular column considering multiple aspects with C90 and R90
C90(?Con) ^ hasfckC90(?Con, ?Confck) ^ hasName(?Column, ?Name) ^ hasTotalLaborCostColumn(?Column, ?TotalLabCostColumn) ^ hasXC1XD3XS23XF3XA3(?Con, ?expossure) ^ RectangleColumn(?Column) ^ hasConcreteC90(?Column, ?Con) ^ hasTotalCostC90(?Column, ?TCost) ^ hasTotalEmbodiedCO2eC90(?Column, ?TECO2) ^ hasWidth(?Column, ?Width) ^ hasLength(?Column, ?Length) ^ meetDesignConditionC90(?Column, "Yes") ^ hasCover(?Column, ?Cover) ^ hasFireResistanceTime90(?Column, "R90") -> sqwrl:select(?Column, ?Name, ?Width, ?Length, ?Cover, ?Confck, ?expossure, ?TECO2, ?TCost, ?TotalLabCostColumn)

The workflow of the developed multi-objective knowledge base considering C25/30 and R120 as construction material and fire resistance, respectively, can be shown in Figure 53, for instance:

Step 1:

If **C25/30** was selected a **concrete material** → Calculate **ultimate load capacity** of all rectangular columns → Compare the **ultimate load capacity** to the **axial load capacity** (applied axial load) → If the **ultimate load capacity** is **less than** the **applied axial load** → **Failed – Neglect column.**

Step 2:

If **C25/30** was used as the **concrete material** → Calculate the **ultimate load capacity** of all rectangular columns → Compare the **ultimate load capacity** to the **axial load capacity** (applied axial load) → If the **ultimate load capacity** is **greater than** or **equal** to the **applied axial load** → **Pass – Select the rectangular columns that meet the axial applied axial load**.

Then if the **concrete cover** was **greater than** or **equal to** 25 mm **AND** If the **column width** was **greater than** or **equal** to 350 mm → S**elect all the columns that meet the Fire Resistance condition (R120)** → show and calculate the following parameters:

Recommend **Exposure conditions**

Calculate the **total cost of material** for each rectangular column

Calculate the **total labour cost** for each rectangular column

Calculate the **total Embodied CO2e** for each rectangular column

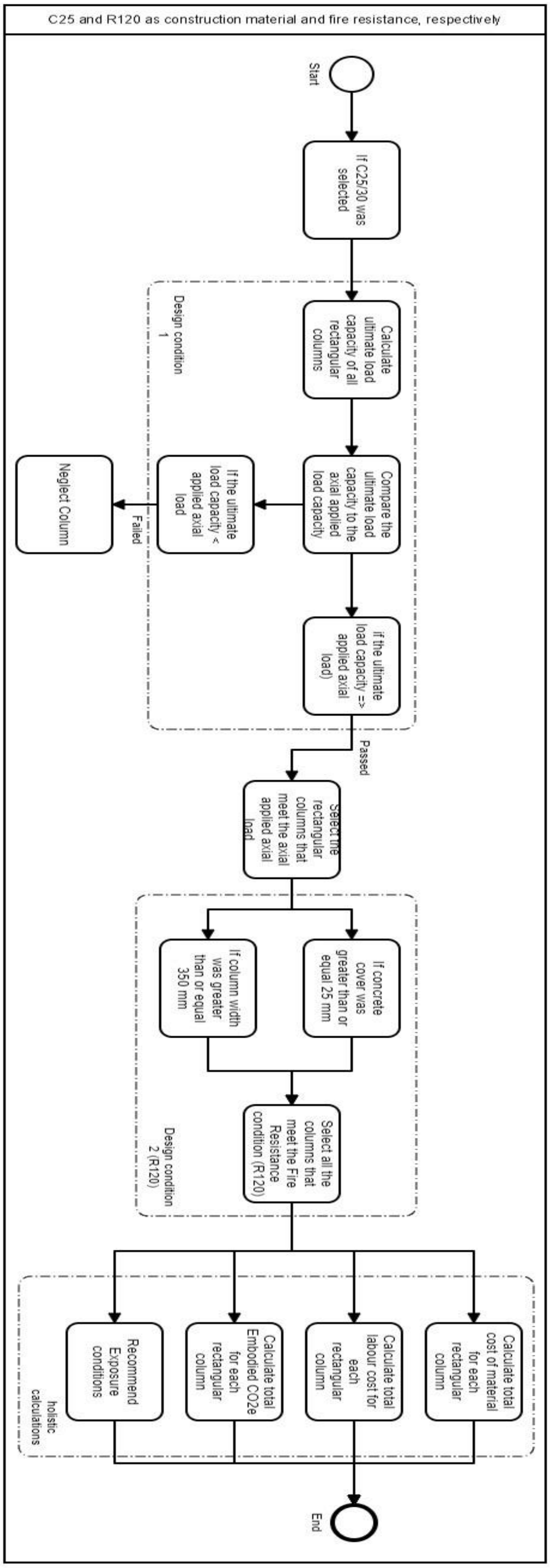


Figure 53: Workflow of the developed multi-objective knowledge base considering C25/30 and R120 as construction material and fire resistance, respectively.

Another scenario considering C90/105 as the concrete material and R90 as the fire resistance was conducted to elaborate more on the developed multi-objective knowledge base. The workflow can be shown in Figure 54. It will first go through step 1, as explained before, and then continue with step 2.

Step 1:

If **C90/105** was selected a **concrete material** → Calculate **ultimate load capacity** of all rectangular columns → Compare the **ultimate load capacity** to the **axial load capacity** (applied axial load) → If the **ultimate load capacity** is **less than** the **applied axial load** → **Failed – Neglect column.**

Step 2:

If **C90/105** was used as the **concrete material** → Calculate the **ultimate load capacity** of all rectangular columns → Compare the **ultimate load capacity** to the **axial load capacity** (applied axial load) → If the **ultimate load capacity** is **greater than** or **equal** to the **applied axial load** → **Pass – Select the rectangular columns that meet the axial applied axial load** →

Then if the **concrete cover** was **greater than** or **equal to** 25 mm **AND** If the **column width** was **greater than** or **equal** to 300 mm **AND** if the **column width** was **less than** 350mm → S**elect all the columns that meet the Fire Resistance condition (R90)** → show and calculate the following parameters:

Recommend **Exposure conditions**

Calculate the **total cost of material** for each rectangular column

Calculate the **total labour cost** for each rectangular column

Calculate the **total Embodied CO2e** for each rectangular column

The outputs of executing the “Holistic design of rectangular column considering multiple aspects with C25 and R120” and “Holistic design of rectangular column considering multiple aspects with C90 and R90” queries are presented in Tables 13 and 14, respectively.

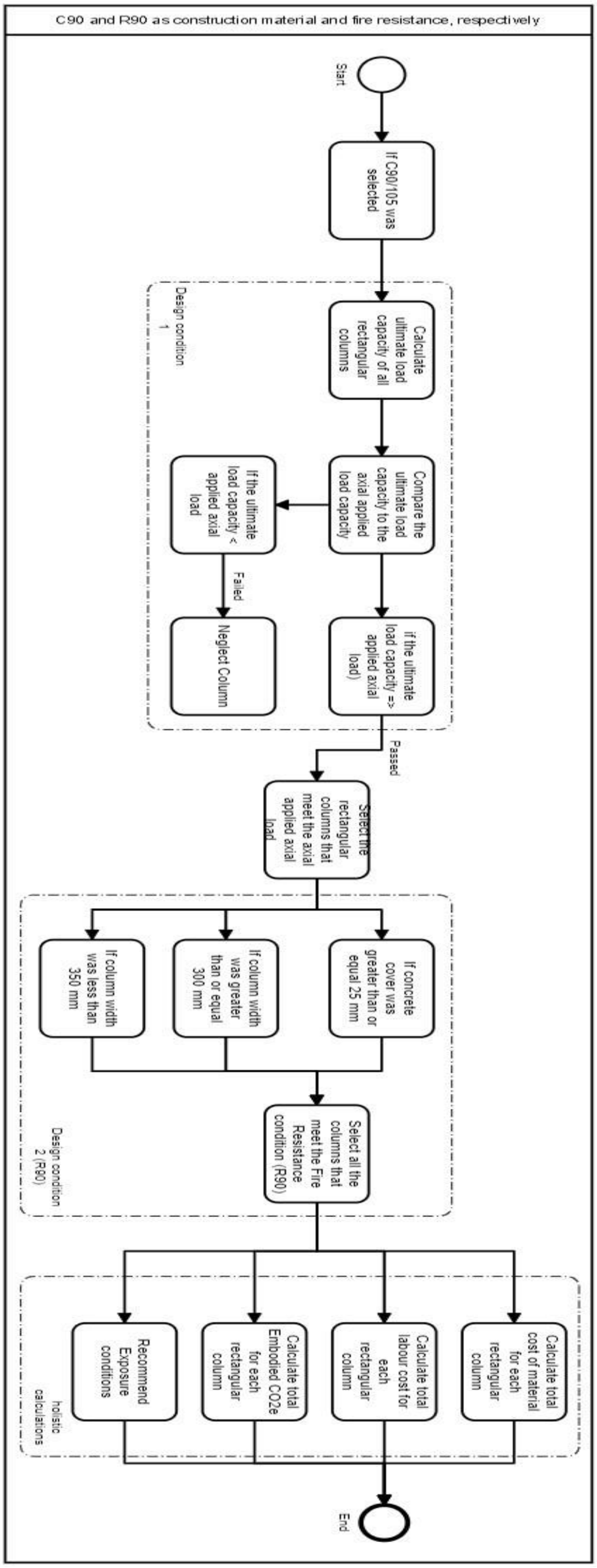


Figure 54: Workflow of the developed multi-objective knowledge base considering C90 and R90 as construction material and fire resistance, respectively

Table 13: Results collected from the Holistic design of rectangular column query with C25 and R120.

Name	Width (mm)	Length (mm)	Cover (mm)	Fire resistance	Expsossure condition	Concrete strength	Total embodied CO2(kgCO2)	Total material cost(USD$)	Total labor cost(USD$)
Column-1000 x 1000mm	1000	1000	25	R120	suitable for expossure: (1) Carbonation-induced corrosion; (2) Freeze/Thaw Attack	30	1073.97	387.269	147.0822
Column-700 x 1000mm	1000	700	25	R120	suitable for expossure: (1) Carbonation-induced corrosion; (2) Freeze/Thaw Attack	30	751.172	270.869	127.8822
Column-800 x 1200mm	800	1200	25	R120	suitable for expossure: (1) Carbonation-induced corrosion; (2) Freeze/Thaw Attack	30	1030.93	371.749	145.4822

Table 14:Results collected from the Holistic design of rectangular column query with C90 and R90.

Name	Width (mm)	Length (mm)	Cover (mm)	Fire resistance	Expsossure condition	Concrete strength	Total embodied CO2(kgCO2)	Total material cost(USD$)	Total labor cost(USD$)
Column-300 x 750mm	300	750	25	R90	suitable for expossure: (1) Carbonation-induced corrosion; (2) Chloride-induced corrosion; (3) Chloride-induced corrosion from sea-water; (4) Freeze/Thaw Attack; (5) Chemical Attack	105	282.911	232.04	93.28

6.3. Summary

This chapter illustrates the data exchanging framework through scenario-based case testing. It was shown that the framework can provide different data sets by extracting only the critical information and correctly processing the IFC-based BIM model using the proposed data exchange method. Moreover, the data acquisition method helped produce a more dynamic knowledge base that connects real project information to static information related to cost and sustainability efficiently. Consequently, this approach is proved to be more efficient than a manual approach by adding data to the knowledge base. The SWRL rules helped automate all the manual calculations and generate new facts based on the data in an IFC file. The built-in rules allow the end-user to review and compare different design alternatives by considering various factors at an early stage. Hence, the proposed framework provided a multi-objective decision that considers different sources of information together. The proposed framework has the potential to serve as a complex decision-making framework by involving more design factors and building elements. However, further development is required to enhance the data exchange method and extend the multi-objective knowledge base. The developed framework will be delivered to experts in this field to reflect more practical situations for further validation.

Chapter 7. Conclusion

Reflecting on the observations and findings from previous sections, this chapter summarises the research by revisiting the hypothesis and the pre-defined research questions. After that, the research limitations and further improvements that can be made are discussed. Finally, a summary of the research contributions of this thesis is presented.

7.1. Revisiting the hypothesis

In order to clarify the aim of this research, a proposed research hypothesis was assumed as follows:

To establish an automatic data exchanging framework that orchestrates different functions holistically through automatic information exchange supported by an ontological approach for holistic spatial co-ordination building design.

The hypothesis was then decomposed into six research questions, which are discussed below, based on findings from the previous chapters. Although it was initially envisaged that each chapter would concentrate on particular research questions, the findings from all chapters combined are utilised to remind more compressive answers.

Q1: What are the concepts, technologies, and tools existing within the BIM ecosystem to improve interoperability and decision making in the AEC industry? And how are those concepts and tools backing BIM development while considering their scope and limitations?

The previous work determined that the development of OpenBIM concepts is not focused only on the technical aspects. Instead, it goes further to cover the joint efforts issues, including the legal, semantic, knowledge processing, and organizational aspects, to reach a high level of collaboration and integration. Thus, the BIM ecosystem can be described as a system of technologies, processes and policies that should interact and dynamically function together to embrace continuous variations in the AEC industry.

It was shown that data exchange and decision making play a major part in the development process of the BIM ecosystem. However, they are still facing interoperability issues that require further consideration. The IFC format has been used to provide information exchange, which was supported by concepts such as IDM and MVD. However, the development process of an IDM-MVD is complicated, as was concluded from the literature, but those concepts also become evident in assessing and developing the proposed framework in Chapters 4 and 5. Moreover, it was concluded that this format was only utilised to deliver the information within the sender's scope. It does not infer any new information. Thus, additional methods, technologies, and formats must be utilised with this format and concepts to enhance its performance, especially since BIM is moving towards knowledge processing.

Technology such as ontology has shown the potential to improve interoperability issues within BIM models by implementing domain knowledge, which can provide semantic enrichment of the BIM model and also enhance IFC format performance. Several ontologies have been developed, as was concluded from the literature. However, most of the developed ontologies were developed separately to serve a single objective decision and require manual input to process the data in a BIM model. There is a lack of a multi-objective knowledge base within the BIM context that combines different resources and conditions such as standards and databases to produce a collective decision. Despite the effort to develop a multi-objective decision-making knowledge base using ontology within the BIM context, most of the research did not provide a method that can work in parallel with a BIM model automatically.

Taking advantage of BIM rich building model throughout the building lifecycle and combining it with a well-established multi-objective knowledge base can achieve a significant viable data exchanging framework. Consequently, considering the literature review that was carried out in this thesis, an automatic data exchanging framework that combines both a data exchange method and semantic web technology was proposed to eliminate inefficiencies in data sharing and improve decision-making in the early design stage.

Q2: What is required to identify a data exchanging framework to support the collaborative design and decision-making from a data processing perspective?

The first requirement consists in providing a data exchange method that can help in exchanging only the critical information by identifying a “single truth of information” that can be shared among different domains and throughout the lifecycle. This MCDS can help developers use the same mapped data set to create their idea and extend this mapped data set to fit their specific use case, eliminating the time wasted on rebuilding this common information again. Consequently, a CDA referencing various concepts such as the standardised IDM, MVD and the concept of the semantic intersection was designed in Chapter 4 to conclude a “single truth of information” and “partial truth of information” data sets that form the basis for the proposed framework from a data processing perspective.

Three BIM models were created: the architectural model was considered as the source model, whereas the structural and cost models represent the models delivered to the end-users. The analysis found that despite the different use of the BIM domain-specific partial model, they share some commonalities that are not exclusive to a specific domain. The more commonalities exist within two models, the less data loss will exist between the two models, and the flow of data is not always bidirectional. It flows from the more informed data sets to the less informed ones. However, there is always a set of information that cannot be exchanged, which can be direction sensitive or not defined since it is out of the sender's design scope.

Through the literature review that was carried out in Chapter 2 and the analysis conducted in Chapter 4, it was concluded that the IFC format is only utilised to deliver the information within the sender's scope without inferring any new statements or information. Hence, the second requirement was to provide a multi-objective knowledge base that can support the IFC format by inferring any missing information or new facts based on existing information received from the sender. Thus, as explained in Section 2.4, different data sources were put together, which were demonstrated in Chapters 5 and 6, to move from a single objective to a multi-objective decision that considers various factors together, which will help in extending interoperability to other knowledge domains.

Q3: What needs to be considered to build a data exchange method to convert from A model to B model in order to automatically realise data exchange?

In chapter 5, this study presented a data exchange method to eliminate inefficiencies in data sharing. In order to map the defined datasets, which was concluded in Chapter 4, to the IFC data structure, an object-oriented modelling notation approach based on Express-G was utilised (Section 5.1.1). Following that, a data extraction tool was developed based on the IFCOpenShell library. As shown in Chapter 6, the data exchange method developed was used to improve the efficiency of reasoning and querying since this procedure can help reduce the processing time by only showing only the critical information instead of the entire model. The proposed method played a significant role in simplifying the process of obtaining related data from a complex BIM model. The tool uses the IFC format, which is a neutral format. Consequently, it can provide a flexible input that can merge easily with other technologies and data.

Q4: What needs to be considered for designing a holistic knowledge base that considers various aspects such as design conditions, sustainability, and cost to support building design?

Q5: Can the holistic knowledge base be aligned with the data exchange method to provide an automated framework?

On the other hand, there are many factors required to be considered in order to achieve the best possible design. It was shown that ontologies are capable of integrating data and resources from different design perspectives. Thus, a taxonomy, which includes the basic underlying concepts discussed in Chapter 2, was then used to implement a multi-objective knowledge base using an ontological approach, as demonstrated in Chapters 5 and 6. The proposed knowledge base was developed based on the common data and aspects, which were investigated in Chapters 2 and 4. It helped provide decisions associated with various aspects such as sustainability and cost while considering design conditions. Several SWRL and SQWRL queries were developed to provide and facilitate the multi-objective knowledge base (Section 5.2.2). It was shown in Chapter 6 that an ontology representation can allow retrieval of different queries that consider different conditions together, which in return provide a semantically rich environment. However, it relies significantly on domain knowledge.

Furthermore, despite the effort to develop a multi-objective decision-making knowledge, most of the research, as was concluded in Chapter 2, did not provide a method that can work in parallel with a BIM model automatically. The previously developed knowledge bases covered only static knowledge and required manual input to process the data in a BIM model. A static knowledge base does not allow the engineers to review several design choices based on the real data within a model. Consequently, one of the main contributions of this research was the proposed data acquisition method that dynamically links data between the IFC-based BIM model and the multi-objective knowledge base, which can eliminate human involvement by decreasing manual input. Hence, in this research, a data acquisition method was proposed to automatically align the static knowledge with the information provided from an IFC file (Section 5.2.2).

Q6: Can the proposed framework provide the necessary information automatically and at the same time help end-users to compare different design choices related to sustainability and cost factors while considering design conditions based on the existing data in an IFC-based BIM model?

The above research question is related to testing the framework implemented in Chapter 5. Several objectives were defined to test and validate the proposed framework in Chapter 6. However, more objectives need to be identified from industry practice for completeness. A scenario-based case testing on an airport model was used to test the proposed data exchange method and knowledge base. Moreover, the built knowledge base was verified through a pellet reasoner to ensure that the developed ontology is syntactically valid and meets the requirements. This ensured that all the concepts and information were correctly linked and defined.

The Data exchange method helped extract various data sets and the necessary information. The extracted data was utilised in the proposed ontology instead of the whole BIM model. Using the whole BIM model will slow down the ontology since it requires a high-performance computer to process the reasoning. The MCDS was used as input for the proposed data acquisition method. The data acquisition method helped merge the data existing in the original multi-objective knowledge base, which includes various static knowledge related to design conditions, sustainability, and cost, with the data extracted from the IFC file. The multi-objective knowledge base provided decisions associated with sustainability and cost, allowing users to review and compare the existing elements in an IFC-based BIM model to various factors while considering design conditions. The SWRL rules automated all the manual calculations and linked different resources. The reasoning queries were computed in a relatively speedy manner. However, it was concluded that aligning different resources using the semantic web requires significant effort and composite rules to provide complex relations among different factors or equations.

It was shown that the proposed framework can process the IFC-based BIM model correctly and also helped in generating new facts based on the IFC file data. The validation process proved that the proposed framework is functional and reliable for data exchange and holistic decision-making. Thus, it can be concluded that the hypothesis is true. Utilising only the necessary information from an IFC file and combining it with semantic web technology has the potential to provide a rich semantic environment that can solve some of the interoperability issues. However, implementing such a framework requires much investment in refining the MCDS and linking all the relevant data, which also requires knowledge from domain experts. The proposed framework has the potential to serve as a complex decision-making framework by involving more design factors and building elements. However, further development to extend it is required.

7.2. Research limitations and future works

The limitations and future work of this research are discussed below:

- Integrating advanced technologies with BIM into one system has become a hot research topic. Merging these technologies and taking advantage of their power can be an excellent opportunity to build a support framework that can level up BIM and the decision-making process across the building lifecycle. However, it may take several years to achieve that. This study is not meant to be exclusive. There are many technologies, concepts, and ideas that can be added to help BIM reach its full potential but are not covered in this study. Moreover, the analysis in Chapter 2 was based on the dataset retrieved from WoS and only included literature in English. Furthermore, in addition to the quantitative analysis, a qualitative study was considered, which cannot cover all the literature collected because of the large volume. Samples of the collected papers were reviewed to give insight into the current research. Thus, this study may not reflect the entire BIM literature on those topics.

- The current proposed framework is in the “proof of concept ” stage and requires further improvement. The developed framework is semi-automated, and the knowledge base covers the most necessary building elements and aspects. Future work will include fully automating the framework by developing a web-based platform that can host both the data exchange method and the developed multi-objective knowledge base, where they can communicate with each other without any human involvement.

- The creation of the BIM model in practice affects the use of the automatic data exchange method, which may vary according to the user's way of modelling. In this research, the required information has been classified into common information shared among several disciplines and specific common information related to a specific domain. The CDA conducted in this research is focused on the design stage. Consequently, data sets from other stages can be investigated in the future to refine the MCDS since it represents a refined form of a data set that can be shared among different domains and throughout the lifecycle.

- In this research, due to limited time, the developed ontology, including the SWRL rules embedded, rely mainly on the currently available resources. The reasoning and query functions focused primarily on rectangular columns. A thorough knowledge base covering various building elements will be developed in the future. Further work on integrating more concrete materials can be carried out. Moreover, other factors related to the overall cost can be added, such as the cost of the construction equipment.

- This study focused mainly on the design stage, with a bit of focus on the construction stage. Further work will include the investigation of more models from different stages. For instance, another aspect that can be added to the ontology is the safety of workers on the construction site by looking at safety precautions that need to be considered by the workers when they are building certain elements.

- The current developed multi-objective knowledge base considers various aspects. Future work can include assigning a weight coefficient for each aspect, which can help rank those aspects from major impact to a minimum impact. Consequently, it can help users observe which factor needs to be considered before the other, which can enhance the decision made within a project.

- The proposed framework was applied in the area of building design. However, it can be extended and applied to other fields, such as the infrastructure domain.

7.3. Research contribution

This research contains work related to several theoretical and practical developments in an effort to provide a data exchanging framework that combines both a data exchange method and a semantic web technology to eliminate inefficiencies in data sharing and improve the decision-making in the early design stage. Taking into account the findings and development presented in this dissertation, the main contributions resulting from this research are listed below:

1) Data mapping can differ from one developer to another. Consequently, a CDA referencing various concepts such as the IDM, MVD and the concept of the semantic intersection was designed, which formed the basis for a theoretical data exchanging framework to support collaborative design from data perception. This analysis helped in providing a minimum common data set (MCDS), which can help the developer to use the same mapped data set when they create their idea and extend this mapped data set to fit their specific use case, which can eliminate the time wasted on rebuilding this common information again. A data exchange method was implemented based on this MCDS. The developed data exchange method plays a significant role in simplifying the process of obtaining related data from a BIM model where users can extract only the critical information instead of working with a complex model. The tool uses the IFC format, which is a neutral format. Consequently, it can provide a flexible input that can merge easily with other technologies and data. A detailed article entitled: “A theoretical holistic decision-making framework supporting collaborative design based on common data analysis (CDA) method” was published in the Journal of Building Engineering.

2) To provide a multi-objective knowledge base that can assist engineers who lack knowledge associated with sustainability and cost in comparing different design choices while considering design conditions to develop an ideal design in the early stage. The multi-objective knowledge involves factors from different related sources. The proposed ontology was developed using a machine-readable format, allowing the chance to add more concepts to it in the future and work with other automated tools. A conference paper: “Knowledge-driven holistic decision making supporting multi-objective Innovative Design”, was published in the proceedings of the 2nd International Conference on Sustainable Smart Manufacturing 2019 and was indexed in the book “Industry 4.0 – Shaping the Future of The Digital World”. Furthermore, joint research with an MSc student resulted in a paper entitled “nD

knowledge base for comprehensive structural design", submitted to the journal "Advances in Engineering Software" and is currently under review.

3) One of the main contributions of this research is to align the developed multi-objective knowledge base with the data exchange method to extract information from an IFC file and merge them with the data presented in the developed ontology to eliminate the human involvement by decreasing manual input. Consequently, this study developed a method that dynamically links data between the IFC-based BIM model and the multi-objective knowledge base. This combined method is unlike previous research approaches where data and instances are entered manually one by one. It presents a more direct way to work with IFC-based BIM models in order to evaluate various aspects. The extraction tool proposed in this research will serve various data sets that can be utilised in the proposed ontology instead of the whole BIM model since using the whole BIM model will slow down the ontology. The technical contribution in this research, including the data exchange method, the multi-objective ontology and the automatic data acquisition method, is currently in the preparation stage and will be submitted to the journal "Advances in Engineering Software". Moreover, joint research entitled "Aligning BIM and ontology for information retrieve and reasoning in value for money assessment", which was based on using ontology and BIM, was carried out and published in "Journal of Automation in Construction".

Bibliography

2, E. (2004) ‘Eurocode 2: Design of concrete structures - Part 1-2: General rules - Structural fire design’, 1. Available at: https://www.phd.eng.br/wp-content/uploads/2015/12/en.1992.2.2005.pdf.

Abanda, F. H. *et al.* (2011) ‘An ontology-driven building construction labour cost estimation in Cameroon’, *Electronic Journal of Information Technology in Construction*, 16(June 2018), pp. 617–634.

Abanda, F. H., Kamsu-Foguem, B. and Tah, J. H. M. (2017) ‘BIM – New rules of measurement ontology for construction cost estimation’, *Engineering Science and Technology, an International Journal*. Karabuk University, 20(2), pp. 443–459. doi: 10.1016/j.jestch.2017.01.007.

Abualdenien, J., Pfuhl, S. and Braun, A. (2019) ‘Development of an MVD for checking fire-safety and pedestrian simulation requirements’, *Proc. of the 31th Forum Bauinformatik*, (September), p. 191. Available at: https://publications.cms.bgu.tum.de/abualdenien_pfuhl_braun_fbi2019.pdf (accessed 2 November2020).

Ag, S. (2003) ‘Plant and Systems Engineering Information modelling – Getting started with EXPRESS – G’.

AIA California Council (2007) ‘Integrated Project Delivery: A Guide’, *American Institute of Architects*, pp. 1–62. doi: 10.1016/j.autcon.2010.09.002.

Aldegeily, M., Hu, Y. and Ph, D. (2018) ‘From Architectural Design to Structural Analysis : A Data- Driven Approach to Study Building Information Modeling (BIM) Interoperability’, *54th ASC Annual international conference proceedings*, pp. 537–545. Available at: http://ascpro0.ascweb.org/archives/cd/2018/paper/CPRT152002018.pdf (accessed 2 May 2021).

America, A. G. C. of (2010) ‘The Contractor’s Guide to BIM’, 2nd editio, p. 48.

Bahar, Y. *et al.* (2013) ‘A Thermal Simulation Tool for Building and Its Interoperability through the Building Information Modeling (BIM) Platform’, *Buildings*, 3(2), pp. 380–398. doi: 10.3390/buildings3020380.

Bavafa, M. (2015) ‘Enhancing Information Quality through Building Information Modelling Implementation within UK Structural Engineering Organisations.’, *PhD thesis, University of Salford*.

Beach, T. *et al.* (2017) ‘Management of Collaborative BIM Data by Federating Distributed BIM Models’, *Journal of Computing in Civil Engineering*, 31(4), p. 04017009. doi: 10.1061/(ASCE)CP.1943-5487.0000657.

Bhatija, V. P., Thomas, N. and Dawood, N. (2017) ‘A Preliminary Approach towards Integrating Knowledge Management with Building Information Modeling (KBIM) for the Construction Industry’, *International Journal of Innovation, Management and Technology*, 8(1), pp. 64–70. doi: 10.18178/ijimt.2017.8.1.704.

BIM Levels explained | NBS (2021). Available at: https://www.thenbs.com/knowledge/bim-levels-explained (Accessed: 13 April 2021).

Boje, C. and Li, H. (2018) ‘Advanced Engineering Informatics Crowd simulation-based knowledge mining supporting building evacuation design’, *Advanced Engineering Informatics*. Elsevier, 37(May), pp. 103–118. doi: 10.1016/j.aei.2018.05.002.

Bosché, F. and Guenet, E. (2014) 'Automating surface flatness control using terrestrial laser scanning and building information models', *Automation in Construction*. Elsevier B.V., 44, pp. 212–226. doi: 10.1016/j.autcon.2014.03.028.

Bradley, A. *et al.* (2016) 'BIM for infrastructure: An overall review and constructor perspective', *Automation in Construction*. Elsevier B.V., 71, pp. 139–152. doi: 10.1016/j.autcon.2016.08.019.

British Standards Institution (2006) 'Concrete — Complementary British Standard to BS EN 206-1', *Part 2: Specification for constituent materials and concrete*, (BS 8500-2:2006).

BuildingSMART (2021a) *BuildingSMART Technical*. Available at: https://technical.buildingsmart.org/ (Accessed: 2 July 2021).

BuildingSMART (2021b) *MVD Database - buildingSMART Technical*. Available at: https://technical.buildingsmart.org/standards/ifc/mvd/mvd-database/ (Accessed: 19 March 2021).

BuildingSMART (2021c) *Software Implementations - buildingSMART Technical*. Available at: https://technical.buildingsmart.org/resources/software-implementations/ (Accessed: 18 March 2021).

buildingSMART - The Home of BIM (2021). Available at: https://www.buildingsmart.org/ (Accessed: 1 April 2021).

Chakraborty, D., Elzarka, H. and Bhatnagar, R. (2016) 'Generation of accurate weather files using a hybrid machine learning methodology for design and analysis of sustainable and resilient buildings', *Sustainable Cities and Society*. Elsevier B.V., 24, pp. 33–41. doi: 10.1016/j.scs.2016.04.009.

Chen, C. *et al.* (2010) 'LCA allocation procedure used as an incitative method for waste recycling: An application to mineral additions in concrete', *Resources, Conservation and Recycling*. Elsevier B.V., 54(12), pp. 1231–1240. doi: 10.1016/j.resconrec.2010.04.001.

Chen, P. H. *et al.* (2005a) 'Implementation of IFC-based web server for collaborative building design between architects and structural engineers', *Automation in Construction*, 14.1, pp. 115–128. doi: 10.1016/j.autcon.2004.08.013.

Chen, P. H. *et al.* (2005b) 'Implementation of IFC-based web server for collaborative building design between architects and structural engineers', *Automation in Construction*, 14(1), pp. 115–128. doi: 10.1016/j.autcon.2004.08.013.

Chen, X. S., Liu, C. C. and Wu, I. C. (2018) 'A BIM-based visualization and warning system for fire rescue', *Advanced Engineering Informatics*. Elsevier, 37(April), pp. 42–53. doi: 10.1016/j.aei.2018.04.015.

Chen, Z., Masood, M. K. and Soh, Y. C. (2016) 'A fusion framework for occupancy estimation in office buildings based on environmental sensor data', *Energy and Buildings*. Elsevier B.V., 133, pp. 790–798. doi: 10.1016/j.enbuild.2016.10.030.

Cheng, C. P. *et al.* (2008) 'Domain-specific ontology mapping by corpus-based semantic similarity', *Proceedings of 2008 NSF CMMI Engineering Research and Innovation Conference*. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.5614&rep=rep1&type=pdf.

Cheung, F. K. T. *et al.* (2012) 'Automation in Construction Early stage multi-level cost estimation for schematic BIM models', *Automation in Construction*. Elsevier B.V., 27, pp. 67–77. doi: 10.1016/j.autcon.2012.05.008.

Chi, H. L., Wang, X. and Jiao, Y. (2015) 'BIM-Enabled Structural Design: Impacts and Future Developments in Structural Modelling, Analysis and Optimisation Processes', *Archives of Computational Methods in Engineering*, 22(1), pp. 135–151. doi: 10.1007/s11831-014-9127-7.

Cho, Y. K. *et al.* (2015) 'A Framework for Cloud-based Energy Evaluation and Management for Sustainable Decision Support in the Built Environments', *Procedia Engineering*. Elsevier B.V., 118, pp. 442–448. doi: 10.1016/j.proeng.2015.08.445.

Choi, J. *et al.* (2016) 'Automation in Construction Development of openBIM-based energy analysis software to improve the interoperability of energy performance assessment', *Automation in Construction*. Elsevier B.V., 72, pp. 52–64. doi: 10.1016/j.autcon.2016.07.004.

Choi, J., Kim, H. and Kim, I. (2015) 'Open BIM-based quantity take-off system for schematic estimation of building frame in early design stage', *Journal of Computational Design and Engineering*. Elsevier, 2(1), pp. 16–25. doi: 10.1016/j.jcde.2014.11.002.

Chong Johnson, S. and Chong Johnson Lim, S. (2015) *ONTOLOGY IN DESIGN ENGINEERING: STATUS AND CHALLENGES*.

Chung, M. T. *et al.* (2016) 'Using Docker in high performance computing applications', in *2016 IEEE 6th International Conference on Communications and Electronics, IEEE ICCE 2016*. Institute of Electrical and Electronics Engineers Inc., pp. 52–57. doi: 10.1109/CCE.2016.7562612.

Ciribini, A. L. C. *et al.* (2017) 'Tracking Users' Behaviors through Real-time Information in BIMs: Workflow for Interconnection in the Brescia Smart Campus Demonstrator', *Procedia Engineering*. The Author(s), 180, pp. 1484–1494. doi: 10.1016/j.proeng.2017.04.311.

Computers and Structures (2013) 'Technical Note IFC4 Import and Export', pp. 1–27. Available at: https://www.csiamerica.com/sites/default/files/pdf/S-TN-IFC-001.pdf (accessed 31 May 2020).

continuumio/miniconda3 - Docker Hub (no date). Available at: https://hub.docker.com/r/continuumio/miniconda3 (Accessed: 29 January 2021).

Corcho, O., Fernández-López, M. and Gómez-Pérez, A. (2003) 'Methodologies, tools and languages for building ontologies. Where is their meeting point?', *Data and Knowledge Engineering*, 46(1), pp. 41–64. doi: 10.1016/S0169-023X(02)00195-7.

Costa, G. and Madrazo, L. (2015) 'Connecting building component catalogues with BIM models using semantic technologies: An application for precast concrete components', *Automation in Construction*. Elsevier B.V., 57, pp. 239–248. doi: 10.1016/j.autcon.2015.05.007.

Dakhli, Z., Lafhaj, Z. and Mossman, A. (2019) 'The Potential of Blockchain in Building Construction', *Buildings*, 9(4), p. 77. doi: 10.3390/buildings9040077.

Das, M., Cheng, J. C. and Kumar, S. S. (2015) 'Social BIMCloud: a distributed cloud-based BIM platform for object-based lifecycle information exchange', *Visualization in Engineering*, 3(1). doi: 10.1186/s40327-015-0022-6.

Dave, B. *et al.* (2016) 'Opportunities for enhanced lean construction management using Internet of Things standards', *Automation in Construction*. Elsevier B.V., 61, pp. 86–97. doi: 10.1016/j.autcon.2015.10.009.

Deng, X. Y. and Chang, T.-Y. P. (2006) 'Creating structural model from IFC-based architectural model', *Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, (Montreal), pp. 3687–3695. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.232.2379&rep=rep1&type=pdf

(accessed 31 March 2020).

Dibley, M. *et al.* (2012) 'An ontology framework for intelligent sensor-based building monitoring', *Automation in Construction*. Elsevier B.V., 28, pp. 1–14. doi: 10.1016/j.autcon.2012.05.018.

Eastman, C. M. *et al.* (2010) 'Exchange model and exchange object concepts for implementation of national BIM standards', *Journal of Computing in Civil Engineering*, 24(1), pp. 25–34. doi: 10.1061/(ASCE)0887-3801(2010)24:1(25).

Eck, V. and Rousseau, R. (2014) *Visualizing bibliometric networks*.

Edmondson, V. *et al.* (2018) 'A smart sewer asset information model to enable an "Internet of Things" for operational wastewater management', *Automation in Construction*. Elsevier, 91(March), pp. 193–205. doi: 10.1016/j.autcon.2018.03.003.

Eurocode 2 (2004) 'Eurocode 2: Design of concrete structures - Part 1-1: General rules and rules for buildings', 1(2005). Available at: https://www.phd.eng.br/wp-content/uploads/2015/12/en.1992.1.1.2004.pdf.

Fang, Y. *et al.* (2016) 'Case Study of BIM and Cloud–Enabled Real-Time RFID Indoor Localization for Construction Management Applications', *Journal of Construction Engineering and Management*, 142(7), p. 05016003. doi: 10.1061/(ASCE)CO.1943-7862.0001125.

Gao, G. *et al.* (2015) 'A query expansion method for retrieving online BIM resources based on Industry Foundation Classes', *Automation in Construction*. Elsevier B.V., 56, pp. 14–25. doi: 10.1016/j.autcon.2015.04.006.

Gao, T. *et al.* (2015) 'An approach to combine progressively captured point clouds for BIM update', *Advanced Engineering Informatics*. Elsevier Ltd, 29(4), pp. 1001–1012. doi: 10.1016/j.aei.2015.08.005.

Gielingh, W. (2008) 'An assessment of the current state of product data technologies', *CAD Computer Aided Design*, 40(7), pp. 750–759. doi: 10.1016/j.cad.2008.06.003.

Gokce, K. U. and Gokce, H. U. (2013) 'IFC View for Bill of Quantities for Software Interoperability', *Proceedings of the 30th CIB W78 International Conference*, pp. 827–836. doi: https://itc.scix.net/pdfs/w78-2013-paper-102.pdf (accessed 2 October 2021).

Grilo, A. and Jardim-Goncalves, R. (2010) 'Value proposition on interoperability of BIM and collaborative working environments', *Automation in Construction*. Elsevier B.V., 19(5), pp. 522–530. doi: 10.1016/j.autcon.2009.11.003.

Grilo, A. and Jardim-Goncalves, R. (2011) 'Challenging electronic procurement in the AEC sector: A BIM-based integrated perspective', *Automation in Construction*. Elsevier B.V., 20(2), pp. 107–114. doi: 10.1016/j.autcon.2010.09.008.

Gui, N. *et al.* (2019) 'IFC-Based Partial Data Model Retrieval for Distributed Collaborative Design', *Journal of Computing in Civil Engineering*, 33(3), pp. 1–10. doi: 10.1061/(ASCE)CP.1943-5487.0000829.

Hammond, G. *et al.* (2011) 'The Inventory of Carbon and Energy (ICE): Embodied energy and carbon in construction materials', *Ice*, p. 136. Available at: http://www.ihsti.com/tempimg/57c152b-ENVIRO2042201160372.pdf%0Awww.bath.ac.uk/mech-eng/sert/embodied%0A.

Han, K., Degol, J. and Golparvar-Fard, M. (2018) 'Geometry- and Appearance-Based Reasoning of Construction Progress Monitoring', *Journal of Construction Engineering and Management*, 144(2), pp. 1–12. doi: 10.1061/(ASCE)CO.1943-7862.0001428.

Hassanien Serror, M. *et al.* (2008) 'Shared computer-aided structural design model for construction industry (infrastructure)’, *CAD Computer Aided Design*, 40(7), pp. 778–788. doi: 10.1016/j.cad.2007.07.003.

Howard, R. (2006) ‘DESCRIBING THE CHANGES IN ARCHITECTURAL INFORMATION FREE-FORM ARCHITECTURAL EXPRESSION’, 11(January), pp. 395–408.

Hu, Z. Z. *et al.* (2016) ‘Improving interoperability between architectural and structural design models: An industry foundation classes-based approach with web-based tools’, *Automation in Construction*. Elsevier B.V., 66, pp. 29–42. doi: 10.1016/j.autcon.2016.02.001.

Hu, Z., Zhang, J. and Deng, Z. (2008) ‘Construction process simulation and safety analysis based on building information model and 4D technology’, *Tsinghua Science and Technology*, 13(S1), pp. 266–272. doi: 10.1016/S1007-0214(08)70160-3.

Iivari, J. and Venable, J. (2009) ‘Action research and design science research - Seemingly similar but decisively dissimilar’, *17th European Conference on Information Systems, ECIS 2009*.

Iqbal, R. *et al.* (2013) ‘An analysis of ontology engineering methodologies: A literature review’, *Research Journal of Applied Sciences, Engineering and Technology*, 6(16), pp. 2993–3000. doi: 10.19026/rjaset.6.3684.

Ismail, B. I. *et al.* (2016) ‘Evaluation of Docker as Edge computing platform’, in *ICOS 2015 - 2015 IEEE Conference on Open Systems*. Institute of Electrical and Electronics Engineers Inc., pp. 130–135. doi: 10.1109/ICOS.2015.7377291.

Jadid, M. N. and Idrees, M. M. (2007) ‘Cost estimation of structural skeleton using an interactive automation algorithm: A conceptual approach’, *Automation in Construction*, 16(6), pp. 797–805. doi: 10.1016/j.autcon.2007.02.007.

Jardim-Goncalves, R. and Grilo, A. (2010) ‘SOA4BIM: Putting the building and construction industry in the Single European Information Space’, *Automation in Construction*. Elsevier B.V., 19(4), pp. 388–397. doi: 10.1016/j.autcon.2009.11.009.

Jiang, C. *et al.* (2016) ‘Indoor occupancy estimation from carbon dioxide concentration’, *Energy and Buildings*. Elsevier B.V., 131, pp. 132–141. doi: 10.1016/j.enbuild.2016.09.002.

Jiao, Y. *et al.* (2013) ‘Automation in Construction Towards cloud Augmented Reality for construction application by BIM and SNS integration’, *Automation in Construction*. Elsevier B.V., 33, pp. 37–47. doi: 10.1016/j.autcon.2012.09.018.

Johannesson, P. and Perjons, E. (2014) *An introduction to design science*, *An Introduction to Design Science*. doi: 10.1007/978-3-319-10632-8.

Kang, K., Lin, J. and Zhang, J. (2018) ‘BIM- and IoT-based monitoring framework for building performance management’, *Journal of Structural Integrity and Maintenance*. Taylor & Francis, 3(4), pp. 254–261. doi: 10.1080/24705314.2018.1536318.

Karan, Ebrahim P and Irizarry, J. (2015) ‘Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services’, *Automation in Construction*. Elsevier B.V., 53, pp. 1–12. doi: 10.1016/j.autcon.2015.02.012.

Karan, Ebrahim P. and Irizarry, J. (2015) ‘Extending BIM interoperability to preconstruction operations using geospatial analyses and semantic web services’, *Automation in Construction*. Elsevier B.V., 53, pp. 1–12. doi: 10.1016/j.autcon.2015.02.012.

König, M., Dirnbek, J. and Stankovski, V. (2013) ‘Architecture of an open knowledge base for sustainable buildings based on Linked Data technologies’, *Automation in Construction*. Elsevier B.V., 35, pp. 542–550. doi: 10.1016/j.autcon.2013.07.002.

Lagüela, S. *et al.* (2013) ‘Automatic thermographic and RGB texture of as-built BIM for energy rehabilitation purposes', *Automation in Construction*. Elsevier B.V., 31, pp. 230–240. doi: 10.1016/j.autcon.2012.12.013.

Lai, H. and Deng, X. (2018) 'Interoperability analysis of ifc-based data exchange between heterogeneous BIM software', *Journal of Civil Engineering and Management*, 24(7), pp. 537–555. doi: 10.3846/jcem.2018.6132.

Lai, H., Zhou, C. and Deng, X. (2019) 'Exchange Requirement-Based Delivery Method of Structural Design Information for Collaborative Design Using Industry Foundation Classes', *Journal of Civil Engineering and Management*, 25(6), pp. 559–573. doi: 10.3846/jcem.2019.9870.

Larrard, F. De and Sedran, T. (2002) 'Mixture-proportioning of high-performance concrete', *Cement and Concrete Research*, 32(11), pp. 1699–1704. doi: 10.1016/S0008-8846(02)00861-X.

Lee, G. (2009) 'Concept-Based Method for Extracting Valid Subsets from an EXPRESS Schema', *Journal of Computing in Civil Engineering*, 23(2), pp. 128–135. doi: 10.1061/(asce)0887-3801(2009)23:2(128).

Lee, G. *et al.* (2011) 'Metrics for Quantifying the Similarities and Differences between IFC Files', *Journal of Computing in Civil Engineering*, 25(2), pp. 172–181. doi: 10.1061/(asce)cp.1943-5487.0000077.

Lee, G. (2011) 'What information can or cannot be exchanged?', *Journal of Computing in Civil Engineering*, 25(1), pp. 1–9. doi: 10.1061/(ASCE)CP.1943-5487.0000062.

Lee, G., Park, Y. H. and Ham, S. (2013) 'Extended Process to Product Modeling (xPPM) for integrated and seamless IDM and MVD development', *Advanced Engineering Informatics*. Elsevier Ltd, 27(4), pp. 636–651. doi: 10.1016/j.aei.2013.08.004.

Lee, I. (2017) 'Big data: Dimensions, evolution, impacts, and challenges', *Business Horizons*. 'Kelley School of Business, Indiana University', 60(3), pp. 293–303. doi: 10.1016/j.bushor.2017.01.004.

Lee, S. K., Kim, K. R. and Yu, J. H. (2014) 'BIM and ontology-based approach for building cost estimation', *Automation in Construction*. Elsevier B.V., 41, pp. 96–105. doi: 10.1016/j.autcon.2013.10.020.

Lee, Y. C., Eastman, C. M. and Solihin, W. (2016) 'An ontology-based approach for developing data exchange requirements and model views of building information modeling', *Advanced Engineering Informatics*, 30(3), pp. 354–367. doi: 10.1016/j.aei.2016.04.008.

Li, C. Z. *et al.* (2018) 'An Internet of Things-enabled BIM platform for on-site assembly services in prefabricated construction', *Automation in Construction*, 89(February), pp. 146–161. doi: 10.1016/j.autcon.2018.01.001.

Li, J., Greenwood, D. and Kassem, M. (2019) 'Blockchain in the built environment and construction industry: A systematic review, conceptual models and practical use cases', *Automation in Construction*. Elsevier, 102(August 2018), pp. 288–307. doi: 10.1016/j.autcon.2019.02.005.

Li, X. *et al.* (2017) 'Mapping the knowledge domains of Building Information Modeling (BIM): A bibliometric approach', *Automation in Construction*, pp. 195–206. doi: 10.1016/j.autcon.2017.09.011.

Lim, C. H., Yoon, Y. S. and Kim, J. H. (2004) 'Genetic algorithm in mix proportioning of high-performance concrete', *Cement and Concrete Research*, 34(3), pp. 409–420. doi: 10.1016/j.cemconres.2003.08.018.

Liu, H., Lu, M. and Al-Hussein, M. (2016) 'Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry', *Advanced Engineering Informatics*. Elsevier Ltd, 30(2), pp. 190–207. doi: 10.1016/j.aei.2016.03.001.

Liu, J. *et al.* (2018) 'Dimensional accuracy and structural performance assessment of spatial structure components using 3D laser scanning', *Automation in Construction*. Elsevier, 96(October), pp. 324–336. doi: 10.1016/j.autcon.2018.09.026.

Liu, Y. (2013) 'Consistency checking based on ontology of design information', *Applied Mechanics and Materials*, 438–439(1978), pp. 1992–1997. doi: 10.4028/www.scientific.net/AMM.438-439.1992.

Liu, Z.-Q., Li, Y.-G. and Zhang, H.-Y. (2010) 'IFC-based integration tool for supporting information exchange from architectural model to structural model', *J. Cent. South Univ. Technol*, 17, pp. 1344–1350. doi: 10.1007/s11771−010−0640−z.

Manso, M. Á., Wachowicz, M. and Bernabé, M. Á. (2009) 'Towards an integrated model of interoperability for spatial data infrastructures', *Transactions in GIS*, 13(1), pp. 43–67. doi: 10.1111/j.1467-9671.2009.01143.x.

Marasco, D. E. and Kontokosta, C. E. (2016) 'Applications of machine learning methods to identifying and predicting building retrofit opportunities', *Energy and Buildings*. Elsevier B.V., 128, pp. 431–441. doi: 10.1016/j.enbuild.2016.06.092.

McArthur, J. J. *et al.* (2018) 'Machine learning and BIM visualization for maintenance issue classification and enhanced data collection', *Advanced Engineering Informatics*. Elsevier, 38(June), pp. 101–112. doi: 10.1016/j.aei.2018.06.007.

McGlinn, K. *et al.* (2017) 'Usability evaluation of a web-based tool for supporting holistic building energy management', *Automation in Construction*. Elsevier, 84(September), pp. 154–165. doi: 10.1016/j.autcon.2017.08.033.

Mill, T., Alt, A. and Liias, R. (2013) 'Combined 3D Building Surveying Techniques – Terrestrial Laser Scanning (Tls) and Total Station Surveying for Bim Data Management Purposes', *Journal of Civil Engineering and Management*. Taylor & Francis, 19(Supplement_1), pp. S23–S32. doi: 10.3846/13923730.2013.795187.

Mockienė, J., Keras, V. and Gilys, A. (2015) 'Problems and rudimental tends on implementation of European Union's directives on Sustainable building', *Journal of Sustainable Architecture and Civil Engineering*, 10(1), pp. 87–98. doi: 10.5755/j01.sace.10.1.8165.

Modelling, B. I. and Plan, S. (2015) 'Digital Built Britain Level 3 Building Information Modelling - Strategic Plan', (February), pp. 1–47.

Müller, H. S., Haist, M. and Vogel, M. (2014) 'Assessment of the sustainability potential of concrete and concrete structures considering their environmental impact, performance and lifetime', *Construction and Building Materials*, 67(PART C), pp. 321–337. doi: 10.1016/j.conbuildmat.2014.01.039.

Munkley, J., Kassem, M. and Dawood, N. (2016) 'Synchronous BIM Collaboration in the Cloud : benefits and challenges from the implementation of a bespoke solution'. Available at: https://research.tees.ac.uk/ws/portalfiles/portal/4180928/595714.pdf (accessed 2 October 2021).

Nawari O Nawari, K. B. and Ravindran, S. (2019) 'Blockchain technology and BIM process: review and potential applications', *Journal of Information Technology in Construction (ITcon)*, 24(October 2018), pp. 209–238. Available at: http://www.itcon.org/2019/12.

Nepal, M. P. *et al.* (2013) 'Ontology-Based Feature Modeling for Construction Information

Extraction from a Building Information Model', *Journal of Computing in Civil Engineering*, 27.5, pp. 555–569. doi: 10.1061/(ASCE)CP.1943-5487.0000230.

Nguyen, T. H. and Kim, J. L. (2011) 'Building code compliance checking using BIM technology', *Proceedings - Winter Simulation Conference*. IEEE, pp. 3395–3400. doi: 10.1109/WSC.2011.6148035.

Niehaves, B., Ortbach, K. and Tavakoli, A. (2012) 'On the relationship between the IT artifact and design theory: The case of virtual social facilitation', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 354–370. doi: 10.1007/978-3-642-29863-9_26.

Niknam, M. and Karshenas, S. (2017) 'A shared ontology approach to semantic representation of BIM data', *Automation in Construction*. Elsevier B.V., 80, pp. 22–36. doi: 10.1016/j.autcon.2017.03.013.

NIST (2011) *IFC File Analyzer, National Institute of Standards and Technology (NIST)*. Available at: https://www.nist.gov/services-resources/software/ifc-file-analyzer (Accessed: 25 February 2021).

Noy, N. F. and McGuinness, D. L. (2001) 'Ontology Development 101: A Guide to Creating Your First Ontology', *Stanford Knowledge Systems Laboratory*, p. 25. doi: 10.1016/j.artmed.2004.01.014.

openBIM - buildingSMART International (no date). Available at: https://www.buildingsmart.org/about/openbim/ (Accessed: 25 March 2021).

Oraee, M. *et al.* (2018) 'Bibliometric Analysis of Published Studies in ASCE Construction Research', (April). doi: 10.1061/9780784481301.002.

Østergård, T., Jensen, R. L. and Maagaard, S. E. (2016) 'Building simulations supporting decision making in early design - A review', *Renewable and Sustainable Energy Reviews*. Elsevier, 61, pp. 187–201. doi: 10.1016/j.rser.2016.03.045.

Pasini, D. (2018) 'Connecting BIM and IoT for addressing user awareness toward energy savings', *Journal of Structural Integrity and Maintenance*. Taylor & Francis, 3(4), pp. 243–253. doi: 10.1080/24705314.2018.1535235.

Pauwels, P., Zhang, S. and Lee, Y. C. (2017) 'Semantic web technologies in AEC industry: A literature overview', *Automation in Construction*. Elsevier B.V., 73, pp. 145–165. doi: 10.1016/j.autcon.2016.10.003.

Paviot, T. *et al.* (2011) 'A generic multiCAD/multiPDM interoperability framework', *International Journal of Services Operations and Informatics*, 6, pp. 124–137.

Peffers, K. *et al.* (2007) 'A design science research methodology for information systems research', *Journal of Management Information Systems*, 24(3), pp. 45–77. doi: 10.2753/MIS0742-1222240302.

Petri, I. *et al.* (2017) 'Coordinating multi-site construction projects using federated clouds', *Automation in Construction*. Elsevier, 83(August), pp. 273–284. doi: 10.1016/j.autcon.2017.08.011.

Pətrəucean, V. *et al.* (2015) 'State of research in automatic as-built modelling', *Advanced Engineering Informatics*, 29(2), pp. 162–171. doi: 10.1016/j.aei.2015.01.001.

Preidel, C. *et al.* (2018) 'Common data environment', *Building Information Modeling: Technology Foundations and Industry Practice*, pp. 279–291. doi: 10.1007/978-3-319-92862-3_15.

Qin, L., Deng, X. Y. and Liu, X. La (2011) 'Industry foundation classes based integration of architectural design and structural analysis', *Journal of Shanghai Jiaotong University (Science)*, 16(1), pp. 83–90. doi: 10.1007/s12204-011-1099-2.

Ramaji, I. J. (2016) 'An Integrated Building Information Modeling (BIM) Framework For Multi-story Modular Buildings', *The Pennsylvania State University*, (May). Available at: https://etda.libraries.psu.edu/files/final_submissions/11685 (accessed 31 May 2020).

Ramaji, I. J. and Memari, A. M. (2016) 'Interpreted Information Exchange: Systematic Approach for BIM to Engineering Analysis Information Transformations', *Journal of Computing in Civil Engineering*, 142(10), pp. 1–14. doi: 10.1061/(ASCE)CO.1943-7862.0001159.

Ramaji, I. J. and Memari, A. M. (2018) 'Interpretation of structural analytical models from the coordination view in building information models', *Automation in Construction*. Elsevier, 90(February), pp. 117–133. doi: 10.1016/j.autcon.2018.02.025.

Ramaji, I. J. and Memari, A. M. (2020) 'Interpreted information exchange: Implementation point of view', *Journal of Information Technology in Construction*, 25(December 2017), pp. 123–139. doi: 10.36680/j.itcon.2020.008.

Redmond, A. *et al.* (2012) 'Exploring how information exchanges can be enhanced through Cloud BIM', *Automation in Construction*. Elsevier B.V., 24, pp. 175–183. doi: 10.1016/j.autcon.2012.02.003.

Ren, G., Ding, R. and Li, H. (2019) 'Building an ontological knowledgebase for bridge maintenance', *Advances in Engineering Software*. Elsevier, 130(July 2018), pp. 24–40. doi: 10.1016/j.advengsoft.2019.02.001.

Rezgui, Y. (2007) 'Exploring virtual team-working effectiveness in the construction sector', *Interacting with Computers*, 19(1), pp. 96–112. doi: 10.1016/j.intcom.2006.07.002.

RIBA (2020) 'The Structural Plan of Work 2020 : Overview and Guidance', (June).

Rokooei, S. (2015) 'Building Information Modeling in Project Management: Necessities, Challenges and Outcomes', *Procedia - Social and Behavioral Sciences*. Elsevier B.V., 210(February), pp. 87–95. doi: 10.1016/j.sbspro.2015.11.332.

Ryu, S. H. and Moon, H. J. (2016) 'Development of an occupancy prediction model using indoor environmental data based on machine learning techniques', *Building and Environment*. Elsevier Ltd, 107, pp. 1–9. doi: 10.1016/j.buildenv.2016.06.039.

Santos, R., Costa, A. A. and Grilo, A. (2017) 'Automation in Construction Bibliometric analysis and review of Building Information Modelling literature published between 2005 and 2015', 80, pp. 118–136. doi: 10.1016/j.autcon.2017.03.005.

Saunders, M., Lewis, P. and Thornhill, A. (2015) 'Understanding research philosophy and approaches to theory development Learning', (January 2009), pp. 122–161. Available at: https://www.researchgate.net/publication/309102603_Understanding_research_philosophies_and_approaches.

Schapke, S.-E. *et al.* (2018) 'Collaborative Data Management', in *Building Information Modeling*. Cham: Springer International Publishing, pp. 251–277. doi: 10.1007/978-3-319-92862-3_14.

Sherif, A., Jinkook, L. E. E. and Chuck, E. (2011) 'Automated Cost Analysis of Concept Design BIM Models', *Proceedings of the 14th International Conference on Computer Aided Architectural Design Futures*, pp. 403–418. Available at: ISBN 9782874561429.

Shourangiz, E. *et al.* (2011) 'Flexibility of BIM towards Design Change', *Proceedings of 2nd International Conference on Construction and Project Management*, 15, pp. 79–83.

Sørensen, K. B., Christiansson, P. and Svidt, K. (2010) 'Ontologies to support RFID-based link between virtual models and construction components', *Computer-Aided Civil and Infrastructure Engineering*, 25(4), pp. 285–302. doi: 10.1111/j.1467-8667.2009.00638.x.

Steel, J., Drogemuller, R. and Toth, B. (2012) 'Model interoperability in building information modelling', *Software & Systems Modeling*, 11(1), pp. 99–109. doi: 10.1007/s10270-010-0178-4.

Succar, B. (2009) 'Building information modelling framework : A research and delivery foundation for industry stakeholders', *Automation in Construction*. Elsevier B.V., 18(3), pp. 357–375. doi: 10.1016/j.autcon.2008.10.003.

Sun, C. *et al.* (2017) 'A literature review of the factors limiting the application of BIM in the construction industry', *Technological and Economic Development of Economy*, 23(5), pp. 764–779. doi: 10.3846/20294913.2015.1087071.

Tamke, M. *et al.* (2016) 'Tracking Changes in Buildings over Time - Fully Automated Reconstruction and Difference Detection of 3d Scan and BIM files', *34th eCAADe Conference*, 2, pp. 643–651. Available at: http://papers.cumincad.org/cgi-bin/works/Show?ecaade2016_147.

Tan, K. (2018) 'The Framework of Combining Artificial Intelligence and Construction 3D Printing in Civil Engineering', *MATEC Web of Conferences*, 206, p. 01008. doi: 10.1051/matecconf/201820601008.

Taylor, C. *et al.* (2009) 'Final Report Collaboration Platform', *Architectural Design*, (Final), p. 144. Available at: http://www.construction-innovation.info/images/pdfs/2._Final_Report__Edited__11.08.09.pdf.

Tixier, A. J. P. *et al.* (2017) 'Construction Safety Clash Detection: Identifying Safety Incompatibilities among Fundamental Attributes using Data Mining', *Automation in Construction*. Elsevier B.V., 74, pp. 39–54. doi: 10.1016/j.autcon.2016.11.001.

Turk, Ž. (2020) 'Interoperability in Construction – Mission Impossible ?', *Developments in the Built Environment*. Elsevier Ltd, p. 100018. doi: 10.1016/j.dibe.2020.100018.

Turk, Ž. and Klinc, R. (2017) 'Potentials of Blockchain Technology for Construction Management', *Procedia Engineering*. The Author(s), 196(June), pp. 638–645. doi: 10.1016/j.proeng.2017.08.052.

UK BIM Alliance (2016) 'BIM in the UK : Past , Present & Future'. Available at: https://docplayer.net/31610196-Bim-in-the-uk-past-present-future.html (accessed 2 May 2021).

Venugopal, M. *et al.* (2012) 'Semantics of model views for information exchanges using the industry foundation class schema', *Advanced Engineering Informatics*. doi: 10.1016/j.aei.2012.01.005.

Venugopal, M., Eastman, Charles M. and Teizer, J. (2015a) 'An ontology-based analysis of the industry foundation class schema for building information model exchanges', *Advanced Engineering Informatics*. Elsevier Ltd, 29(4), pp. 940–957. doi: 10.1016/j.aei.2015.09.006.

Venugopal, M., Eastman, Charles M and Teizer, J. (2015) 'An ontology-based analysis of the industry foundation class schema for building information model exchanges', *Advanced Engineering Informatics*. Elsevier Ltd, 29(4), pp. 940–957. doi: 10.1016/j.aei.2015.09.006.

Venugopal, M., Eastman, Charles M. and Teizer, J. (2015b) 'An ontology-based analysis of the industry foundation class schema for building information model exchanges', *Advanced Engineering Informatics*. Elsevier Ltd, 29(4), pp. 940–957. doi: 10.1016/j.aei.2015.09.006.

Venugopal, M., Eastman, Charles M. and Teizer, J. (2015c) 'An ontology-based analysis of the industry foundation class schema for building information model exchanges', *Advanced Engineering Informatics*. Elsevier Ltd, 29(4), pp. 940–957. doi: 10.1016/j.aei.2015.09.006.

VOSviewer - Visualizing scientific landscapes (2021). Available at: https://www.vosviewer.com/ (Accessed: 11 January 2021).

Wan, C., Chen, P. and Tiong, R. L. K. (2004) 'Assessment of IFC for structural analysis domain', *ITcon*, 9(May), pp. 75–95. Available at: http://www.itcon.org/2004/5.

Wang, C. and Cho, Y. K. (2015) 'Performance Evaluation of Automatically Generated BIM from Laser Scanner Data for Sustainability Analyses', *Procedia Engineering*. Elsevier B.V., 118, pp. 918–925. doi: 10.1016/j.proeng.2015.08.531.

Wang, H. H., Boukamp, F. and Elghamrawy, T. (2011) 'Ontology-based approach to context representation and reasoning for managing context-sensitive construction information', *Journal of Computing in Civil Engineering*, 25(5), pp. 331–346. doi: 10.1061/(ASCE)CP.1943-5487.0000094.

Wang, Q. and Kim, M. K. (2019) 'Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018', *Advanced Engineering Informatics*. Elsevier, 39(September 2018), pp. 306–319. doi: 10.1016/j.aei.2019.02.007.

Wang, X., Yang, H. and Zhang, Q. L. (2015) 'Research of the IFC-based Transformation Methods of Geometry Information for Structural Elements', *Journal of Intelligent and Robotic Systems: Theory and Applications*, 79(3–4), pp. 465–473. doi: 10.1007/s10846-014-0111-0.

Wang, Y. R., Yu, C. Y. and Chan, H. H. (2012) 'Predicting construction cost and schedule success using artificial neural networks ensemble and support vector machines classification models', *International Journal of Project Management*. Elsevier Ltd and IPMA, 30(4), pp. 470–478. doi: 10.1016/j.ijproman.2011.09.002.

Wetzel, E. M. and Thabet, W. Y. (2015) 'The use of a BIM-based framework to support safe facility management processes', *Automation in Construction*. Elsevier B.V., 60, pp. 12–24. doi: 10.1016/j.autcon.2015.09.004.

Won, J. *et al.* (2013) 'No-Schema Algorithm for Extracting a Partial Model from an IFC Instance Model', 27(December), pp. 585–592. doi: 10.1061/(ASCE)CP.1943-5487.0000320.

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

Wu, W. *et al.* (2015) 'Improving Data Center Energy Efficiency Using a Cyber-physical Systems Approach: Integration of Building Information Modeling and Wireless Sensor Networks', *Procedia Engineering*. Elsevier B.V., 118, pp. 1266–1273. doi: 10.1016/j.proeng.2015.08.481.

Xu, S., Liu, K. and Tang, L. C. M. (2013) 'Cost estimation in building information models', *ICCREM 2013: Construction and Operation in the Context of Sustainability - Proceedings of the 2013 International Conference on Construction and Real Estate Management*, (October), pp. 555–566. doi: 10.1061/9780784413135.053.

Yaman, H. and Taş, E. (2007) 'A building cost estimation model based on functional elements', *ITU A| Z*, 4.1, pp. 73–87. Available at: https://jag.journalagent.com/itujfa/pdfs/itujfa-17037-theory_articles-yaman.pdf (accessed 2 October 2021).

Yang, K. H., Song, J. K. and Song, K. Il (2013) 'Assessment of CO 2 reduction of alkali-activated concrete', *Journal of Cleaner Production*. Elsevier Ltd, 39, pp. 265–272. doi: 10.1016/j.jclepro.2012.08.001.

Yuan, L., Guo, J. and Wang, Q. (2020) 'Automatic classification of common building materials from 3D terrestrial laser scan data', *Automation in Construction*. Elsevier, 110(October 2019), p. 103017. doi: 10.1016/j.autcon.2019.103017.

Zhang, J. *et al.* (2018) 'An ontology-based approach supporting holistic structural design with the consideration of safety, environmental impact and cost', *Advances in Engineering Software*. Elsevier Ltd, 115, pp. 26–39. doi: 10.1016/j.advengsoft.2017.08.010.

Zhang, L. and El-gohary, N. M. (2020) 'Automated IFC-based building information modelling and extraction for supporting value analysis of buildings', *International Journal of Construction Management*. Taylor & Francis, 20(4), pp. 269–288. doi: 10.1080/15623599.2018.1484850.

Zhang, L. and Issa, R. R. A. (2013) 'Ontology-Based Partial Building Information Model Extraction', *Journal of Computing in Civil Engineering*, 27(6), pp. 576–584. doi: 10.1061/(ASCE)CP.1943-5487.0000277.

Zhao, X. (2017) 'A scientometric review of global BIM research: Analysis and visualization', *Automation in Construction*. Elsevier, 80(April), pp. 37–47. doi: 10.1016/j.autcon.2017.04.002.

Zheng, R. *et al.* (2019) 'BcBIM: A Blockchain-Based Big Data Model for BIM Modification Audit and Provenance in Mobile Cloud', *Mathematical Problems in Engineering*, 2019. doi: 10.1155/2019/5349538.

Zhong, B. *et al.* (2018) 'Ontology-based framework for building environmental monitoring and compliance checking under BIM environment', *Building and Environment*. Elsevier, 141(December 2017), pp. 127–142. doi: 10.1016/j.buildenv.2018.05.046.

Zhong, R. Y. *et al.* (2017) 'Prefabricated construction enabled by the Internet-of-Things', *Automation in Construction*. Elsevier B.V., 76, pp. 59–70. doi: 10.1016/j.autcon.2017.01.006.

Zou, X., Yue, W. L. and Vu, H. Le (2018) 'Visualization and analysis of mapping knowledge domain of road safety studies', *Accident Analysis and Prevention*. Elsevier, 118(March), pp. 131–145. doi: 10.1016/j.aap.2018.06.010.

Appendix A: Additional information exchange requirements based on a student MSc project conducted at Cardiff University and a sample of the proposed MCDS

Information needed during the architectural and structural design of a hospital building

	Concept design (Design Philosophy)	Design idea /requirements	`
	site analysis	Location site access Surrounding infrastructure Form and massing Ventilation noise current site condition Fire and Evacuation	
Architectural design	Design response	Master planning	Orientation Ease of access and navigation Green Space Form and massing Exterior design
		Building Design Approach	Functional Content Departmental Relationships Operational policies Floor plans Flows
		Detailed Planning	Day Treatment Unit Reception and waiting area Chemotherapy Multi-chair bay Single room with ensuite facility Single person office
	Evidence-based design ideas for a therapeutic environment	General design guidance for healthcare buildings	
Structural Design	Conceptual design (Building Philosophy)	Structural System Think about materials Provide initial structural elements	
	Design Approach, Codes and Software Used	Structural Layout	
		Loading - code	DL, LL, snow L & wind L
	Structural design	Design Process Codes and Procedure Superstructure design	Core Shear wall internal– Table A Steel Bridge Design - Table B Internal/corner Waffle Slab - Table C Column – Table D
		Substructure design	Pile Group Design Table E Retaining Wall Table F

Table A - Core Shear wall internal

Superstructure	Core Shear wall (input data)	factors	Concrete	Partial safety factor
			steel	Partial safety factor Long term effect factor
		Material properties	Concrete	Compressive crushing strength Mean tensile strength Design compressive crushing strength Density Design modulus of elasticity
			Steel	Yield strength Design yield strength of reinforcement Design modulus of elasticity Design yield strain
		Geometry	Building geometry	Block height Block width Height of storey

		Width of storey
		Depth of storey
Loading	Shear wall	Length
		Depth
		Width
	Loaded area	Length of loaded face
		Height of loaded face
		Loaded area per shear core
	Applicability	Dynamic augmentation factor
		Applicability Check
	Wind speed	Basic wind speed
		Site altitude
		Direction factor
		Seasonal factor
		Probability factor
		Site wind speed
		Terrain and building factor
		Effective wind speed
	External wind pressure (windward)	Dynamic pressure
		External pressure coefficient (windward)
		Size effect factor
		External surface pressure
	External wind pressure (leeward)	External pressure coefficient (leeward)
		External surface pressure (leeward)
	Internal wind pressure	Volume of storey
		Diagonal dimension
		Size effect factor for internal pressures
		Internal pressure coefficient
		Internal surface pressure
	Net surface pressure	Net surface pressure (windward)
		Net surface pressure (leeward)
	Net surface loads	Net surface load (windward)
		Net surface load (leeward)
	Overall load	Overall load
Shear Reinforcement Design & check		

Table B - Steel Bridge Design

superstructure	Steel Bridge Design	factors		Permanent load factor
				Imposed load factor (worst case)
				Serviceability safety factor
				Partial safety factors
				Resistance of members and cross-sections
				Shear resistance
		Material properties	General	Yield strength
				Modulus of elasticity
				Shear modulus
				Density
				Design strength
				Poisson's ratio in elastic range
				Ultimate strength
				Yield strength (Characteristic)
		Geometry	section	Assume section
				Plastic modulus
				Second moment of area
			Bridge deck	Width
				Length
				thickness
			Beams	Web thickness
				Depth of section
				Width of section
				Flange thickness
				Area of section
				Root radius
				Depth between fillets
				Shear area
				Assumed stiff bearing length
		Loading	Effective area	Bridge Deck
			Imposed loading	Live Load on floor area
			Permanent loading	Bridge deck self-weight
				Beam self-weight
			Total loading	Total load on beam
		Design		

Table C – requirements for corner or internal waffle slab are the same. However, the design might vary.

superstructure	Internal Waffle Slab	factors	Concrete	Partial safety factor
			steel	Partial safety factor
				Long term effect factor
			Load	Permanent load factor
				Imposed load factor(worst case)
		Material properties	Concrete	Compressive crushing strength
				Mean tensile strength
				Design compressive crushing strength
				Density
				Design modulus of elasticity
			Steel	Yield strength
				Design yield strength of reinforcement
				Design modulus of elasticity
				Ultimate strength
		Geometry	Slab	Short span (x direction)
				Long span (y direction)
				Assumed rib spacing, centre to centre (same in both x and y direction)
				Assumed rib width
				Structural slab depth
				Depth of topping
			Cover	Assumed reinforcement diameter
				Minimum cover due to bond requirement
				Minimum cover due to environmental conditions
				Additive safety element
				Reduction of minimum cover for use of stainless steel
				Reduction of minimum cover for use of additional protection
				Assume 90-minute fire resistance
				Minimum slab thickness
				Minimum fire cover
				Minimum cover
				Deviation cover
				Nominal cover
				Effective depth to tension bar
				Depth to neutral axis
		Loading	Imposed loading	Live Load on floor area
			Permanent loading	Slab self-weight
				Permanent loading due to superimposed dead load (Finishes, screed, insulation etc)
				Total permanent load
			Total loading	Load per unit area
		Design		

Table D - Columns

superstructure	Column	factors	Slenderness	Partial safety factor
			Concrete	Partial safety factor
				Long term effect factor
				Relative flexibility of rotational restraints:
		Material properties	Concrete	Compressive crushing strength
				Mean tensile strength
				Design compressive crushing strength
				Density
				Design modulus of elasticity
			Steel	Yield strength
				Partial factor for Persistent and Transient load
				Design yield strength of reinforcement
				Design modulus of elasticity
				Design yield strain
		Geometry	column	Column height
				Depth
				Width
				Long span of supported slab
				Short span of supported slab
				Assumed diameter of longitudinal steel
				Assumed diameter of links
				Minimum cover due to bond requirement
				Minimum cover due to environmental conditions
				Additive safety element

				Reduction of minimum cover for use of stainless steel
				Reduction of minimum cover for use of additional protection
				Assume 90-minute fire resistance
				Minimum slab thickness
				Minimum fire cover
				Minimum cover
				Deviation cover
				Nominal cover
				Assumed area of longitudinal steel
				Concrete area
				Effective length
			Load factors	Imposed loading
				Permanent loading
			Effective area	Effective area of loading on column
			Imposed loading	Imposed load on floor area
				Imposed load on roof
				Snow zone number
				Site altitude
				Snow loading on ground
			Snow loading	Exposure coefficient
		Loading		Thermal coefficient
				Shape coefficient
				Snow loading on roof
				Snow loading factor
				Slab self-weight
			Permanent loading	Permanent loading due to superimposed dead load (Finishes, screed, insulation etc)
				Total permanent load
			Total loading	Load per unit area
			Column loading	Snow loading on roof
				Design value of the applied axial force
		Design		

Table E - Pile Group Design

		factors	Total group pile efficiency
			Depth of sand
		Ground layout	Depth of clay
			Depth of bedrock
			Water table level
			Ground level
			Bottom of Basement level
		Geometry	Assumed pile diameter
			Area of pile base
Substructure design	Pile Design		Surface area in sand
			Surface area in clay
			Number of piles under block B
			Applied compressive stress
		Applied stress	Total pile area (cross-sectional)
			Applied compressive stress
		Bedrock resistance	Sandstone uniaxial compressive strength
			Factored bedrock resistance to account for block failure
		Resistance check	Utilisation Factor
			Resistance Check

Table F - Retaining Wall

		factors	Permanent load safety factor
		Material properties	Unit weight of soil
			Unit weight of water
			Angle of internal friction
			Water table level
			Ground level
			Bottom of Basement level
Substructure design	Retaining Wall Design		Height of exposed retaining wall
		Geometry	Height of retaining wall above water table
			Height of retaining wall below water table
			Assumed value of depth below ground
			Assumed anchor depth below ground level
			Assumed length of anchor rod per tie
		Retaining wall design	Applied compressive stress
			Total pile area (cross-sectional)
			Applied compressive stress

MCDS defined based on CDA

Data item	Data item representation in IFC	Attribute	Attribute representation in IFC	Type
Project details	IfcProject	Name	-	String
		Identity(GlobalId)	-	String
		Owner History	IfcOwnerHistory	String
		Measuring Unit	IfcUnitAssignment	Integer / Float
		Description	-	String
		Phase	-	String
Site	IfcSite	Name	-	String
		Identity(GlobalId)	-	String
		Owner History	IfcOwnerHistory	String
		Description	-	String
		Site address	-	String
		Elevation	-	Integer / Float
Building	IfcBuilding	Name	-	String
		Identity(GlobalId)	-	String
		Owner History	IfcOwnerHistory	String
		Building address	-	String
Storey levels	IfcBuildingStorey	Name	-	String
		Identity(GlobalId)	-	String
		Owner History	IfcOwnerHistory	String
		Elevation	-	Integer / Float
Spaces	IfcSpace	Name	-	String
		Identity(GlobalId)	-	String
		Owner History	IfcOwnerHistory	String
Column	IfcColumn	Name	-	String
		Identity(GlobalId)	-	String
		Owner History	IfcOwnerHistory	String
		Object Type	-	String
		Representation	IfcProductDefinitionShape	String
		Tag	-	string
		Predefined Type	-	String
Wall	IfcWall	Name	-	String
		Identity(GlobalId)	-	String
		Owner History	IfcOwnerHistory	String
		Object Type	-	String
		Representation	IfcProductDefinitionShape	String
		Tag	-	string
		Predefined Type	-	String
Beam	IfcBeam	Name	-	String
		Identity(GlobalId)	-	String
		Owner History	IfcOwnerHistory	String
		Object Type	-	String
		Representation	IfcProductDefinitionShape	String
		Tag	-	string
		Predefined Type	-	String
Slab	IfcSlab	Name	-	String
		Identity(GlobalId)	-	String
		Owner History	IfcOwnerHistory	String
		Object Type	-	String
		Representation	IfcProductDefinitionShape	String
		Tag	-	string
		Predefined Type	-	String

Appendix B: Tools and Libraries supporting IFC schema

Name	Open-source	Functions of free IFC components	Language	Support IFC4
IfcEngine	Partly	A STEP Toolbox with the ability to generate 3D geometry	C++, C#	√
IFC-SDK	X	Used for reading and writing IFC files	C++	√
IFCOpenShell	√	Used to edit or add new content to an IFC file	Java, python	√
PythonOCC-Core	√	Python OpenCascade provides 3D modelling	python	√
IfcPlusPlus	√	Used for reading and writing IFC files in STEP format	C++	√
BIMserver.org	√	Enables users to store and manage information	Java	√
IFCSchemaReader	√	Used to parse the IFC file	Python	X
ST-Developer	X	Read, write, create, and modify IFC file defined by EXPRESS	C & C++	√
IFC Quick Browser	√	To text-browse large IFC files	C++	X
IfcKit	√	A toolkit for implementing IFC, and the IFC schema	C#	√
IFC File Analyze	√	Useful in helping the extraction of different information from an IFC-based BIM model.	Tcl (Tool Command Language)	√
XBim	Partly	Used to read, create, and view Building Information (BIM) Models in the IFC format. It uses OpenCascade	C++, C#, .NET	√
Eurostep IFC Toolbox	√	Provides pure object-oriented programming methodology to access IFC data.	C++	X
Apstex IFC Framework	√	provides tools for accessing and visualizing IFC-based BIM	Java	√

Appendix C: Data extraction tool with GUI application

```
# imported libraries
import os
from tkinter import *
from tkinter import ttk
from tkinter import filedialog,
import ifcopenshell
import tkinter.messagebox
import ifcopenshell.geom
import tkinter.messagebox
import webbrowser

# main window parameters, menubar and drop down menu

window = Tk()
statusbar = Label(window, text='Welcome to data extraction Engine', bd=1,
relief=SUNKEN, anchor=W)
statusbar.pack(side=BOTTOM, fill=X)
menubar = Menu(window)
window.config(menu=menubar)
submenu = Menu(menubar)
menubar.add_cascade(label='File', menu=submenu)
Helpmenu = Menu(menubar)
menubar.add_cascade(label='Help', menu=Helpmenu)  # a drop down menu

# open file / exist file / about us command

def fileopen():
    for widget in rightFrame_bottom.winfo_children():
        widget.destroy()

    global filename
    filename = filedialog.askopenfilename(initialdir='/', title='select
file',
                                          filetypes=(('ifc', '*.ifc'),
('all files', '*.*'), ('executables', '*.exe')))
    apps.append(os.path.basename(filename))
    print(filename)
    for app in apps:
        label1 = ttk.Label(rightFrame_bottom, text=app)
        label1.pack()

submenu.add_command(label='Import File', command=fileopen)
submenu.add_command(label="Save as")
submenu.add_separator()
submenu.add_command(label='Quit', command=window.destroy)

def about_us():
    tkinter.messagebox.showinfo('Data extraction Engine', 'Produced by: ALI
- supports ifc4')

Helpmenu.add_command(label="About us", command=about_us)

# -------------------------------(tkinter window setup)--------------------
-------------
window.title('Data extraction Engine')
window.geometry('500x500')
```

```
apps = []

# (frames)--------

leftFrame = Frame(window, relief=RAISED, borderwidth=1)
leftFrame.pack(side=LEFT, fill='both')

# left frame is divided into top mid and buttom frame
leftFrame_top = Frame(leftFrame, relief=RAISED, borderwidth=1)
leftFrame_top.pack(fill='both')
leftFrame_Mid = Frame(leftFrame, relief=RAISED, borderwidth=1)
leftFrame_Mid.pack(fill='both')
leftFrame_Mid2 = Frame(leftFrame, relief=RAISED, borderwidth=1)
leftFrame_Mid2.pack(fill='both')
leftFrame_Mid3 = Frame(leftFrame, relief=RAISED, borderwidth=1)
leftFrame_Mid3.pack(fill='both')
leftFrame_bottom = Frame(leftFrame, relief=RAISED, borderwidth=1)
leftFrame_bottom.pack(fill='both')

# (right frame)-------------
rightFrame = Frame(window, relief=RAISED, borderwidth=1, bg='blue')
rightFrame.pack(side=RIGHT, fill='both', expand=1)

# right frame is divided into top mid2 and buttom frame
rightFrame_top = Frame(rightFrame, relief=RAISED, borderwidth=1,
bg='yellow')
rightFrame_top.pack(fill='both', expand=1)
rightFrame_Mid2 = Frame(rightFrame, relief=RAISED, borderwidth=1,
bg='yellow')
rightFrame_Mid2.pack(fill='both', expand=1)
rightFrame_bottom = Frame(rightFrame, relief=RAISED, borderwidth=1,
bg='blue')
rightFrame_bottom.pack()

# (labels and titles)-------------
Elementextractiontitle = Label(leftFrame_top, text='Element extraction',
font='none 10 bold')
Elementextractiontitle.pack(side=TOP, anchor=NW)

Setextractiontitle = Label(leftFrame_Mid, text='Extraction as required',
font='none 10 bold')
Setextractiontitle.pack(side=TOP, anchor=NW)

# main development: codes/ functionalities

Elementextractionlabel = Label(leftFrame_top,text='After importing an IFC
file, use this section to extract building elements:')
Elementextractionlabel.pack(side=TOP, anchor=NW, pady=5)


# to extract building elements: columns......
class Columns():
    def __init__(self):
        try:
            self.ifc_file = ifcopenshell.open(filename)
            self.process()
        except:
            print('Error')

    def process(self):
        Products = self.ifc_file.by_type('IfcSlab')
```

```
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcSlabType')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcWall')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcWallType')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcBeam')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcBeamType')
        for Product in Products:
            self.ifc_file.remove(Product)

        self.ifc_file.write("Columns.ifc")

if __name__ == '__main__':
    Columns()
Column = ttk.Button(leftFrame_top, text='Columns', width=8,
command=Columns).pack(side=LEFT, anchor=NE, padx=5, pady=10)


# to extract building elements: Walls......
class Walls():
    def __init__(self):
        try:
            self.ifc_file = ifcopenshell.open(filename)
            self.process()
        except:
            print('Error')

    def process(self):
        Products = self.ifc_file.by_type('IfcSlab')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcSlabType')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcColumn')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcColumnType')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcBeam')
        for Product in Products:
            self.ifc_file.remove(Product)
```

```
        Products = self.ifc_file.by_type('IfcBeamType')
        for Product in Products:
            self.ifc_file.remove(Product)

        self.ifc_file.write("Walls.ifc")

if __name__ == '__main__':
    Walls()
Wall = ttk.Button(leftFrame_top, text='Walls', width=8,
command=Walls).pack(side=LEFT, anchor=NE, padx=5, pady=10)


# to extract building elements: Slabs......
class Slabs():
    def __init__(self):
        try:
            self.ifc_file = ifcopenshell.open(filename)
            self.process()
        except:
            print('Error')

    def process(self):
        Products = self.ifc_file.by_type('IfcWall')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcWallType')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcColumn')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcColumnType')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcBeam')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcBeamType')
        for Product in Products:
            self.ifc_file.remove(Product)

        self.ifc_file.write("Slabs.ifc")

if __name__ == '__main__':
    Slabs()
Slab = ttk.Button(leftFrame_top, text='Slabs', width=8,
command=Slabs).pack(side=LEFT, anchor=NE, padx=5, pady=10)


# to extract building elements: Beams......
class Beams():
    def __init__(self):
        try:
            self.ifc_file = ifcopenshell.open(filename)
            self.process()
        except:
```

```
            print('Error')

    def process(self):
        Products = self.ifc_file.by_type('IfcSlab')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcSlabType')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcColumn')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcColumnType')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcWall')
        for Product in Products:
            self.ifc_file.remove(Product)

        Products = self.ifc_file.by_type('IfcWallType')
        for Product in Products:
            self.ifc_file.remove(Product)

        self.ifc_file.write("Beams.ifc")

if __name__ == '__main__':
    Beams()
Slab = ttk.Button(leftFrame_top, text='Beams', width=8,
command=Beams).pack(side=LEFT, anchor=NE, padx=5, pady=10)

# Partial model extraction
# label: Description/ the purpose of this section:
Setextractionlabel = Label(leftFrame_Mid, text='After importing an IFC
file, use this section to extract data sets:')
Setextractionlabel.pack(side=TOP, anchor=NW, pady=5)

class Manydataset():
    def __init__(self):
        try:
            self.ifc_file = ifcopenshell.open(filename)
            self.ifc_file2 = ifcopenshell.file()
            self.process()
        except:
            print('Error')

    def process(self):
        IfcRelAggregates = self.ifc_file.by_type('IfcRelAggregates')
        for Product in IfcRelAggregates:
            print(f'RelAggregates : {Product}')
            self.ifc_file2.add(Product)

        IfcSite = self.ifc_file.by_type('IfcSite')
        for Product in IfcSite:
            print(f'Site: {Product}')
            self.ifc_file2.add(Product)

        IfcBuilding = self.ifc_file.by_type('IfcBuilding')
```

```
        for Product in IfcBuilding:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcBuildingStorey = self.ifc_file.by_type('IfcBuildingStorey')
        for Product in IfcBuildingStorey:
            print(f'BuildingStorey: {Product}')
            self.ifc_file2.add(Product)

        IfcRelContainedInSpatialStructure =
self.ifc_file.by_type('IfcRelContainedInSpatialStructure')
        for Product in IfcRelContainedInSpatialStructure:
            print(f'RelContainedInSpatialStructure : {Product}')
            self.ifc_file2.add(Product)

        IfcActorRole = self.ifc_file.by_type('IfcActorRole')
        for Product in IfcActorRole:
            print(f'ActorRole: {Product}')
            self.ifc_file2.add(Product)

        IfcProject = self.ifc_file.by_type('IfcProject')
        for Product in IfcProject:
            print(f'Project: {Product}')
            self.ifc_file2.add(Product)

        IfcColumnStandardCase =
self.ifc_file.by_type('IfcColumnStandardCase')
        for Product in IfcColumnStandardCase:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcSlabStandardCase = self.ifc_file.by_type('IfcSlabStandardCase')
        for Product in IfcSlabStandardCase:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcWallStandardCase = self.ifc_file.by_type('IfcWallStandardCase')
        for Product in IfcWallStandardCase:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcWallStandardCase = self.ifc_file.by_type('IfcWallStandardCase')
        for Product in IfcWallStandardCase:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcBeam = self.ifc_file.by_type('IfcBeam')
        for Product in IfcBeam:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralCurveMember =
self.ifc_file.by_type('IfcStructuralCurveMember')
        for Product in IfcStructuralCurveMember:
            print(f'StructuralCurveMember: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralSurfaceMember =
self.ifc_file.by_type('IfcStructuralSurfaceMember')
        for Product in IfcStructuralSurfaceMember:
            print(f'StructuralCurveMember: {Product}')
```

```
            self.ifc_file2.add(Product)

        IfcArbitraryClosedProfileDef =
self.ifc_file.by_type('IfcArbitraryClosedProfileDef')
        for Product in IfcArbitraryClosedProfileDef:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcCircleProfileDef = self.ifc_file.by_type('IfcCircleProfileDef')
        for Product in IfcCircleProfileDef:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcExtrudedAreaSolid =
self.ifc_file.by_type('IfcExtrudedAreaSolid')
        for Product in IfcExtrudedAreaSolid:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcRectangleProfileDef =
self.ifc_file.by_type('IfcRectangleProfileDef')
        for Product in IfcRectangleProfileDef:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)


        IfcProductDefinitionShape =
self.ifc_file.by_type('IfcProductDefinitionShape')
        for Product in IfcProductDefinitionShape:
            print(f'ProductDefinitionShape: {Product}')
            self.ifc_file2.add(Product)


        IfcGeometricRepresentationContext =
self.ifc_file.by_type('IfcGeometricRepresentationContext')
        for Product in IfcGeometricRepresentationContext:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcShapeRepresentation =
self.ifc_file.by_type('IfcShapeRepresentation')
        for Product in IfcShapeRepresentation:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcApplication = self.ifc_file.by_type('IfcApplication')
        for Product in IfcApplication:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcOrganization = self.ifc_file.by_type('IfcOrganization')
        for Product in IfcOrganization:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcOwnerHistory = self.ifc_file.by_type('IfcOwnerHistory')
        for Product in IfcOwnerHistory:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPerson = self.ifc_file.by_type('IfcPerson')
```

```
        for Product in IfcPerson:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPersonAndOrganization =
self.ifc_file.by_type('IfcPersonAndOrganization')
        for Product in IfcPersonAndOrganization:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcDerivedUnit = self.ifc_file.by_type('IfcDerivedUnit')
        for Product in IfcDerivedUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcDerivedUnitElement =
self.ifc_file.by_type('IfcDerivedUnitElement')
        for Product in IfcDerivedUnitElement:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcSIUnit = self.ifc_file.by_type('IfcSIUnit')
        for Product in IfcSIUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcUnitAssignment = self.ifc_file.by_type('IfcUnitAssignment')
        for Product in IfcUnitAssignment:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)



        self.ifc_file2.write("many Data Set.ifc")


if __name__ == '__main__':
    Manydataset()

MANYTOMANY = ttk.Button(leftFrame_Mid, text='Single data Set \n(Many to
many) ', width=22, command=Manydataset)
MANYTOMANY.pack(side=LEFT, anchor=NE, padx=5, pady=10)

# to extract structual data set:
class Structuraldataset():
    def __init__(self):
        try:
            self.ifc_file = ifcopenshell.open(filename)
            self.ifc_file2 = ifcopenshell.file()
            self.process()
        except:
            print('Error')

    def process(self):
        IfcRelAggregates = self.ifc_file.by_type('IfcRelAggregates')
        for Product in IfcRelAggregates:
            print(f'RelAggregates : {Product}')
            self.ifc_file2.add(Product)

        IfcSite = self.ifc_file.by_type('IfcSite')
        for Product in IfcSite:
```

```
            print(f'Site: {Product}')
            self.ifc_file2.add(Product)

        IfcBuilding = self.ifc_file.by_type('IfcBuilding')
        for Product in IfcBuilding:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcBuildingStorey = self.ifc_file.by_type('IfcBuildingStorey')
        for Product in IfcBuildingStorey:
            print(f'BuildingStorey: {Product}')
            self.ifc_file2.add(Product)

        IfcRelContainedInSpatialStructure =
self.ifc_file.by_type('IfcRelContainedInSpatialStructure')
        for Product in IfcRelContainedInSpatialStructure:
            print(f'RelContainedInSpatialStructure : {Product}')
            self.ifc_file2.add(Product)

        IfcActorRole = self.ifc_file.by_type('IfcActorRole')
        for Product in IfcActorRole:
            print(f'ActorRole: {Product}')
            self.ifc_file2.add(Product)

        IfcProject = self.ifc_file.by_type('IfcProject')
        for Product in IfcProject:
            print(f'Project: {Product}')
            self.ifc_file2.add(Product)

        IfcColumnStandardCase =
self.ifc_file.by_type('IfcColumnStandardCase')
        for Product in IfcColumnStandardCase:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcSlabStandardCase = self.ifc_file.by_type('IfcSlabStandardCase')
        for Product in IfcSlabStandardCase:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcWallStandardCase = self.ifc_file.by_type('IfcWallStandardCase')
        for Product in IfcWallStandardCase:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcWallStandardCase = self.ifc_file.by_type('IfcWallStandardCase')
        for Product in IfcWallStandardCase:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcBeam = self.ifc_file.by_type('IfcBeam')
        for Product in IfcBeam:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralCurveMember =
self.ifc_file.by_type('IfcStructuralCurveMember')
        for Product in IfcStructuralCurveMember:
            print(f'StructuralCurveMember: {Product}')
            self.ifc_file2.add(Product)
```

```
        IfcStructuralSurfaceMember =
self.ifc_file.by_type('IfcStructuralSurfaceMember')
        for Product in IfcStructuralSurfaceMember:
            print(f'StructuralCurveMember: {Product}')
            self.ifc_file2.add(Product)

        IfcArbitraryClosedProfileDef =
self.ifc_file.by_type('IfcArbitraryClosedProfileDef')
        for Product in IfcArbitraryClosedProfileDef:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcCircleProfileDef = self.ifc_file.by_type('IfcCircleProfileDef')
        for Product in IfcCircleProfileDef:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcExtrudedAreaSolid =
self.ifc_file.by_type('IfcExtrudedAreaSolid')
        for Product in IfcExtrudedAreaSolid:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcRectangleProfileDef =
self.ifc_file.by_type('IfcRectangleProfileDef')
        for Product in IfcRectangleProfileDef:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcRelAssociatesMaterial =
self.ifc_file.by_type('IfcRelAssociatesMaterial')
        for Product in IfcRelAssociatesMaterial:
            print(f'RelAssociatesMaterial: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterial = self.ifc_file.by_type('IfcMaterial')
        for Product in IfcMaterial:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialLayer = self.ifc_file.by_type('IfcMaterialLayer')
        for Product in IfcMaterialLayer:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialLayerSet = self.ifc_file.by_type('IfcMaterialLayerSet')
        for Product in IfcMaterialLayerSet:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialLayerSetUsage =
self.ifc_file.by_type('IfcMaterialLayerSetUsage')
        for Product in IfcMaterialLayerSetUsage:
            print(f'RelAssociatesMaterial: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialProfile = self.ifc_file.by_type('IfcMaterialProfile')
        for Product in IfcMaterialProfile:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)
```

```
        IfcMaterialProfileSet =
self.ifc_file.by_type('IfcMaterialProfileSet')
        for Product in IfcMaterialProfileSet:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialProfileSetUsage =
self.ifc_file.by_type('IfcMaterialProfileSetUsage')
        for Product in IfcMaterialProfileSetUsage:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcProductDefinitionShape =
self.ifc_file.by_type('IfcProductDefinitionShape')
        for Product in IfcProductDefinitionShape:
            print(f'ProductDefinitionShape: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialProperties =
self.ifc_file.by_type('IfcMaterialProperties')
        for Product in IfcMaterialProperties:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcComplexProperty = self.ifc_file.by_type('IfcComplexProperty')
        for Product in IfcComplexProperty:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcBuildingElementProxy =
self.ifc_file.by_type('IfcBuildingElementProxy')
        for Product in IfcBuildingElementProxy:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPropertySingleValue =
self.ifc_file.by_type('IfcPropertySingleValue')
        for Product in IfcPropertySingleValue:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcGeometricRepresentationContext =
self.ifc_file.by_type('IfcGeometricRepresentationContext')
        for Product in IfcGeometricRepresentationContext:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcShapeRepresentation =
self.ifc_file.by_type('IfcShapeRepresentation')
        for Product in IfcShapeRepresentation:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcApplication = self.ifc_file.by_type('IfcApplication')
        for Product in IfcApplication:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcOrganization = self.ifc_file.by_type('IfcOrganization')
        for Product in IfcOrganization:
            print(f'Element: {Product}')
```

```
            self.ifc_file2.add(Product)

        IfcOwnerHistory = self.ifc_file.by_type('IfcOwnerHistory')
        for Product in IfcOwnerHistory:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPerson = self.ifc_file.by_type('IfcPerson')
        for Product in IfcPerson:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPersonAndOrganization =
self.ifc_file.by_type('IfcPersonAndOrganization')
        for Product in IfcPersonAndOrganization:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcDerivedUnit = self.ifc_file.by_type('IfcDerivedUnit')
        for Product in IfcDerivedUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcDerivedUnitElement =
self.ifc_file.by_type('IfcDerivedUnitElement')
        for Product in IfcDerivedUnitElement:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcSIUnit = self.ifc_file.by_type('IfcSIUnit')
        for Product in IfcSIUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcUnitAssignment = self.ifc_file.by_type('IfcUnitAssignment')
        for Product in IfcUnitAssignment:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcAxis2Placement2D = self.ifc_file.by_type('IfcAxis2Placement2D')
        for Product in IfcAxis2Placement2D:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcAxis2Placement3D = self.ifc_file.by_type('IfcAxis2Placement3D')
        for Product in IfcAxis2Placement3D:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcCartesianPoint = self.ifc_file.by_type('IfcCartesianPoint')
        for Product in IfcCartesianPoint:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcDirection = self.ifc_file.by_type('IfcDirection')
        for Product in IfcDirection:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcLocalPlacement = self.ifc_file.by_type('IfcLocalPlacement')
        for Product in IfcLocalPlacement:
```

```
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcRelAssignsToGroup =
self.ifc_file.by_type('IfcRelAssignsToGroup')
        for Product in IfcRelAssignsToGroup:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcRelAssignsToGroupByFactor =
self.ifc_file.by_type('IfcRelAssignsToGroupByFactor')
        for Product in IfcRelAssignsToGroupByFactor:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcRelAssignsToGroupByFactor =
self.ifc_file.by_type('IfcRelAssignsToGroupByFactor')
        for Product in IfcRelAssignsToGroupByFactor:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcTopologyRepresentation =
self.ifc_file.by_type('IfcTopologyRepresentation')
        for Product in IfcTopologyRepresentation:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcBoundaryNodeCondition =
self.ifc_file.by_type('IfcBoundaryNodeCondition')
        for Product in IfcBoundaryNodeCondition:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcRelConnectsStructuralActivity =
self.ifc_file.by_type('IfcRelConnectsStructuralActivity')
        for Product in IfcRelConnectsStructuralActivity:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcRelConnectsStructuralMember =
self.ifc_file.by_type('IfcRelConnectsStructuralMember')
        for Product in IfcRelConnectsStructuralMember:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcRelServicesBuildings =
self.ifc_file.by_type('IfcRelServicesBuildings')
        for Product in IfcRelServicesBuildings:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralAnalysisModel =
self.ifc_file.by_type('IfcStructuralAnalysisModel')
        for Product in IfcStructuralAnalysisModel:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralLoadCase =
self.ifc_file.by_type('IfcStructuralLoadCase')
        for Product in IfcStructuralLoadCase:
            print(f'RelAssignsToGroup: {Product}')
```

```
            self.ifc_file2.add(Product)

        IfcStructuralLoadGroup =
self.ifc_file.by_type('IfcStructuralLoadGroup')
        for Product in IfcStructuralLoadGroup:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralLoadPlanarForce =
self.ifc_file.by_type('IfcStructuralLoadPlanarForce')
        for Product in IfcStructuralLoadPlanarForce:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralPlanarAction =
self.ifc_file.by_type('IfcStructuralPlanarAction')
        for Product in IfcStructuralPlanarAction:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralPointConnection =
self.ifc_file.by_type('IfcStructuralPointConnection')
        for Product in IfcStructuralPointConnection:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcEdge = self.ifc_file.by_type('IfcEdge')
        for Product in IfcEdge:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcEdgeLoop = self.ifc_file.by_type('IfcEdgeLoop')
        for Product in IfcEdgeLoop:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcFaceBound = self.ifc_file.by_type('IfcFaceBound')
        for Product in IfcFaceBound:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcFaceSurface = self.ifc_file.by_type('IfcFaceSurface')
        for Product in IfcFaceSurface:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcOrientedEdge = self.ifc_file.by_type('IfcOrientedEdge')
        for Product in IfcOrientedEdge:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPlane = self.ifc_file.by_type('IfcPlane')
        for Product in IfcPlane:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcVertexPoint = self.ifc_file.by_type('IfcVertexPoint')
        for Product in IfcVertexPoint:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)
```

```
        IfcPolyline = self.ifc_file.by_type('IfcPolyline')
        for Product in IfcPolyline:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcRelAssignsToProduct =
self.ifc_file.by_type('IfcRelAssignsToProduct')
        for Product in IfcRelAssignsToProduct:
            print(f'RelAssignsToProduct: {Product}')
            self.ifc_file2.add(Product)

        IfcRelDefinesByProperties =
self.ifc_file.by_type('IfcRelDefinesByProperties')
        for Product in IfcRelDefinesByProperties:
            print(f'RelDefinesByProperties: {Product}')
            self.ifc_file2.add(Product)

        IfcComplexProperty = self.ifc_file.by_type('IfcComplexProperty')
        for Product in IfcComplexProperty:
            print(f'ComplexProperty: {Product}')
            self.ifc_file2.add(Product)

        IfcTypeProduct = self.ifc_file.by_type('IfcTypeProduct')
        for Product in IfcTypeProduct:
            print(f'TypeProduct: {Product}')
            self.ifc_file2.add(Product)

        self.ifc_file2.write("Structural Data Set.ifc")


if __name__ == '__main__':
    Structuraldataset()

STRUCTURALDATASET = ttk.Button(leftFrame_Mid, text='Structural Data Set \n
(one to one) ', width=22, command=Structuraldataset)
STRUCTURALDATASET.pack(side=LEFT, anchor=NE, padx=5, pady=10)


# to extract Cost data set:
class CostDataSet():
    def __init__(self):
        try:
            self.ifc_file = ifcopenshell.open(filename)
            self.ifc_file2 = ifcopenshell.file()
            self.process()
        except:
            print('Error')

    def process(self):
        IfcRelAggregates = self.ifc_file.by_type('IfcRelAggregates')
        for Product in IfcRelAggregates:
            print(f'RelAggregates : {Product}')
            self.ifc_file2.add(Product)

        IfcSite = self.ifc_file.by_type('IfcSite')
        for Product in IfcSite:
            print(f'Site: {Product}')
            self.ifc_file2.add(Product)

        IfcBuilding = self.ifc_file.by_type('IfcBuilding')
        for Product in IfcBuilding:
```

```
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcBuildingStorey = self.ifc_file.by_type('IfcBuildingStorey')
        for Product in IfcBuildingStorey:
            print(f'BuildingStorey: {Product}')
            self.ifc_file2.add(Product)

        IfcRelContainedInSpatialStructure = 
self.ifc_file.by_type('IfcRelContainedInSpatialStructure')
        for Product in IfcRelContainedInSpatialStructure:
            print(f'RelContainedInSpatialStructure : {Product}')
            self.ifc_file2.add(Product)

        IfcColumn = self.ifc_file.by_type('IfcColumn')
        for Product in IfcColumn:
            print(f'RelAssociatesMaterial: {Product}')
            self.ifc_file2.add(Product)

        IfcColumnType = self.ifc_file.by_type('IfcColumnType')
        for Product in IfcColumnType:
            print(f'ProductDefinitionShape: {Product}')
            self.ifc_file2.add(Product)

        IfcSlab = self.ifc_file.by_type('IfcSlab')
        for Product in IfcSlab:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcSlabType = self.ifc_file.by_type('IfcSlabType')
        for Product in IfcSlabType:
            print(f'TypeProduct: {Product}')
            self.ifc_file2.add(Product)

        IfcWall = self.ifc_file.by_type('IfcWall')
        for Product in IfcWall:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcWallType = self.ifc_file.by_type('IfcWallType')
        for Product in IfcWallType:
            print(f'TypeProduct: {Product}')
            self.ifc_file2.add(Product)

        IfcBeam = self.ifc_file.by_type('IfcBeam')
        for Product in IfcBeam:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcBeamType = self.ifc_file.by_type('IfcBeamType')
        for Product in IfcBeamType:
            print(f'TypeProduct: {Product}')
            self.ifc_file2.add(Product)

        IfcOpeningElement = self.ifc_file.by_type('IfcOpeningElement')
        for Product in IfcOpeningElement:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterial = self.ifc_file.by_type('IfcMaterial')
        for Product in IfcMaterial:
```

```
            print(f'ActorRole: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialList = self.ifc_file.by_type('IfcMaterialList')
        for Product in IfcMaterialList:
            print(f'RelDefinesByProperties: {Product}')
            self.ifc_file2.add(Product)

        IfcRelAssociatesMaterial =
self.ifc_file.by_type('IfcRelAssociatesMaterial')
        for Product in IfcRelAssociatesMaterial:
            print(f'ComplexProperty: {Product}')
            self.ifc_file2.add(Product)

        IfcPropertySet = self.ifc_file.by_type('IfcPropertySet')
        for Product in IfcPropertySet:
            print(f'IfcMaterialProperties: {Product}')
            self.ifc_file2.add(Product)

        IfcRelDefinesByProperties =
self.ifc_file.by_type('IfcRelDefinesByProperties')
        for Product in IfcRelDefinesByProperties:
            print(f'ComplexProperty: {Product}')
            self.ifc_file2.add(Product)

        IfcPropertySingleValue =
self.ifc_file.by_type('IfcPropertySingleValue')
        for Product in IfcPropertySingleValue:
            print(f'IfcMaterialProperties: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialProperties =
self.ifc_file.by_type('IfcMaterialProperties')
        for Product in IfcMaterialProperties:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcComplexProperty = self.ifc_file.by_type('IfcComplexProperty')
        for Product in IfcComplexProperty:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcBuildingElementProxy =
self.ifc_file.by_type('IfcBuildingElementProxy')
        for Product in IfcBuildingElementProxy:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcGeometricRepresentationContext =
self.ifc_file.by_type('IfcGeometricRepresentationContext')
        for Product in IfcGeometricRepresentationContext:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcGeometricRepresentationSubContext =
self.ifc_file.by_type('IfcGeometricRepresentationSubContext')
        for Product in IfcGeometricRepresentationSubContext:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialDefinitionRepresentation =
```

```
self.ifc_file.by_type('IfcMaterialDefinitionRepresentation')
        for Product in IfcMaterialDefinitionRepresentation:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcProductDefinitionShape =
self.ifc_file.by_type('IfcProductDefinitionShape')
        for Product in IfcProductDefinitionShape:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcShapeRepresentation =
self.ifc_file.by_type('IfcShapeRepresentation')
        for Product in IfcShapeRepresentation:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcStyledRepresentation =
self.ifc_file.by_type('IfcStyledRepresentation')
        for Product in IfcStyledRepresentation:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcApplication = self.ifc_file.by_type('IfcApplication')
        for Product in IfcApplication:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcOrganization = self.ifc_file.by_type('IfcOrganization')
        for Product in IfcOrganization:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcOwnerHistory = self.ifc_file.by_type('IfcOwnerHistory')
        for Product in IfcOwnerHistory:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPerson = self.ifc_file.by_type('IfcPerson')
        for Product in IfcPerson:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPersonAndOrganization =
self.ifc_file.by_type('IfcPersonAndOrganization')
        for Product in IfcPersonAndOrganization:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcProject = self.ifc_file.by_type('IfcProject')
        for Product in IfcProject:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcSIUnit = self.ifc_file.by_type('IfcSIUnit')
        for Product in IfcSIUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcUnitAssignment = self.ifc_file.by_type('IfcUnitAssignment')
        for Product in IfcUnitAssignment:
```

```
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcMonetaryUnit = self.ifc_file.by_type('IfcMonetaryUnit')
        for Product in IfcMonetaryUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcMeasureWithUnit = self.ifc_file.by_type('IfcMeasureWithUnit')
        for Product in IfcMeasureWithUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcDimensionalExponents =
self.ifc_file.by_type('IfcDimensionalExponents')
        for Product in IfcDimensionalExponents:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcConversionBasedUnit =
self.ifc_file.by_type('IfcConversionBasedUnit')
        for Product in IfcConversionBasedUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcRelConnectsPathElements =
self.ifc_file.by_type('IfcRelConnectsPathElements')
        for Product in IfcRelConnectsPathElements:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcRelDefinesByType = self.ifc_file.by_type('IfcRelDefinesByType')
        for Product in IfcRelDefinesByType:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcColourRgb = self.ifc_file.by_type('IfcColourRgb')
        for Product in IfcColourRgb:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPresentationLayerAssignment =
self.ifc_file.by_type('IfcPresentationLayerAssignment')
        for Product in IfcPresentationLayerAssignment:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPresentationStyleAssignment =
self.ifc_file.by_type('IfcPresentationStyleAssignment')
        for Product in IfcPresentationStyleAssignment:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcStyledItem = self.ifc_file.by_type('IfcStyledItem')
        for Product in IfcStyledItem:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcSurfaceStyle = self.ifc_file.by_type('IfcSurfaceStyle')
        for Product in IfcSurfaceStyle:
            print(f'Element: {Product}')
```

```
            self.ifc_file2.add(Product)

        IfcSurfaceStyleRendering =
self.ifc_file.by_type('IfcSurfaceStyleRendering')
        for Product in IfcSurfaceStyleRendering:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcElementQuantity = self.ifc_file.by_type('IfcElementQuantity')
        for Product in IfcElementQuantity:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcQuantityArea = self.ifc_file.by_type('IfcQuantityArea')
        for Product in IfcQuantityArea:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcQuantityCount = self.ifc_file.by_type('IfcQuantityCount')
        for Product in IfcQuantityCount:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcQuantityLength= self.ifc_file.by_type('IfcQuantityLength')
        for Product in IfcQuantityLength:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcQuantityVolume = self.ifc_file.by_type('IfcQuantityVolume')
        for Product in IfcQuantityVolume:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcAxis2Placement3D = self.ifc_file.by_type('IfcAxis2Placement3D')
        for Product in IfcAxis2Placement3D:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcCartesianPoint = self.ifc_file.by_type('IfcCartesianPoint')
        for Product in IfcCartesianPoint:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcDirection = self.ifc_file.by_type('IfcDirection')
        for Product in IfcDirection:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcLocalPlacement = self.ifc_file.by_type('IfcLocalPlacement')
        for Product in IfcLocalPlacement:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcClosedShell = self.ifc_file.by_type('IfcClosedShell')
        for Product in IfcClosedShell:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPolyline = self.ifc_file.by_type('IfcPolyline')
        for Product in IfcPolyline:
            print(f'Element: {Product}')
```

```
            self.ifc_file2.add(Product)

        IfcPolyLoop = self.ifc_file.by_type('IfcPolyLoop')
        for Product in IfcPolyLoop:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcFace = self.ifc_file.by_type('IfcFace')
        for Product in IfcFace:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcFaceBound = self.ifc_file.by_type('IfcFaceBound')
        for Product in IfcFaceBound:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcFaceOuterBound = self.ifc_file.by_type('IfcFaceOuterBound')
        for Product in IfcFaceOuterBound:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcFacetedBrep = self.ifc_file.by_type('IfcFacetedBrep')
        for Product in IfcFacetedBrep:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        self.ifc_file2.write("Cost Data Set.ifc")


if __name__ == '__main__':
    CostDataSet()

COSTDATASET = ttk.Button(leftFrame_Mid, text='Cost Data Set \n (one to one)
', width=22, command=CostDataSet)
COSTDATASET.pack(side=LEFT, anchor=NE, padx=5, pady=10)


class onetomany():
    def __init__(self):
        try:
            self.ifc_file = ifcopenshell.open(filename)
            self.ifc_file2 = ifcopenshell.file()
            self.ifc_file3 = ifcopenshell.file()
            self.process()
            self.process1()
        except:
            print('Error')

    def process(self):
        IfcRelAggregates = self.ifc_file.by_type('IfcRelAggregates')
        for Product in IfcRelAggregates:
            print(f'RelAggregates : {Product}')
            self.ifc_file2.add(Product)

        IfcSite = self.ifc_file.by_type('IfcSite')
        for Product in IfcSite:
            print(f'Site: {Product}')
            self.ifc_file2.add(Product)

        IfcBuilding = self.ifc_file.by_type('IfcBuilding')
```

```
        for Product in IfcBuilding:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcBuildingStorey = self.ifc_file.by_type('IfcBuildingStorey')
        for Product in IfcBuildingStorey:
            print(f'BuildingStorey: {Product}')
            self.ifc_file2.add(Product)

        IfcRelContainedInSpatialStructure =
self.ifc_file.by_type('IfcRelContainedInSpatialStructure')
        for Product in IfcRelContainedInSpatialStructure:
            print(f'RelContainedInSpatialStructure : {Product}')
            self.ifc_file2.add(Product)

        IfcActorRole = self.ifc_file.by_type('IfcActorRole')
        for Product in IfcActorRole:
            print(f'ActorRole: {Product}')
            self.ifc_file2.add(Product)

        IfcProject = self.ifc_file.by_type('IfcProject')
        for Product in IfcProject:
            print(f'Project: {Product}')
            self.ifc_file2.add(Product)

        IfcColumnStandardCase =
self.ifc_file.by_type('IfcColumnStandardCase')
        for Product in IfcColumnStandardCase:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcSlabStandardCase = self.ifc_file.by_type('IfcSlabStandardCase')
        for Product in IfcSlabStandardCase:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcWallStandardCase = self.ifc_file.by_type('IfcWallStandardCase')
        for Product in IfcWallStandardCase:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcWallStandardCase = self.ifc_file.by_type('IfcWallStandardCase')
        for Product in IfcWallStandardCase:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcBeam = self.ifc_file.by_type('IfcBeam')
        for Product in IfcBeam:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralCurveMember =
self.ifc_file.by_type('IfcStructuralCurveMember')
        for Product in IfcStructuralCurveMember:
            print(f'StructuralCurveMember: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralSurfaceMember =
self.ifc_file.by_type('IfcStructuralSurfaceMember')
        for Product in IfcStructuralSurfaceMember:
            print(f'StructuralCurveMember: {Product}')
```

```
            self.ifc_file2.add(Product)

        IfcArbitraryClosedProfileDef =
self.ifc_file.by_type('IfcArbitraryClosedProfileDef')
        for Product in IfcArbitraryClosedProfileDef:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcCircleProfileDef = self.ifc_file.by_type('IfcCircleProfileDef')
        for Product in IfcCircleProfileDef:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcExtrudedAreaSolid =
self.ifc_file.by_type('IfcExtrudedAreaSolid')
        for Product in IfcExtrudedAreaSolid:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcRectangleProfileDef =
self.ifc_file.by_type('IfcRectangleProfileDef')
        for Product in IfcRectangleProfileDef:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcRelAssociatesMaterial =
self.ifc_file.by_type('IfcRelAssociatesMaterial')
        for Product in IfcRelAssociatesMaterial:
            print(f'RelAssociatesMaterial: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterial = self.ifc_file.by_type('IfcMaterial')
        for Product in IfcMaterial:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialLayer = self.ifc_file.by_type('IfcMaterialLayer')
        for Product in IfcMaterialLayer:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialLayerSet = self.ifc_file.by_type('IfcMaterialLayerSet')
        for Product in IfcMaterialLayerSet:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialLayerSetUsage =
self.ifc_file.by_type('IfcMaterialLayerSetUsage')
        for Product in IfcMaterialLayerSetUsage:
            print(f'RelAssociatesMaterial: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialProfile = self.ifc_file.by_type('IfcMaterialProfile')
        for Product in IfcMaterialProfile:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialProfileSet =
self.ifc_file.by_type('IfcMaterialProfileSet')
        for Product in IfcMaterialProfileSet:
            print(f'Building: {Product}')
```

```
            self.ifc_file2.add(Product)

        IfcMaterialProfileSetUsage =
self.ifc_file.by_type('IfcMaterialProfileSetUsage')
        for Product in IfcMaterialProfileSetUsage:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcProductDefinitionShape =
self.ifc_file.by_type('IfcProductDefinitionShape')
        for Product in IfcProductDefinitionShape:
            print(f'ProductDefinitionShape: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialProperties =
self.ifc_file.by_type('IfcMaterialProperties')
        for Product in IfcMaterialProperties:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcComplexProperty = self.ifc_file.by_type('IfcComplexProperty')
        for Product in IfcComplexProperty:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcBuildingElementProxy =
self.ifc_file.by_type('IfcBuildingElementProxy')
        for Product in IfcBuildingElementProxy:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPropertySingleValue =
self.ifc_file.by_type('IfcPropertySingleValue')
        for Product in IfcPropertySingleValue:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcGeometricRepresentationContext =
self.ifc_file.by_type('IfcGeometricRepresentationContext')
        for Product in IfcGeometricRepresentationContext:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcShapeRepresentation =
self.ifc_file.by_type('IfcShapeRepresentation')
        for Product in IfcShapeRepresentation:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcApplication = self.ifc_file.by_type('IfcApplication')
        for Product in IfcApplication:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcOrganization = self.ifc_file.by_type('IfcOrganization')
        for Product in IfcOrganization:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcOwnerHistory = self.ifc_file.by_type('IfcOwnerHistory')
        for Product in IfcOwnerHistory:
```

```
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPerson = self.ifc_file.by_type('IfcPerson')
        for Product in IfcPerson:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPersonAndOrganization =
self.ifc_file.by_type('IfcPersonAndOrganization')
        for Product in IfcPersonAndOrganization:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcDerivedUnit = self.ifc_file.by_type('IfcDerivedUnit')
        for Product in IfcDerivedUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcDerivedUnitElement =
self.ifc_file.by_type('IfcDerivedUnitElement')
        for Product in IfcDerivedUnitElement:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcSIUnit = self.ifc_file.by_type('IfcSIUnit')
        for Product in IfcSIUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcUnitAssignment = self.ifc_file.by_type('IfcUnitAssignment')
        for Product in IfcUnitAssignment:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcAxis2Placement2D = self.ifc_file.by_type('IfcAxis2Placement2D')
        for Product in IfcAxis2Placement2D:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcAxis2Placement3D = self.ifc_file.by_type('IfcAxis2Placement3D')
        for Product in IfcAxis2Placement3D:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcCartesianPoint = self.ifc_file.by_type('IfcCartesianPoint')
        for Product in IfcCartesianPoint:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcDirection = self.ifc_file.by_type('IfcDirection')
        for Product in IfcDirection:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcLocalPlacement = self.ifc_file.by_type('IfcLocalPlacement')
        for Product in IfcLocalPlacement:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcRelAssignsToGroup =
```

```
self.ifc_file.by_type('IfcRelAssignsToGroup')
        for Product in IfcRelAssignsToGroup:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcRelAssignsToGroupByFactor =
self.ifc_file.by_type('IfcRelAssignsToGroupByFactor')
        for Product in IfcRelAssignsToGroupByFactor:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcRelAssignsToGroupByFactor =
self.ifc_file.by_type('IfcRelAssignsToGroupByFactor')
        for Product in IfcRelAssignsToGroupByFactor:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcTopologyRepresentation =
self.ifc_file.by_type('IfcTopologyRepresentation')
        for Product in IfcTopologyRepresentation:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcBoundaryNodeCondition =
self.ifc_file.by_type('IfcBoundaryNodeCondition')
        for Product in IfcBoundaryNodeCondition:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcRelConnectsStructuralActivity =
self.ifc_file.by_type('IfcRelConnectsStructuralActivity')
        for Product in IfcRelConnectsStructuralActivity:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcRelConnectsStructuralMember =
self.ifc_file.by_type('IfcRelConnectsStructuralMember')
        for Product in IfcRelConnectsStructuralMember:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcRelServicesBuildings =
self.ifc_file.by_type('IfcRelServicesBuildings')
        for Product in IfcRelServicesBuildings:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralAnalysisModel =
self.ifc_file.by_type('IfcStructuralAnalysisModel')
        for Product in IfcStructuralAnalysisModel:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralLoadCase =
self.ifc_file.by_type('IfcStructuralLoadCase')
        for Product in IfcStructuralLoadCase:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralLoadGroup =
self.ifc_file.by_type('IfcStructuralLoadGroup')
```

```
        for Product in IfcStructuralLoadGroup:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralLoadPlanarForce =
self.ifc_file.by_type('IfcStructuralLoadPlanarForce')
        for Product in IfcStructuralLoadPlanarForce:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralPlanarAction =
self.ifc_file.by_type('IfcStructuralPlanarAction')
        for Product in IfcStructuralPlanarAction:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcStructuralPointConnection =
self.ifc_file.by_type('IfcStructuralPointConnection')
        for Product in IfcStructuralPointConnection:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcEdge = self.ifc_file.by_type('IfcEdge')
        for Product in IfcEdge:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcEdgeLoop = self.ifc_file.by_type('IfcEdgeLoop')
        for Product in IfcEdgeLoop:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcFaceBound = self.ifc_file.by_type('IfcFaceBound')
        for Product in IfcFaceBound:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcFaceSurface = self.ifc_file.by_type('IfcFaceSurface')
        for Product in IfcFaceSurface:
            print(f'RelAssignsToGroup: {Product}')
            self.ifc_file2.add(Product)

        IfcOrientedEdge = self.ifc_file.by_type('IfcOrientedEdge')
        for Product in IfcOrientedEdge:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPlane = self.ifc_file.by_type('IfcPlane')
        for Product in IfcPlane:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcVertexPoint = self.ifc_file.by_type('IfcVertexPoint')
        for Product in IfcVertexPoint:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPolyline = self.ifc_file.by_type('IfcPolyline')
        for Product in IfcPolyline:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)
```

```
        IfcRelAssignsToProduct =
self.ifc_file.by_type('IfcRelAssignsToProduct')
        for Product in IfcRelAssignsToProduct:
            print(f'RelAssignsToProduct: {Product}')
            self.ifc_file2.add(Product)

        IfcRelDefinesByProperties =
self.ifc_file.by_type('IfcRelDefinesByProperties')
        for Product in IfcRelDefinesByProperties:
            print(f'RelDefinesByProperties: {Product}')
            self.ifc_file2.add(Product)

        IfcComplexProperty = self.ifc_file.by_type('IfcComplexProperty')
        for Product in IfcComplexProperty:
            print(f'ComplexProperty: {Product}')
            self.ifc_file2.add(Product)

        IfcTypeProduct = self.ifc_file.by_type('IfcTypeProduct')
        for Product in IfcTypeProduct:
            print(f'TypeProduct: {Product}')
            self.ifc_file2.add(Product)

        self.ifc_file2.write("Structural Data Set- one to many.ifc")

        # to extract Cost data set:

    def process1(self):
        IfcRelAggregates = self.ifc_file.by_type('IfcRelAggregates')
        for Product in IfcRelAggregates:
            print(f'RelAggregates : {Product}')
            self.ifc_file2.add(Product)

        IfcSite = self.ifc_file.by_type('IfcSite')
        for Product in IfcSite:
            print(f'Site: {Product}')
            self.ifc_file2.add(Product)

        IfcBuilding = self.ifc_file.by_type('IfcBuilding')
        for Product in IfcBuilding:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcBuildingStorey = self.ifc_file.by_type('IfcBuildingStorey')
        for Product in IfcBuildingStorey:
            print(f'BuildingStorey: {Product}')
            self.ifc_file2.add(Product)

        IfcRelContainedInSpatialStructure =
self.ifc_file.by_type('IfcRelContainedInSpatialStructure')
        for Product in IfcRelContainedInSpatialStructure:
            print(f'RelContainedInSpatialStructure : {Product}')
            self.ifc_file2.add(Product)

        IfcColumn = self.ifc_file.by_type('IfcColumn')
        for Product in IfcColumn:
            print(f'RelAssociatesMaterial: {Product}')
            self.ifc_file2.add(Product)

        IfcColumnType = self.ifc_file.by_type('IfcColumnType')
        for Product in IfcColumnType:
```

```
            print(f'ProductDefinitionShape: {Product}')
            self.ifc_file2.add(Product)

        IfcSlab = self.ifc_file.by_type('IfcSlab')
        for Product in IfcSlab:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcSlabType = self.ifc_file.by_type('IfcSlabType')
        for Product in IfcSlabType:
            print(f'TypeProduct: {Product}')
            self.ifc_file2.add(Product)

        IfcWall = self.ifc_file.by_type('IfcWall')
        for Product in IfcWall:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcWallType = self.ifc_file.by_type('IfcWallType')
        for Product in IfcWallType:
            print(f'TypeProduct: {Product}')
            self.ifc_file2.add(Product)

        IfcBeam = self.ifc_file.by_type('IfcBeam')
        for Product in IfcBeam:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcBeamType = self.ifc_file.by_type('IfcBeamType')
        for Product in IfcBeamType:
            print(f'TypeProduct: {Product}')
            self.ifc_file2.add(Product)

        IfcOpeningElement = self.ifc_file.by_type('IfcOpeningElement')
        for Product in IfcOpeningElement:
            print(f'Building: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterial = self.ifc_file.by_type('IfcMaterial')
        for Product in IfcMaterial:
            print(f'ActorRole: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialList = self.ifc_file.by_type('IfcMaterialList')
        for Product in IfcMaterialList:
            print(f'RelDefinesByProperties: {Product}')
            self.ifc_file2.add(Product)

        IfcRelAssociatesMaterial =
self.ifc_file.by_type('IfcRelAssociatesMaterial')
        for Product in IfcRelAssociatesMaterial:
            print(f'ComplexProperty: {Product}')
            self.ifc_file2.add(Product)

        IfcPropertySet = self.ifc_file.by_type('IfcPropertySet')
        for Product in IfcPropertySet:
            print(f'IfcMaterialProperties: {Product}')
            self.ifc_file2.add(Product)

        IfcRelDefinesByProperties =
self.ifc_file.by_type('IfcRelDefinesByProperties')
```

```
        for Product in IfcRelDefinesByProperties:
            print(f'ComplexProperty: {Product}')
            self.ifc_file2.add(Product)

        IfcPropertySingleValue =
self.ifc_file.by_type('IfcPropertySingleValue')
        for Product in IfcPropertySingleValue:
            print(f'IfcMaterialProperties: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialProperties =
self.ifc_file.by_type('IfcMaterialProperties')
        for Product in IfcMaterialProperties:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcComplexProperty = self.ifc_file.by_type('IfcComplexProperty')
        for Product in IfcComplexProperty:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcBuildingElementProxy =
self.ifc_file.by_type('IfcBuildingElementProxy')
        for Product in IfcBuildingElementProxy:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcGeometricRepresentationContext =
self.ifc_file.by_type('IfcGeometricRepresentationContext')
        for Product in IfcGeometricRepresentationContext:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcGeometricRepresentationSubContext =
self.ifc_file.by_type('IfcGeometricRepresentationSubContext')
        for Product in IfcGeometricRepresentationSubContext:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcMaterialDefinitionRepresentation =
self.ifc_file.by_type('IfcMaterialDefinitionRepresentation')
        for Product in IfcMaterialDefinitionRepresentation:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcProductDefinitionShape =
self.ifc_file.by_type('IfcProductDefinitionShape')
        for Product in IfcProductDefinitionShape:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcShapeRepresentation =
self.ifc_file.by_type('IfcShapeRepresentation')
        for Product in IfcShapeRepresentation:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcStyledRepresentation =
self.ifc_file.by_type('IfcStyledRepresentation')
        for Product in IfcStyledRepresentation:
            print(f'Element: {Product}')
```

```
            self.ifc_file2.add(Product)

        IfcApplication = self.ifc_file.by_type('IfcApplication')
        for Product in IfcApplication:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcOrganization = self.ifc_file.by_type('IfcOrganization')
        for Product in IfcOrganization:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcOwnerHistory = self.ifc_file.by_type('IfcOwnerHistory')
        for Product in IfcOwnerHistory:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPerson = self.ifc_file.by_type('IfcPerson')
        for Product in IfcPerson:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPersonAndOrganization =
self.ifc_file.by_type('IfcPersonAndOrganization')
        for Product in IfcPersonAndOrganization:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcProject = self.ifc_file.by_type('IfcProject')
        for Product in IfcProject:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcSIUnit = self.ifc_file.by_type('IfcSIUnit')
        for Product in IfcSIUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcUnitAssignment = self.ifc_file.by_type('IfcUnitAssignment')
        for Product in IfcUnitAssignment:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcMonetaryUnit = self.ifc_file.by_type('IfcMonetaryUnit')
        for Product in IfcMonetaryUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcMeasureWithUnit = self.ifc_file.by_type('IfcMeasureWithUnit')
        for Product in IfcMeasureWithUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcDimensionalExponents =
self.ifc_file.by_type('IfcDimensionalExponents')
        for Product in IfcDimensionalExponents:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcConversionBasedUnit =
self.ifc_file.by_type('IfcConversionBasedUnit')
```

```
        for Product in IfcConversionBasedUnit:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcRelConnectsPathElements = 
self.ifc_file.by_type('IfcRelConnectsPathElements')
        for Product in IfcRelConnectsPathElements:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcRelDefinesByType = self.ifc_file.by_type('IfcRelDefinesByType')
        for Product in IfcRelDefinesByType:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcColourRgb = self.ifc_file.by_type('IfcColourRgb')
        for Product in IfcColourRgb:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPresentationLayerAssignment = 
self.ifc_file.by_type('IfcPresentationLayerAssignment')
        for Product in IfcPresentationLayerAssignment:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcPresentationStyleAssignment = 
self.ifc_file.by_type('IfcPresentationStyleAssignment')
        for Product in IfcPresentationStyleAssignment:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcStyledItem = self.ifc_file.by_type('IfcStyledItem')
        for Product in IfcStyledItem:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcSurfaceStyle = self.ifc_file.by_type('IfcSurfaceStyle')
        for Product in IfcSurfaceStyle:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcSurfaceStyleRendering = 
self.ifc_file.by_type('IfcSurfaceStyleRendering')
        for Product in IfcSurfaceStyleRendering:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcElementQuantity = self.ifc_file.by_type('IfcElementQuantity')
        for Product in IfcElementQuantity:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcQuantityArea = self.ifc_file.by_type('IfcQuantityArea')
        for Product in IfcQuantityArea:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        IfcQuantityCount = self.ifc_file.by_type('IfcQuantityCount')
        for Product in IfcQuantityCount:
            print(f'Element: {Product}')
```

```
    self.ifc_file2.add(Product)

IfcQuantityLength = self.ifc_file.by_type('IfcQuantityLength')
for Product in IfcQuantityLength:
    print(f'Element: {Product}')
    self.ifc_file2.add(Product)

IfcQuantityVolume = self.ifc_file.by_type('IfcQuantityVolume')
for Product in IfcQuantityVolume:
    print(f'Element: {Product}')
    self.ifc_file2.add(Product)

IfcAxis2Placement3D = self.ifc_file.by_type('IfcAxis2Placement3D')
for Product in IfcAxis2Placement3D:
    print(f'Element: {Product}')
    self.ifc_file2.add(Product)

IfcCartesianPoint = self.ifc_file.by_type('IfcCartesianPoint')
for Product in IfcCartesianPoint:
    print(f'Element: {Product}')
    self.ifc_file2.add(Product)

IfcDirection = self.ifc_file.by_type('IfcDirection')
for Product in IfcDirection:
    print(f'Element: {Product}')
    self.ifc_file2.add(Product)

IfcLocalPlacement = self.ifc_file.by_type('IfcLocalPlacement')
for Product in IfcLocalPlacement:
    print(f'Element: {Product}')
    self.ifc_file2.add(Product)

IfcClosedShell = self.ifc_file.by_type('IfcClosedShell')
for Product in IfcClosedShell:
    print(f'Element: {Product}')
    self.ifc_file2.add(Product)

IfcPolyline = self.ifc_file.by_type('IfcPolyline')
for Product in IfcPolyline:
    print(f'Element: {Product}')
    self.ifc_file2.add(Product)

IfcPolyLoop = self.ifc_file.by_type('IfcPolyLoop')
for Product in IfcPolyLoop:
    print(f'Element: {Product}')
    self.ifc_file2.add(Product)

IfcFace = self.ifc_file.by_type('IfcFace')
for Product in IfcFace:
    print(f'Element: {Product}')
    self.ifc_file2.add(Product)

IfcFaceBound = self.ifc_file.by_type('IfcFaceBound')
for Product in IfcFaceBound:
    print(f'Element: {Product}')
    self.ifc_file2.add(Product)

IfcFaceOuterBound = self.ifc_file.by_type('IfcFaceOuterBound')
for Product in IfcFaceOuterBound:
    print(f'Element: {Product}')
    self.ifc_file2.add(Product)
```

```
        IfcFacetedBrep = self.ifc_file.by_type('IfcFacetedBrep')
        for Product in IfcFacetedBrep:
            print(f'Element: {Product}')
            self.ifc_file2.add(Product)

        self.ifc_file3.write("Cost Data Set- one to many.ifc")


if __name__ == '__main__':
    onetomany()

onetomany = ttk.Button(leftFrame_Mid, text='Multiple data sets \n   (one to 
many)', width=22, command=onetomany)
onetomany.pack(side=LEFT, anchor=NE, padx=5, pady=10)


window.mainloop()
```

Appendix D: Automatic data acquisition method to align the proposed ontology with the IFC file using the Rdflib and IfcOpenShell libraries

```
import rdflib
from rdflib import Literal, URIRef, Namespace, XSD, RDF
import ifcopenshell

# namespace of the ontology
Ontology = Namespace("https://www.semanticweb.org/Ontology#")

# list with asked ifcelements
BuildingElementlist = []


# Open the IFC file using IfcOpenShell
ifc_file = ifcopenshell.open(r"C:\Users\aliya\Desktop\manytomany.ifc")


for Project in ifc_file.by_type("IfcProject"):
    BuildingElementlist.append(Project)

for Site in ifc_file.by_type("IfcSite"):
    BuildingElementlist.append(Site)

for Building in ifc_file.by_type("IfcBuilding"):
    BuildingElementlist.append(Building)

for BuildingStorey in ifc_file.by_type("IfcBuildingStorey"):
    BuildingElementlist.append(BuildingStorey)

IfcColumn = ifc_file.by_type('IfcColumn')
for IfcProductDefinitionShape in IfcColumn:
    if IfcProductDefinitionShape.Representation:
        IfcProductDefinitionShape =
IfcProductDefinitionShape.Representation
        if IfcProductDefinitionShape.Representations:
            IfcShapeRepresentation =
IfcProductDefinitionShape.Representations[0]
            if IfcShapeRepresentation.Items:
                IfcMappedItem = IfcShapeRepresentation.Items[0]
                if IfcMappedItem.MappingSource:
                    IfcRepresentationMap = IfcMappedItem.MappingSource
                    if IfcRepresentationMap.MappedRepresentation:
                        IfcShapeRepresentation =
IfcRepresentationMap.MappedRepresentation
                        if IfcShapeRepresentation.Items:
                            IfcExtrudedAreaSolid =
IfcShapeRepresentation.Items[0]

BuildingElementlist.append(IfcExtrudedAreaSolid)
                            if IfcExtrudedAreaSolid.SweptArea:
                                SweptArea = IfcExtrudedAreaSolid.SweptArea
                                BuildingElementlist.append(SweptArea)


# Initiate RDF file to save IFC instances in RDF
```

```
g = rdflib.Graph()
format_ =
rdflib.util.guess_format(r'C:\Users\aliya\Desktop\NewOntology.ttl')
g.parse(r"C:\Users\aliya\Desktop\NewOntology.ttl", format=format_)
g.namespace_manager.bind('Ontology', Ontology)


# Identify URIRefs

Project =
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#Project")
Site = URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#Siteinfo")
Building =
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#Building")
BuildingStorey =
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#BuildingStorey")

RectangleColumn =
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#RectangleColumn")
CircularColumn =
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#CircularColumn")
Slab = URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#Slab")


Concrete=URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitle
d-ontology-48#Concrete")
hasConcrete=URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/unti
tled-ontology-48#hasConcrete")

ReinforcingBar=URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/u
ntitled-ontology-48#ReinforcingBar")
hasReinforcement=URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7
/untitled-ontology-48#hasReinforcement")



IfcExtrudedAreaSolid =
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#IfcExtrudedAreaSolid")


number = 1

for list in BuildingElementlist:
    list = list.get_info()
    print(list)

    if number < 10:
        bEl = URIRef(Ontology + (list['type']) + "00" + str(number))

    elif number < 100:
        bEl = URIRef(Ontology + (list['type']) + "0" + str(number))

    else:
```

```
        bEl = URIRef(Ontology + (list['type']) + str(number))


    if list['type'] == 'IfcProject':
        g.add((bEl, RDF.type, Project))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasGlobalId"), Literal(list['GlobalId'], datatype=XSD.string)))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasName"), Literal(list['Name'], datatype=XSD.string)))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasPhase"), Literal(list['Phase'], datatype=XSD.string)))
        g.add([bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasDescription"), Literal(list['Description'],
datatype=XSD.string)])
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasLongName"), Literal(list['LongName'], datatype=XSD.string)))

        number += 1

    if list['type'] == 'IfcSite':
        g.add((bEl, RDF.type, Site))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasGlobalId"), Literal(list['GlobalId'], datatype=XSD.string)))
        g.add([bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasName"), Literal(list['Name'], datatype=XSD.string)])
        g.add([bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasDescription"), Literal(list['Description'],
datatype=XSD.string)])
        g.add([bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasSiteAddress"), Literal(list['SiteAddress'],
datatype=XSD.string)])

        number += 1

    if list['type'] == 'IfcBuilding':
        g.add((bEl, RDF.type, Building))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasGlobalId"), Literal(list['GlobalId'], datatype=XSD.string)))
        g.add([bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasName"), Literal(list['Name'], datatype=XSD.string)])
        g.add([bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasDescription"), Literal(list['Description'],
datatype=XSD.string)])
        g.add([bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasElevationOfTerrain"), Literal(list['ElevationOfTerrain'],
datatype=XSD.string)])

        number += 1
```

```
    if list['type'] == 'IfcBuildingStorey':
        g.add((bEl, RDF.type, BuildingStorey))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasGlobalId"),Literal(list['GlobalId'], datatype=XSD.string)))
        g.add([bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasName"), Literal(list['Name'], datatype=XSD.string)])
        g.add([bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasDescription"),Literal(list['Description'],
datatype=XSD.string)])
        g.add([bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasElevation"),Literal(list['Elevation'], datatype=XSD.float)])

        number += 1

    if list['type'] == 'IfcRectangleProfileDef':
        g.add((bEl, RDF.type, RectangleColumn))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasName"), Literal(list['ProfileName'], datatype=XSD.string)))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasWidth"), Literal(list['YDim'], datatype=XSD.float)))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasLength"), Literal(list['XDim'], datatype=XSD.float)))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasDepth"),Literal(4000, datatype=XSD.float)))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasNbar"),Literal(6, datatype=XSD.float)))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasCover"),Literal(25, datatype=XSD.float)))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasConcreteC25"),
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#C25")))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-
48#hasConcreteC35"),URIRef("http://www.semanticweb.org/aliya/ontologies/202
1/7/untitled-ontology-48#C35")))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-
48#hasConcreteC80"),URIRef("http://www.semanticweb.org/aliya/ontologies/202
1/7/untitled-ontology-48#C80")))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-
48#hasConcreteC90"),URIRef("http://www.semanticweb.org/aliya/ontologies/202
1/7/untitled-ontology-48#C90")))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
```

```
ontology-
48#hasReinforcement"),URIRef("http://www.semanticweb.org/aliya/ontologies/2
021/7/untitled-ontology-48#ReinforcingBar")))
        # g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-
48#hasConstructionWorker"),URIRef("http://www.semanticweb.org/aliya/ontolog
ies/2021/7/untitled-ontology-48#ConstructionWorker")))

        number += 1



    if list['type'] == 'IfcCircleProfileDef':
        g.add((bEl, RDF.type, CircularColumn))
        g.add((bEl,
URIRef("http://www.semanticweb.org/aliya/ontologies/2021/7/untitled-
ontology-48#hasName"), Literal(list['ProfileName'], datatype=XSD.string)))


        number += 1

g.serialize(destination=r'C:\Users\aliya\Desktop\OntologyMixed.ttl',
format='turtle')
```

Appendix E: SWRL rules and SQWRL queries

SWRL rules
Rule1-1: Ag of rectColumn
RectangleColumn(?Column) ^ hasWidth(?Column, ?Cb) ^ hasLength(?Column, ?Ch) ^ swrlb:multiply(?ColAg, ?Cb, ?Ch) -> hasAg(?Column, ?ColAg)
Rule1-2: As of rectColumn longitudinal reinforcement
RectangleColumn(?Column) ^ hasNbar(?Column, ?CNbar) ^ ReinforcingBar(?RB) ^ hasDiameter(?RB, ?Diameter) ^ swrlb:multiply(?CAs, ?CNbar, ?Diameter, ?Diameter, 3.14, 0.25) -> hasAs(?Column, ?CAs)
Rule1-3: Ac of recColumn
RectangleColumn(?Column) ^ hasAg(?Column, ?ColAg) ^ hasAs(?Column, ?CAs) ^ swrlb:subtract(?ColAc, ?ColAg, ?CAs) -> hasAc(?Column, ?ColAc)
Rule11-1: cost of rectcolumn Concreting
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ LaborConcretingCost(?LaborConCost) ^ hasLaborConcretingCost(?LaborConCost, ?hasLaborConCost) ^ swrlb:multiply(?LCC, ?CV, ?hasLaborConCost) -> hasLaborCostConcreting(?Column, ?LCC)
Rule11-2: Labor Cost rectcolumn Reinforcement
RectangleColumn(?Column) ^ hasWeightSteel(?Column, ?WS) ^ LaborReinforcementCost(?LaborReinfCost) ^ hasLaborReinforcementCost(?LaborReinfCost, ?hasLaborReinfCost) ^ swrlb:multiply(?LCR, ?WS, ?hasLaborReinfCost) -> hasLaborCostReincorcement(?Column, ?LCR)
Rule11-3: Labor Cost rectcolumn formwork
RectangleColumn(?Column) ^ hasTotalAreaShulteringWork(?Column, ?TASHW) ^ LaborShulteringCost(?LaborShultCost) ^ hasLaborShulteringCost(?LaborShultCost, ?hasLaborShultCost) ^ swrlb:multiply(?LCF, ?TASHW, ?hasLaborShultCost) -> hasLaborCostFormwork(?Column, ?LCF)
Rule11-4: Total Labor Cost rectcolumn
RectangleColumn(?Column) ^ hasLaborCostConcreting(?Column, ?LCC) ^ hasLaborCostReincorcement(?Column, ?LCR) ^ hasLaborCostFormwork(?Column, ?LCF) ^ swrlb:add(?TotalLCC, ?LCC, ?LCR, ?LCF) -> hasTotalLaborCostColumn(?Column, ?TotalLCC)
Rule12-1: Total area of shuttering
RectangleColumn(?Column) ^ swrlb:multiply(?y, 2, ?Depth, ?Length, 0.001, 0.001) ^ swrlb:multiply(?x, 2, ?Depth, ?Width, 0.001, 0.001) ^ hasWidth(?Column, ?Width) ^ hasLength(?Column, ?Length) ^ hasDepth(?Column, ?Depth) ^ swrlb:add(?TASHW, ?x, ?y) -> hasTotalAreaShulteringWork(?Column, ?TASHW)
Rule2-1: the ultimate axial load of a rectcolumn
RectangleColumn(?Column) ^ hasAs(?Column, ?CAs) ^ hasAc(?Column, ?ColAc) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasfckC25(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) -> hasNedC25(?Column, ?CNed)
Rule2-2: the ultimate axial load of a rectcolumn
RectangleColumn(?Column) ^ hasAs(?Column, ?CAs) ^ hasAc(?Column, ?ColAc) ^ hasConcreteC35(?Column, ?Con) ^ C35(?Con) ^ hasfckC35(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) -> hasNedC35(?Column, ?CNed)
Rule2-3: the ultimate axial load of a rectcolumn
RectangleColumn(?Column) ^ hasAs(?Column, ?CAs) ^ hasAc(?Column, ?ColAc) ^ hasConcreteC80(?Column, ?Con) ^ C80(?Con) ^ hasfckC80(?Con, ?Confck) ^ ReinforcingBar(?SB) ^

hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) -> hasNedC80(?Column, ?CNed)
Rule2-4: the ultimate axial load of a rectcolumn
RectangleColumn(?Column) ^ hasAs(?Column, ?CAs) ^ hasAc(?Column, ?ColAc) ^ hasConcreteC90(?Column, ?Con) ^ C90(?Con) ^ hasfckC90(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) -> hasNedC90(?Column, ?CNed)
Rule3-1: ultimate axial load meet design condition
RectangleColumn(?Column) ^ hasAc(?Column, ?ColAc) ^ hasAs(?Column, ?CAs) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasfckC25(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) ^ swrlb:greaterThanOrEqual(?CNed, 12000000) -> meetDesignConditionC25(?Column, "Yes")
Rule3-2: ultimate axial load meet design condition
RectangleColumn(?Column) ^ hasAc(?Column, ?ColAc) ^ hasAs(?Column, ?CAs) ^ hasConcreteC35(?Column, ?Con) ^ C35(?Con) ^ hasfckC35(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) ^ swrlb:greaterThanOrEqual(?CNed, 12000000) -> meetDesignConditionC35(?Column, "Yes")
Rule3-3: ultimate axial load meet design condition
RectangleColumn(?Column) ^ hasAc(?Column, ?ColAc) ^ hasAs(?Column, ?CAs) ^ hasConcreteC80(?Column, ?Con) ^ C80(?Con) ^ hasfckC80(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) ^ swrlb:greaterThanOrEqual(?CNed, 12000000) -> meetDesignConditionC80(?Column, "Yes")
Rule3-4: ultimate axial load meet design condition
RectangleColumn(?Column) ^ hasAc(?Column, ?ColAc) ^ hasAs(?Column, ?CAs) ^ hasConcreteC90(?Column, ?Con) ^ C90(?Con) ^ hasfckC90(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) ^ swrlb:greaterThanOrEqual(?CNed, 12000000) -> meetDesignConditionC90(?Column, "Yes")
Rule4-1: Fire resistance time 60
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasWidth(?Column, ?Width) ^ hasCover(?Column, ?Cover) ^ swrlb:greaterThanOrEqual(?Width, 200) ^ swrlb:lessThan(?Width, 300) ^ swrlb:greaterThanOrEqual(?Cover, 25) -> hasFireResistanceTime60(?Column, "R60")
Rule4-2: Fire resistance time 90
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasWidth(?Column, ?Width) ^ hasCover(?Column, ?Cover) ^ swrlb:greaterThanOrEqual(?Width, 300) ^ swrlb:lessThan(?Width, 350) ^ swrlb:greaterThanOrEqual(?Cover, 25) -> hasFireResistanceTime90(?Column, "R90")
Rule4-3: Fire resistance time 120
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasWidth(?Column, ?Width) ^ hasCover(?Column, ?Cover) ^ swrlb:greaterThanOrEqual(?Width, 350) ^ swrlb:greaterThanOrEqual(?Cover, 25) -> hasFireResistanceTime120(?Column, "R120")
Rule5-1: Vol of column
RectangleColumn(?Column) ^ swrlb:multiply(?CV, ?ColAc, ?Depth, 0.0010, 0.0010, 0.0010) ^ hasAc(?Column, ?ColAc) ^ hasDepth(?Column, ?Depth) -> hasVolume(?Column, ?CV)
Rule5-2: Vol As of column
swrlb:multiply(?ColVolAs, ?CAs, ?Depth, 0.0010, 0.0010, 0.0010) ^ RectangleColumn(?Column) ^ hasAs(?Column, ?CAs) ^ hasDepth(?Column, ?Depth) -> hasVolAs(?Column, ?ColVolAs)
Rule6-1: Weight of Concrete Column
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ Concrete(?Con) ^ hasDensity(?Con, ?CD) ^ swrlb:multiply(?CW, ?CV, ?CD) -> hasWeight(?Column, ?CW)
Rule6-2: Weight of steel bars

RectangleColumn(?Column) ^ hasVolAs(?Column, ?ColVolAs) ^ Steel(?St) ^ hasDensitySteel(?st, ?DS) ^ swrlb:multiply(?WS, ?ColVolAs, ?DS) -> hasWeightSteel(?Column, ?WS)
Rule7-1: total embodied CO2eC25
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasEmbodiedCO2eC25(?Con, ?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) -> hasTotalEmbodiedCO2eC25(?Column, ?TECO2)
Rule7-2: total embodied CO2eC35
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ hasConcreteC35(?Column, ?Con) ^ C35(?Con) ^ hasEmbodiedCO2eC35(?Con, ?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) -> hasTotalEmbodiedCO2eC35(?Column, ?TECO2)
Rule7-3: total embodied CO2eC80
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ hasConcreteC80(?Column, ?Con) ^ C80(?Con) ^ hasEmbodiedCO2eC80(?Con, ?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) -> hasTotalEmbodiedCO2eC80(?Column, ?TECO2)
Rule7-4 - total embodied CO2eC90
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ hasConcreteC90(?Column, ?Con) ^ C90(?Con) ^ hasEmbodiedCO2eC90(?Con, ?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) -> hasTotalEmbodiedCO2eC90(?Column, ?TECO2)
Rule9-1: total cost of square columnC25
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasCostC25(?Con, ?Cost) ^ swrlb:multiply(?TCost, ?CV, ?Cost) -> hasTotalCostC25(?Column, ?TCost)
Rule9-2: total cost of square columnC35
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ hasConcreteC35(?Column, ?Con) ^ C35(?Con) ^ hasCostC35(?Con, ?Cost) ^ swrlb:multiply(?TCost, ?CV, ?Cost) -> hasTotalCostC35(?Column, ?TCost)
Rule9-3: total cost of square columnC80
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ hasConcreteC80(?Column, ?Con) ^ C80(?Con) ^ hasCostC80(?Con, ?Cost) ^ swrlb:multiply(?TCost, ?CV, ?Cost) -> hasTotalCostC80(?Column, ?TCost)
Rule9-4: total cost of square columnC90
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ hasConcreteC90(?Column, ?Con) ^ C90(?Con) ^ hasCostC90(?Con, ?Cost) ^ swrlb:multiply(?TCost, ?CV, ?Cost) -> hasTotalCostC90(?Column, ?TCost)

SQWRL queries

Q1-2: As of rectColumn longitudinal reinforcement
RectangleColumn(?Column) ^ hasNbar(?Column, ?CNbar) ^ ReinforcingBar(?RB) ^ hasDiameter(?RB, ?Diameter) ^ hasAs(?Column, ?CAs) -> sqwrl:select(?Column, ?CNbar, ?Diameter, ?CAs)
Q1-3: Ac of recColumn
RectangleColumn(?Column) ^ hasWidth(?Column, ?Width) ^ hasLength(?Column, ?Length) ^ hasAg(?Column, ?ColAg) ^ hasAs(?Column, ?ColAs) ^ hasAc(?Column, ?ColAc) -> sqwrl:select(?Column, ?Width, ?Length, ?ColAg, ?ColAs, ?ColAc)
Q10-1: Selecting concreteC25 column with size constraint
C25(?Con) ^ RectangleColumn(?Column) ^ hasConcreteC25(?Column, ?Con) ^ swrlb:lessThan(?Cb, 800) ^ hasWidth(?Column, ?Cb) ^ swrlb:lessThan(?Ch, 800) ^ hasLength(?Column, ?Ch) ^ meetDesignCondition(?Column, "Yes") -> sqwrl:select(?Column, ?Cb, ?Ch, ?Con)
Q11-1: Labor Cost of rectcolumn Concreting
RectangleColumn(?Column) ^ hasVolume(?Column, ?CV) ^ LaborConcretingCost(?LaborConCost) ^ hasLaborConcretingCost(?LaborConCost, ?hasLaborConCost) ^ swrlb:multiply(?LCC, ?CV,

?hasLaborConCost) ^ hasLaborCostConcreting(?Column, ?LCC) -> sqwrl:select(?Column, ?CV, ?hasLaborConCost, ?LCC)
Q11-2: Labor Cost rectcolumn Reinforcement
RectangleColumn(?Column) ^ hasWeightSteel(?Column, ?WS) ^ LaborReinforcementCost(?LaborReinfCost) ^ hasLaborReinforcementCost(?LaborReinfCost, ?hasLaborReinfCost) ^ swrlb:multiply(?LCR, ?WS, ?hasLaborReinfCost) ^ hasLaborCostReincorcement(?Column, ?LCR) -> sqwrl:select(?Column, ?WS, ?hasLaborReinfCost, ?LCR)
Q11-3: Labor Cost rectcolumn Formwork
RectangleColumn(?Column) ^ hasTotalAreaShulteringWork(?Column, ?TASHW) ^ LaborShulteringCost(?LaborShultCost) ^ hasLaborShulteringCost(?LaborShultCost, ?hasLaborShultCost) ^ swrlb:multiply(?LCF, ?TASHW, ?hasLaborShultCost) ^ hasLaborCostFormwork(?Column, ?LCF) -> sqwrl:select(?Column, ?TASHW, ?hasLaborShultCost, ?LCF)
Q11-4: Total Labor Cost rectcolumn
RectangleColumn(?Column) ^ hasLaborCostConcreting(?Column, ?LCC) ^ hasLaborCostReincorcement(?Column, ?LCR) ^ hasLaborCostFormwork(?Column, ?LCF) ^ swrlb:add(?TotalLCC, ?LCC, ?LCR, ?LCF) ^ hasTotalLaborCostColumn(?Column, ?TotalLCC) -> sqwrl:select(?Column, ?LCC, ?LCR, ?LCF, ?TotalLCC)
Q12-1: Total area of shuttering
RectangleColumn(?Column) ^ swrlb:multiply(?y, 2, ?Depth, ?Length, 0.001, 0.001) ^ swrlb:multiply(?x, 2, ?Depth, ?Width, 0.001, 0.001) ^ hasTotalAreaShulteringWork(?Column, ?TASHW) ^ hasWidth(?Column, ?Width) ^ hasLength(?Column, ?Length) ^ hasDepth(?Column, ?Depth) ^ swrlb:add(?TASHW, ?x, ?y) -> sqwrl:select(?Column, ?Length, ?Width, ?Depth, ?TASHW)
Q2-1: the ultimate axial load of a rectcolumn
RectangleColumn(?Column) ^ hasAs(?Column, ?CAs) ^ hasAc(?Column, ?ColAc) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasfckC25(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) ^ hasNedC25(?Column, ?CNed) -> sqwrl:select(?Column, ?ColAc, ?CAs, ?Confck, ?SBfyk, ?CNed)
Q2-2: the ultimate axial load of a rectcolumn
RectangleColumn(?Column) ^ hasAs(?Column, ?CAs) ^ hasAc(?Column, ?ColAc) ^ hasConcreteC35(?Column, ?Con) ^ C35(?Con) ^ hasfckC35(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) ^ hasNedC35(?Column, ?CNed) -> sqwrl:select(?Column, ?ColAc, ?CAs, ?Confck, ?SBfyk, ?CNed)
Q2-3: the ultimate axial load of a rectcolumn
RectangleColumn(?Column) ^ hasAs(?Column, ?CAs) ^ hasAc(?Column, ?ColAc) ^ hasConcreteC80(?Column, ?Con) ^ C80(?Con) ^ hasfckC80(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) ^ hasNedC80(?Column, ?CNed) -> sqwrl:select(?Column, ?ColAc, ?CAs, ?Confck, ?SBfyk, ?CNed)
Q2-4: the ultimate axial load of a rectcolumn
RectangleColumn(?Column) ^ hasAs(?Column, ?CAs) ^ hasAc(?Column, ?ColAc) ^ hasConcreteC90(?Column, ?Con) ^ C90(?Con) ^ hasfckC90(?Con, ?Confck) ^ ReinforcingBar(?SB) ^ hasfyk(?SB, ?SBfyk) ^ swrlb:multiply(?x, 0.576, ?Confck, ?ColAc) ^ swrlb:multiply(?y, 0.87, ?CAs, ?SBfyk) ^ swrlb:add(?CNed, ?x, ?y) ^ hasNedC90(?Column, ?CNed) -> sqwrl:select(?Column, ?ColAc, ?CAs, ?Confck, ?SBfyk, ?CNed)
Q3-1-1: meetDesignConditionC25
meetDesignConditionC25(?Column, "Yes") -> sqwrl:select(?Column)
Q3-2-1: Selecting a rectcolumns with load capacity larger than
RectangleColumn(?Column) ^ hasConcreteC35(?Column, ?Con) ^ C35(?Con) ^ hasWidth(?Column, ?Cb) ^ hasLength(?Column, ?Ch) ^ hasNedC35(?Column, ?CNed) ^ swrlb:greaterThanOrEqual(?CNed, 12000000) -> sqwrl:select(?Column, ?Cb, ?Ch, ?CNed)
Q4-1: Fire resistance time 60

swrlb:lessThan(?Width, 300) ^ RectangleColumn(?Column) ^ swrlb:greaterThanOrEqual(?Cover, 25) ^ hasFireResistanceTime60(?Column, "R60") ^ hasWidth(?Column, ?Width) ^ hasName(?Column, ?Name) ^ hasCover(?Column, ?Cover) ^ swrlb:greaterThanOrEqual(?Width, 200) -> sqwrl:select(?Column, ?Width, ?Cover, "R60")
Q4-2: Fire resistance time 90
hasFireResistanceTime90(?Column, "R90") ^ RectangleColumn(?Column) ^ swrlb:greaterThanOrEqual(?Width, 300) ^ swrlb:lessThan(?Width, 350) ^ swrlb:greaterThanOrEqual(?Cover, 25) ^ hasWidth(?Column, ?Width) ^ hasName(?Column, ?Name) ^ hasCover(?Column, ?Cover) -> sqwrl:select(?Column, ?Width, ?Cover, "R90")
Q4-3: Fire resistance time 120
hasFireResistanceTime120(?Column, "R120") ^ RectangleColumn(?Column) ^ swrlb:greaterThanOrEqual(?Width, 350) ^ swrlb:greaterThanOrEqual(?Cover, 25) ^ hasWidth(?Column, ?Width) ^ hasName(?Column, ?Name) ^ hasCover(?Column, ?Cover) -> sqwrl:select(?Column, ?Width, ?Cover, "R120")
Q7-1: Total embodied CO2e -C25
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasVolume(?Column, ?CV) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasEmbodiedCO2eC25(?Con, ?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) ^ hasTotalEmbodiedCO2eC25(?Column, ?TECO2) ^ meetDesignConditionC25(?Column, "Yes") ^ hasNedC25(?Column, ?CNed) -> sqwrl:select(?Column, ?Name, ?CV, ?Con, ?CNed, ?ECO2, ?TECO2)
Q7-2: Total embodied CO2e -C35
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasVolume(?Column, ?CV) ^ hasConcreteC35(?Column, ?Con) ^ C35(?Con) ^ hasEmbodiedCO2eC35(?Con, ?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) ^ hasTotalEmbodiedCO2eC35(?Column, ?TECO2) ^ meetDesignConditionC35(?Column, "Yes") ^ hasNedC35(?Column, ?CNed) -> sqwrl:select(?Column, ?Name, ?CV, ?Con, ?CNed, ?ECO2, ?TECO2)
Q7-3: Total embodied CO2e -C80
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasVolume(?Column, ?CV) ^ hasConcreteC80(?Column, ?Con) ^ C80(?Con) ^ hasEmbodiedCO2eC80(?Con, ?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) ^ hasTotalEmbodiedCO2eC80(?Column, ?TECO2) ^ meetDesignConditionC80(?Column, "Yes") ^ hasNedC80(?Column, ?CNed) -> sqwrl:select(?Column, ?Name, ?CV, ?Con, ?CNed, ?ECO2, ?TECO2)
Q7-4: Total embodied CO2e -C90
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasVolume(?Column, ?CV) ^ hasConcreteC90(?Column, ?Con) ^ C90(?Con) ^ hasEmbodiedCO2eC90(?Con, ?ECO2) ^ swrlb:multiply(?TECO2, ?CV, ?ECO2) ^ hasTotalEmbodiedCO2eC90(?Column, ?TECO2) ^ meetDesignConditionC90(?Column, "Yes") ^ hasNedC90(?Column, ?CNed) -> sqwrl:select(?Column, ?Name, ?CV, ?Con, ?CNed, ?ECO2, ?TECO2)
Q8-1: Site info
Siteinfo(?S) ^ hasSiteAddress(?S, ?SiteAddress) ^ hasGlobalId(?S, ?GlobalId) ^ hasName(?S, ?Name) -> sqwrl:select(?S, ?Name, ?GlobalId, ?SiteAddress)
Q8-2: Project info
Project(?P) ^ hasName(?P, ?Name) ^ hasLongName(?P, ?LongName) ^ hasPhase(?P, ?Phase) -> sqwrl:select(?P, ?Name, ?Phase, ?LongName)
Q9-1: total cost of columnC25
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasVolume(?Column, ?CV) ^ hasConcreteC25(?Column, ?Con) ^ C25(?Con) ^ hasCostC25(?Con, ?Cost) ^ hasTotalCostC25(?Column, ?TCost) ^ meetDesignConditionC25(?Column, "Yes") ^ hasNedC25(?Column, ?CNed) -> sqwrl:select(?Column, ?Name, ?CV, ?Con, ?CNed, ?Cost, ?TCost)
Q9-1: total cost of columnC35
RectangleColumn(?Column) ^ hasName(?Column, ?Name) ^ hasVolume(?Column, ?CV) ^ hasConcreteC35(?Column, ?Con) ^ C35(?Con) ^ hasCostC35(?Con, ?Cost) ^ hasTotalCostC35(?Column, ?TCost) ^ meetDesignConditionC35(?Column, "Yes") ^ hasNedC35(?Column, ?CNed) -> sqwrl:select(?Column, ?Name, ?CV, ?Con, ?CNed, ?Cost, ?TCost)

Appendix F: Names and descriptions of the atoms used in the reasoning rules and the proposed ontology

The names and descriptions of the atoms used in the proposed ontology.

Atom Type	Atom Name	Description
Class atom	RectangleColumn(?Column)	RectangleColumn(class) contains instance ?Column
	Project(?P)	Project(Class) contains instance ?P
	Concrete(?Con)	Concrete(class) contains instance ?Con
	ReinforcingBar(?RB)	ReinforcingBar(class) contains instance ?RB
	ResourceSupplier(?RS)	ResourceSupplier(class) contains instance ?RS
Object property atom	hasConcreteC25(?Column, ?Con)	Instance ?Column has Concrete C25 which contains instance ?Con
	hasConcreteC35(?Column, ?Con)	Instance ?Column has Concrete C35 which contains instance ?Con
	hasConcreteC80(?Column, ?Con)	Instance ?Column has Concrete C80 which contains instance ?Con
	hasConcreteC90(?Column, ?Con)	Instance ?Column has Concrete C90 which contains instance ?Con
Data property atom	hasName(?P, ?Name)	Instance ?P has name which contains value ?Name
	hasLongName(?P, ?LongName)	Instance 'P' has a long name that contains value ?LongName
	hasPhase(?P, ?Phase)	Instance ?P has phase which contains value ?Phase
	hasGlobalId(?P, ? GlobalId)	Instance ?P has global identifier which contains value ?GlobalId
	hasDescription(?P, ?Description)	Instance ?P has description which contains value ?Description
	hasSiteAddress(?S, ?SiteAddress)	Instance ?S has Site Address which contains value ?SiteAddress
	hasWidth(?Column, ?Width)	Instance ?Column has width which contains value ?Width
	hasLength(?Column, ?Length)	Instance ?Column has Length, which contains value ?Length
	hasAc(?Column, ?ColAc)	Instance ?Column has Ac width which contains value ?ColAc
	hasAg(?Column, ?ColAg)	Instance ?Column has Ag width which contains value ?ColAg
	hasDepth(?Column, ?Depth)	Instance ?Column has Depth which contains value ?Depth

hasVolume(?Column, ?CV)	Instance ?Column has a volume which contains value ?CV
hasVolAs(?Column, ?ColVolAs)	Instance ?Column has a volume which contains value ?ColVolAs
hasDensity(?Con, ?CD)	Instance ?Column has Density which contains value ?CD
hasWeight(?Column, ?CW)	Instance ?Column has Weight which contains value ?CW
hasWeightSteel(?Column, ?WS)	Instance ?Column has Weight which contains value ?WS
hasNbar(?Column, ?CNbar)	Instance ?Column has number of bars which contains value ?CNbar
hasDiameter(?RB, ?Diameter)	Instance ?RB has a diameter which contains value ?Diameter
hasAs(?Column, ?CAs)	Instance ?Column has area steel which contains value ?CAs
hasEmbodiedCO2eC25(?Con, ?ECO2)	Instance ?Con has EmbodiedCO2eC25 which contains value ?ECO2
hasTotalEmbodiedCO2eC35(?Column, ?TECO2)	Instance ?Column has TotalEmbodiedCO2eC25 which contains value ?TECO2
hasCostC25(?Con, ?Cost)	Instance ?Con hasCostC25 which contains value ?Cost
hasTotalCostC25(?Column, ?TCost)	Instance ?Column hasTotalCostC25 which contains value ?TCost
hasCompanyName(?RS, ?CN)	Instance ?RS has company Name which contains value ?CN
hasCertificateNo(?RS, ?CEN)	Instance ?RS has “certificate No”, which contains value ?CEN
hasRating(?RS, ?CR)	Instance ?RS has rating which contains value ?CR
hasPostCode(?RS, ?PC)	Instance ?RS has Post Code which contains value ?PC
hasFireResistanceTime(?Column, 60)	Instance ?Column has Fire Resistance Time which contains value 60
hasCover(?Column, ?Cover)	Instance ?Column has Cover which contains value ?Cover
hasLaborCostConcreting(?Column, ?LCC)	Instance ?Column has Cover which contains value ?LCC
hasLaborCostReincorcement(?Column, ?LCR)	Instance ?Column has Cover which contains value ?LCR
hasLaborCostFormwork(?Column, ?LCF)	Instance ?Column has Cover which contains value ?LCF

hasTotalLaborCostColumn(?Column, ?TotalLCC)	Instance ?Column has Cover which contains value TotalLCC
hasTotalAreaShulteringWork(?Column, ?TASHW)	Instance ?Column has Cover which contains value ?TASHW
hasNedC25(?Column, ?CNed)	Instance ?Column has Cover which contains value ?CNed
hasNedC35(?Column, ?CNed)	Instance ?Column has Cover which contains value ?CNed
meetDesignConditionC25(?Column, "Yes")	Instance ?Column has Cover which contains value "Yes"
meetDesignConditionC35(?Column, "Yes")	Instance ?Column has Cover which contains value "Yes")
hasFireResistanceTime60(?Column, "R60")	Instance ?Column has Cover which contains value "R60
hasFireResistanceTime90(?Column, "R90")	Instance ?Column has Cover which contains value "R90")
hasFireResistanceTime120(?Column, "R120")	Instance ?Column has Cover which contains value "R120")

Appendix G: Results collected from the proposed multi-objective knowledge base

The axial load capacity of various columns with various concrete strength

Name	UltAxialC25	UltAxialC35	UltAxialC80	UltAxialC90	Applied axial Load
Column-800 x 800mm	11706.86268	17220.18492	35597.92572	39273.47388	12000
Column-700 x 1000mm	12743.66268	18775.38492	38881.12572	42902.27388	12000
Column-800 x 1200mm	17236.46268	25514.58492	53108.32572	58627.07388	12000
Column-500 x 800mm	7559.66268	10999.38492	22465.12572	24758.27388	12000
Column-600 x 600mm	6868.46268	9962.58492	20276.32572	22339.07388	12000
Column-1000 x 1000mm	17927.66268	26551.38492	55297.12572	61046.27388	12000
Column-500 x 1000mm	9287.66268	13591.38492	27937.12572	30806.27388	12000
Column-300 x 750mm	4535.66268	6463.38492	12889.12572	14174.27388	12000

Total embodied CO2 and Total material cost while considering load capacity criteria

Name	TEmbodie d25	TCost25	TEmbodied 35	TCost35	TEmbodied 80	TCost80	TEmbodied 90	TCost90
Column-1000 x 1000mm	1073.97	387.26	1221.693	379.28	1133.85	966.176	1265.61	1038.040
Column-700 x 1000mm	751.172	270.8	854.493	265.28	793.059	675.776	885.211	726.040
Column-800 x 1200mm	1030.93	371.749	1172.7	364.08	1088.419	927.456	1214.891	996.440
Column-500 x 1000mm			609.693	189.28	565.8597	482.176	631.6110	518.04
Column-800 x 800mm			781.05	242.48	724.8997	617.696	809.131	663.640
Column-600 x 600mm					406.819	346.656	454.0910	372.440
Column-500 x 800mm					452.2597	385.376	504.811	414.04
Column-300 x 750mm					253.459	215.976	282.911	232.040

Total labour cost based on labour cost concreting, reinforcement and formwork

Column	LC concreting	LC reinforcement	LC formwork	Total LC
Column-800 x 800mm	25.52464	59.1576	38.4	123.0822
Column-1000 x 1000mm	39.92464	59.1576	48	147.0822
Column-500 x 1000mm	19.92464	59.1576	36	115.0822
Column-700 x 1000mm	27.92464	59.1576	40.8	127.8822
Column-600 x 600mm	14.32464	59.1576	28.8	102.2822
Column-800 x 1200mm	38.32464	59.1576	48	145.4822

Column-500 x 800mm	15.92464	59.1576	31.2	106.2822
Column-300 x 750mm	8.92464	59.1576	25.2	93.28224