Technische Universität München

TUM School of Engineering and Design

Chair of Computational Modeling and Simulation

An end-to-end approach to productivity assessment in Digital Twin Construction

Scientific work to obtain the degree

Master of Science (M.Sc.)

at the TUM School of Engineering and Design of the Technical University of Munich

Supervised by [redacted]

Submitted by [redacted]

Matriculation Number [redacted]

Submitted on [redacted]

Abstract

The thesis proposes an end-to-end approach for assessing productivity in Digital Twin Construction (DTC) by integrating as-planned and as-performed data within a unified framework through the use of graph databases. It builds on existing methods by leveraging BIM models, BIM transformation tools and RDF graphs to integrate and compare real-time performance with initial project plans for decision making and course-correction in construction project management. By utilizing graph databases, the research enables efficient querying and analysis of construction data, which could offer new insights for decision-making and project optimization.

The methodology developed in this work focuses on converting IFC models to RDF graphs to create a knowledge graph that links as-planned data with on-site, as-performed information. This process supports real-time tracking and productivity analysis, allowing for more accurate comparisons between the planned and actual progress. The thesis demonstrates the viability of the approach through a case study, applying the methodology to a real construction project, where both planned and performed data were collected and analysed, allowing for the adjustment of the construction plan and schedule dynamically.

This research addresses a critical gap in the construction industry by providing a comprehensive end-to-end solution for productivity assessment. The integration of data sources and continuous updates enhances the capability to detect deviations, recalibrate project timelines, and make informed management decisions, ultimately contributing to more efficient project execution and resource utilization.

Keywords: Digital Twin Construction, Building Information Modelling, Industry Foundation Classes, Construction Productivity, Real-Time Monitoring, Knowledge Graphs, Construction Management.

Acknowledgement

[redacted]

Contents

List of Figures

List of Tables

Acronyms

AECO	Architecture, Engineering, Construction and Operation
ANN	Artificial Neural Networks
BIM	Building information modeling
BOT	Building Topology Ontology
CDE	Common Data Environment
CPM	Construction Project Monitoring
CNN	Convolution Neural Network
DT	Digital Twin
DTC	Digital Twin Construction
IFC	Industry Foundation Classes
IFCtoRDF	Industry Foundation Classes to Resource Description Framework
RDF	Resource Description Framework
KPI	Key Performance Indicator
LPS	Last Planner System
PDCA	Plan-Do-Check-Act
TTL	Terse RDF Triple Language

Chapter 1

Introduction

Construction productivity has long been a challenge, directly impacting project costs, timelines, and overall project efficiency. The construction industry is notorious for its lower productivity rates compared to other sectors such as manufacturing (McKinsey Global Institute, 2017). Delays in construction projects often lead to significant cost overruns, sometimes as high as 80% beyond the initial budget in large infrastructure projects (Flyvbjerg et al., 2002). These inefficiencies not only strain project budgets but can also delay the delivery of essential services, affecting public infrastructure projects like transportation networks, healthcare facilities, and housing developments. This creates a ripple effect, with implications ranging from the loss of investor confidence to reduced quality of life for communities (Flyvbjerg et al., 2002).

Moreover, construction productivity challenges exacerbate environmental and economic concerns. The construction sector is one of the most resource-intensive industries, consuming vast quantities of raw materials and energy (Deng et al., 2016). When projects are delayed or run inefficiently, this results in excessive resource usage, contributing to environmental degradation and higher greenhouse gas emissions. Additionally, inefficient project delivery can lead to missed economic opportunities, as delayed infrastructure can hamper business operations, restrict mobility, and limit access to services in a society level (Thomas & Mathews, 2016).

In construction progress monitoring (CPM), traditional methods involve manual and work-intensive processes for periodic information gathering, documenting, and reporting (Hegazy, 2002). The reports generated from this process are used for project monitoring and control, compared to the planned schedule, and act as a project status record through the project lifecycle (Son et al., 2015).

Accurate progress reporting is crucial for keeping stakeholders informed and facilitating effective decision-making to prevent delays and cost overruns (El-Sabek et al., 2017). Nevertheless, traditional reporting methods are cumbersome, prone to errors, slow, and often contain redundant information, hindering proactive and timely decision-making by stakeholders (Navon et al., 2022).

The exploration of Digital Twin Construction (DTC) and productivity assessment within the construction industry encompasses diverse methodologies and technological advancements. Current approaches aim to disrupt traditional construction progress monitoring by incorporating data-centric systems, artificial intelligence, data analytics, engineering simulations and innovative monitoring techniques in a holistic approach (Sacks et al., 2020).

A data-centric approach in the Architecture, Engineering, Construction, and Operation (AECO) industry often stems from Building Information Modeling (BIM), which can be regarded as a process of creating models with semantically rich information in a common data environment (CDE) to accelerate digitalization in the AECO industry (Pan et al., 2020). Considering this definition, BIM can be understood as a data repository to store large data collected from data-rich objects, tools and other sources during project execution (Peng et al., 2017).

Prior research proposes different techniques for data collection regarding construction monitoring, some of which involve the use of object detection, deep learning and crane cameras (Pfitzner et al., 2024), the use of real-time data collection from positioning sensors (Cheng et al., 2023), the use of photogrammetric reconstruction, semantic web technologies, and RDF databases (Collins et al., 2022), the use of perspective alignment to match real-time images with the corresponding BIM models, an automated BIM-based construction progress monitoring approach, using three-dimensional point clouds derived from images captured by aerial vehicles (Braun, A., 2020), amongst others.

While these contributions are valuable, a comprehensive approach for productivity assessment that addresses the steps after progress data collection is lacking. Current methodologies often fall short of providing a holistic end-to-end solution that combines monitoring aspects to obtain productivity insights and potential course-correction for delay recovery and optimization of construction processes based on the specificities of the project being monitored.

1.1 Motivation

The main motivation behind this thesis consists of an end-to-end approach to productivity assessment and decision making in construction, stemming from the limitations regarding data-driven solutions for construction progress assessment and monitoring, as discussed in the previous section.

This end-to-end approach should involve a pipeline from the monitoring of construction progress to the analysis of collected data for decision making, potential course correction of the project management strategy and update of the construction schedule based on the insights obtained from the data collection and analysis.

The approach also relies on the utilization of graph databases, for storage and querying of the information collected regarding the as-planned and as-performed versions of a construction project. The graph databases provide a clear representation of the relationships between the elements carrying the necessary information for decision making regarding the construction progress and project management.

With this in mind, this thesis has the goal of verifying the effectiveness of the combination of existing approaches that address different phases of this pipeline in a holistic, end-to-end solution for analyzing and optimizing construction processes.

1.2 Research Objective

The objective of this thesis can be subdivided into two main categories.

1. Propose an end-to-end approach to productivity assessment in Digital Twin Construction, with focus on the development of data pipeline correlating as-planned and as-performed information to support decision-making and course correction regarding construction project management.
2. Evaluate the use of RDF-graphs in storing and querying productivity related information for scheduling and definition of the necessary efforts for the execution of different types of built elements of construction projects.

1.3 Structure of Thesis

Chapter 2 summarizes the fundamental concepts surrounding the thesis topic, detailing construction progress monitoring, lean construction, building information modeling, digital twin construction and the deep learning techniques and graph databases necessary for the understanding of the methodology.

Chapter 3 explores the state-of-the-art in fields and topics used as a basis for this thesis work, dividing the collection of references into the acquisition and processing of as-planned data, as well as as-performed data, essential for productivity assessment and comparison within the proposed methodology. This chapter also highlights the research gaps from them, which are being addressed in the thesis. The works explored in this chapter are directly referenced and utilized in the methodology.

Chapter 4 presents specifics of the thesis work, detailing the overall steps of the proposed methodology, including a processing pipeline for as-planned data, a processing pipeline for as-performed data, and graph operations to connect the two for productivity assessment. The possibility of construction schedule update is also discussed.

Chapter 5 describes a case study in which the proposed methodology in Chapter 4 was applied to a specific construction project. The chapter provides a step-by-step description of the methodology, highlighting the specific tools that were utilized for each of them, as well as the results obtained from the application, in order to verify the feasibility of the proposed methodology in this work.

Chapter 6 discusses the results, and the insights obtained from the application of the methodology in the case study presented. It also talks about the limitations of the approach and discusses what could be further refined and explored for the improvement of the desired results.

Finally, Chapter 7 summarizes key findings, provides a conclusion and offers suggestions for future research regarding the topic.

Chapter 2

Theoretical Background

2.1 Construction Progress Monitoring

Construction progress monitoring (CPM) involves the continuous analysis and comparison of the actual progress and the planned progress. If the result is consistent, the project should continue to be implemented according to the schedule. If there are deviations, their causes should be analyzed, and timely, reasonable measures should be taken to minimize their impact on the overall progress and effectiveness of construction. (Fang et al., 2018).

2.1.1 Traditional CPM

Traditional construction progress monitoring (CPM) involves manual and labor-intensive methods for documenting and reporting the status of a construction project at regular intervals (Hegazy, 2002). These reports are crucial for tracking and managing the project according to the planned schedule serve as a record of the construction process. Accuracy is essential for keeping stakeholders informed about the project's status, enabling them to make informed decisions to prevent delays, budget overruns, and prepare for delay claims (El-Sabek et al., 2017).

Soibelman et al. (2000) critique the traditional methods of construction progress monitoring, highlighting their dependence on manual data collection, which is both time-consuming and prone to errors. They argue that these conventional techniques often result in delays in identifying project deviations and inefficiencies in corrective action implementation. The manual processes not only slow down decision-making but also create discrepancies in the accuracy of progress reports, which can hinder effective project management. Soibelman & Kim (2002) highlight that traditional construction progress monitoring methods lack the ability to effectively analyze and extract deeper insights from the data. As a result, traditional methods are not well-equipped to handle the large volumes of data generated in modern construction projects.

However, poor progress reporting methods and coordination on construction sites—partly due to reporting errors, delays, and redundant information—hinder proactive decision-making by stakeholders and are a major reason for projects exceeding their budgets and timelines (Wolfe et al., 2020).

2.1.2 CPM Technologies

Researchers have developed and implemented various technological advancements for automated monitoring of construction activities, with some now being commercially utilized, as shown in Table 1. These technologies have typically been applied in a segmented manner in the construction industry. Most applications focus on a single aspect, while integration of multiple monitoring technologies is rare (Sacks et al., 2020). Table 1 summarizes data acquisition technologies highlighted by Sacks et. al (2020), including the hardware associated with that technology and the common applications for it.

Table 1: Data acquisition technologies applied to monitoring construction (Sacks et al., 2020)

Technology	Hardware	Common Applications
Electronic location and distance measurement	Robotic total stations, range finders, and so forth. Laser scanning	Record current state of construction
Global Positioning System (GPS)	Differential GPS readers	Locate and measure work done; track production progress
Computer Vision (stills and video)	Video, stills, 360o images	Safety; production progress: labor: and equipment
Audio and sonar	Microphones	Identify equipment function and use
Tag identification systems	Bluetooth Low Energy (BLE), radio-frequency identification (RFID), barcodes	Track materials; worker locations and durations; quantity and quality
Communication networks	Wi-Fi, ultra-wideband (UWB), cellular	Material tracking; worker locations and durations; safety
Smart sensors and sensor networks	Temperature, himidity, pressure, strain, rotation; IoT, edge computing	Monitor construction quality; monitor structural health: monitor safety

The field of automated progress monitoring in construction is currently a very active area of research. Vision-based methods, in particular, have become increasingly popular (Fini et al., 2022).

Review papers related to this topic have honed in on particular elements of the construction progress monitoring process. Omar et al. have explored various data acquisition technologies (2016), Ekanayake et al. (2021) and Patel et al. (2021) have studied the existing literature, pointing out specific challenges that need to be addressed, such as managing the computational load to process data in real-time, occlusion and visual obstacles, indoor environment challenges due to varying lighting conditions and camera movements disrupting continuous monitoring, whereas Ma et al. (2018) have extensively focused on methods such as photogrammetry, laser scanning, and structure from motion for 3D reconstruction.

At present, there lacks a cohesive framework that covers all aspects of Computer Vision-Based Construction Progress Monitoring (CV-CPM), from the initial data collection to the estimation of progress (Reja et al., 2022). Figure 1 illustrates the stages of vision-based progress monitoring, which generally starts with the data acquisition, moving on to the 3D reconstruction, then a comparison phase between the as-planned and as-performed models, then a progress visualization and quantification based on the previous comparison. This conceptualization, however, doesn't detail and specific which methods and techniques could be applied for each of the presented steps.

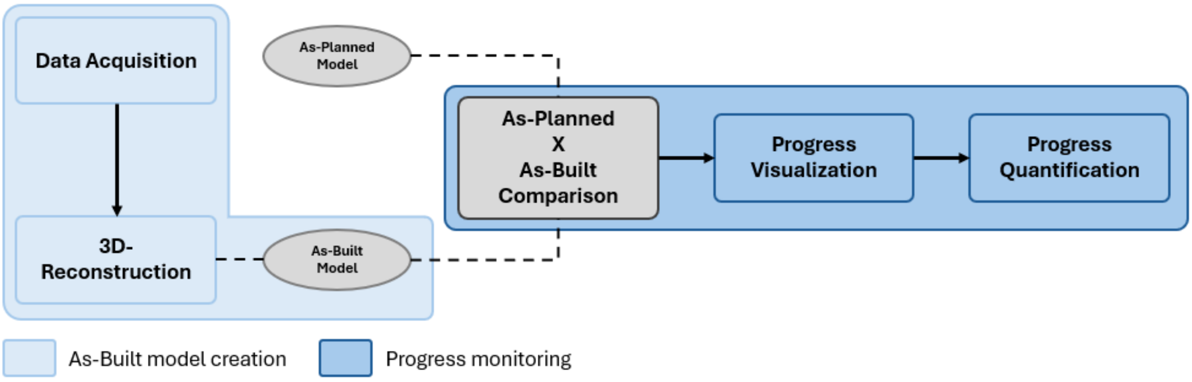


Figure 1: Conceptualization of vision-based progress monitoring (adapted from Reja et al., 2022)

2.2 Lean Construction

Lean construction arises from implementing a modern production management approach in the construction sector. Its key characteristics involve establishing distinct goals for the delivery process to optimize customer satisfaction at the project level, simultaneous product and process design, and continuous production control from the design phase through to delivery. Lean construction prioritizes achievement of smooth production flows with minimal variation and thus minimal waste (Howell, 1999).

Ballard (2000) describes The Last Planner System (LPS) as a method of production control aimed at improving the consistency and predictability of workflows in construction projects. It emphasizes aligning what should be done with what can be done, ensuring that work is ready to be executed without delays. Central to the system are the "Last Planners," typically foremen or team leaders, who commit to completing specific tasks, thereby turning planned work into completed actions. LPS integrates planning with control mechanisms to enhance project performance.

Ballard (2000) also highlighted the importance of drilling down the planning process during construction, from the master schedule to the daily activity, readjusting the premise when new information about the construction process is identified throughout this phase. Figure 2 illustrates this concept, showcasing the different granularities of planning, the expected activities in each of them and how they interact with each other.

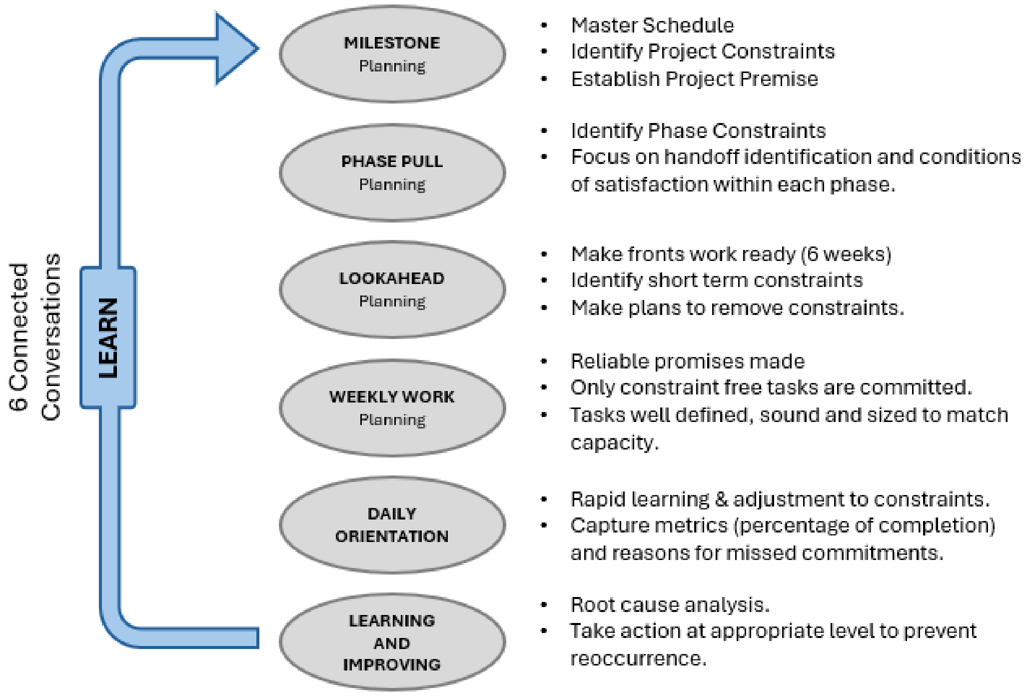


Figure 2: Successful implementation of Last Planner System (adapted from PBC Today, 2023)

Sacks et al. (2010) highlights a significant synergy between lean construction and BIM and digitalization The integration of LPS and BIM helps in improving communication and collaboration among stakeholders, ensuring that project workflows are more synchronized. Together, LPS and BIM provide a more holistic approach to managing construction projects, aligning physical and digital processes.

The Plan-Do-Check-Act (PDCA) cycle, a staple of lean construction (Deming, 1982), is particularly relevant in the context of digital twin systems in construction, in which the PDCA cycles can be used as a tool for construction planning and control (Forbes et al., 2011). The LPS integrates PDCA cycles at various planning stages, with the 'Check' phase demanding detailed updates on all constraints affecting tasks, a challenge particularly pronounced at the look-ahead planning stage (Hamzeh et al., 2015).

2.3 BIM-based Progress Monitoring

Building Information Modeling (BIM) revolves around the continual application of digital models for the full lifespan of a facility, from the initial conceptual and detailed design phases, through construction, and extending into the operational stage. BIM enhances the flow of information among all participants involved, boosting efficiency by minimizing the manual and error-prone re-entry of data prevalent in traditional paper-based processes (Borrmann et al., 2018). It creates an accurate 2D or 3D geometry and adds semantic information about the built asset (Braun et al., 2018).

In the construction phase, BIM provides substantial benefits in planning and construction. A 4D model links each component with its respective construction timelines, allowing for the verification of the construction sequence, identification of clashes, and organization of site logistics. Furthermore, a 5D model incorporates cost data, facilitating simulations of cost variations over time. Additionally, BIM can support the billing of tasks and issue management throughout the project (Borrmann et al., 2018).

A series of studies highlight the importance of the integration of Building Information Modeling (BIM) with progress monitoring technologies. Turkan et al. (2012) contributed to the topic by focusing on the integration of BIM with reality capture tools such as laser scanning and photogrammetry. Bosché & Guenet (2014) leverage computer vision and 3D point cloud data to automate construction monitoring to facilitate automated comparisons between the actual site conditions and planned 3D models.

Golparvar-Fard et al (2011) focuses on photogrammetry, 3D reconstruction, and visual data analytics to automate the generation of as-built models, emphasizing the integration of real-time visual data with BIM. Teizer et al. (2010) used real-time data collection systems, such as RFID and GPS, to track the location and movement of construction site resources integrating the collected data with the BIM model to enhance safety and productivity. Finally, Fang et al. (2016) explored the application of robotics and computer vision for construction monitoring, as well as the use of algorithms for detecting and quantifying progress from site images and sensor data, also from the standpoint of BIM integration.

BIM-based progress monitoring enables the automated identification of discrepancies between the actual and planned states of construction, allowing for early detection of deviations in the building process, often captured through photogrammetric surveys. These surveys produce dense point clouds by merging disparity maps generated, which are later compared to the desired state outlined in a 4D model. The process is enhanced by considering the process- and dependency-relations provided by the BIM model, further refining the detection of discrepancies (Borrmann et al., 2018).

Figure 3 represents a conceptualization of BIM-based progress monitoring process.

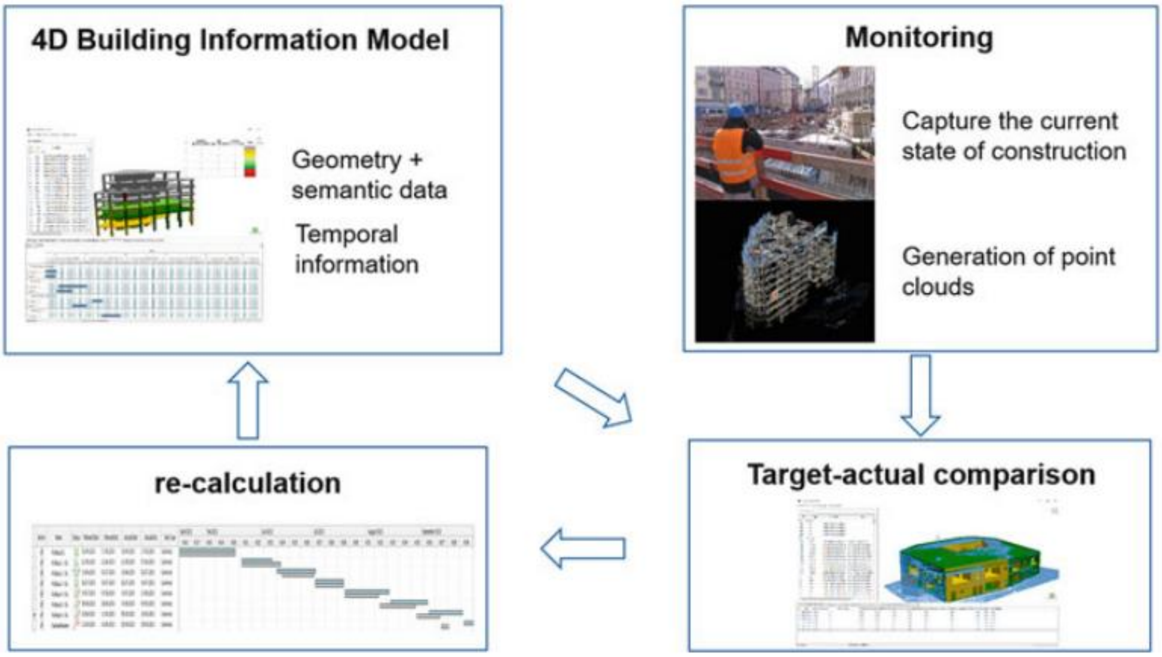


Figure 3: Concept for automated progress monitoring (Braun et al, 2015)

2.4 Digital Twin Construction

Digital Twins (DT) in the construction sector refer to virtual replicas of physical assets, integrating real-time data from IoT devices and BIM models to enhance monitoring and optimization (Boje et al., 2020). DT simulates and reflects the object's state throughout its lifecycle. Successful DT implementation requires advanced software, sensors, and communication tools, enabling synchronized real-time coordination between physical and virtual objects (Bui et al., 2022). The DT concept has predominantly been applied to the operation and maintenance phases of the built environment (Bew et al., 2016). However, DT in construction is still emerging, but it has a potential to revolutionize project management, by enhancing decision-making (Boje et al., 2020).

For the construction phase, there are many applications. For example, Digital Twins can be used for the analysis of structural system integrity at various stages, by continuously monitoring data from sensors embedded within the building elements and analyzing stresses, deformations, and load distribution at various stages of construction (Angjeliu et al., 2020). Digital Twins can also be used in the production of precast concrete parts, demonstrating the importance of real-time data integration for adaptive construction techniques (Gerhard, D. 2020) and creation of as-performed models, providing stakeholders with information for managing the built environment effectively. The as-performed models can subsequently by used for increased efficiency in the operational and maintenance phases (Macchi, M., 2018). These insights are crucial for improving construction processes and managing resources more efficiently.

Digital Twin Construction (DTC) represents a novel concept proposed by Sacks et al., 2020, to construction management that utilizes data from various site monitoring technologies and artificial intelligence capabilities. This approach allows for precise status updates and proactive optimization of design, planning, and production processes. DTC incorporates BIM, principles of lean construction and AI to create a data-driven construction management system.

Digital Twin Construction (DTC) focuses on the construction phase, using real-time data from IoT devices and sensors to dynamically model and monitor ongoing construction projects. Thus, while DT encompasses the entire asset lifecycle, DTC is specifically tailored to improve construction production management (Sacks et al. 2020).

The DTC approach is centered around data usage, in contrast to traditional practices where data is siloed and manually gathered, integrating real-time data from various monitoring technologies into decision-making through systematic PDCA cycles, enhanced by AI for tasks ranging from planning to optimization (Sacks et al., 2020).

2.5 Deep Learning for Image Classification

Deep learning models, which excel in accuracy over traditional machine learning algorithms, have shown significant improvements in image classification due to recent advances in hardware and network architectures. (Yu, 2022).

Traditional image classification methods, categorized under machine learning, include modules for feature extraction and classification but are limited in the range of features they can extract, impacting their effectiveness. Deep learning addresses this by using artificial neural networks (ANNs) that mimic the human brain to enhance data analysis and feature extraction capabilities (Lee et al., 2009). This advanced approach enables neural networks to better identify images by developing sophisticated models that can discern and extract a wider variety of features from a dataset (Krishna et al., 2018).

Neural networks, particularly Convolutional Neural Networks (CNNs), offer efficient automated methods for almost real-time analysis of construction images, especially for image classification tasks (Rehman et al., 2022; Hussain et al. 2018). CNNs incorporate image-specific features into their architecture, enhancing efficiency for image tasks and reducing parameter counts. Unlike ANNs, CNNs efficiently handle the larger computational demands of complex image data, making CNNs more suitable for image analysis (O’Shea et al., 2015). Other advantages of CNNs include weight sharing that further minimizes parameters, and pooling layers that down sample data to decrease volume and simplify input while maintaining crucial information (Li et al., 2021).

Figure 4 illustrates a simple CNN architecture comprised of five layers, as proposed by O’Shea et al (2015).

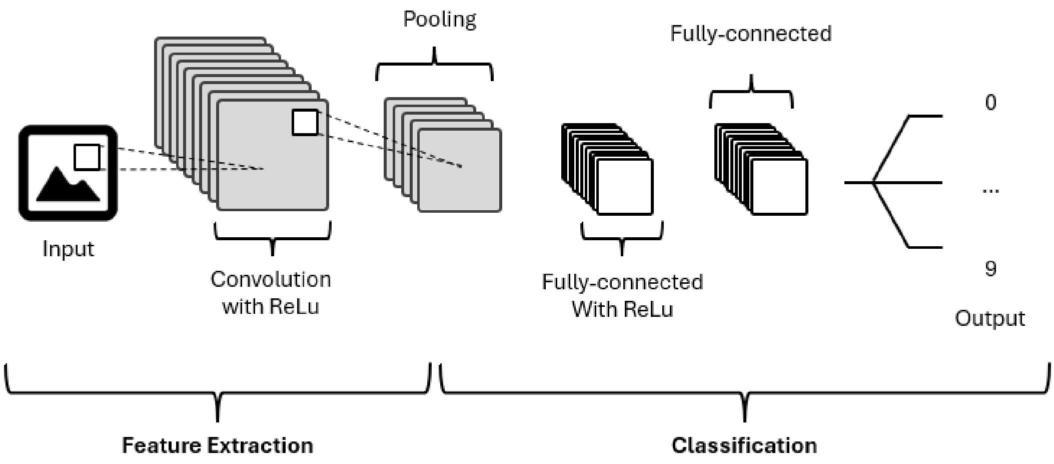


Figure 4: A simple CNN architecture, comprised of just five layers (adapted from O'Shea et al., 2015)

Khallaf & Khallaf (2021) provides a comprehensive review of deep learning applications in the construction industry. It analyzes 80 journal papers, identifying six major application areas such as equipment tracking, crack detection, construction work management, sewer assessment and 3D point cloud enhancement. Khallaf & Khallaf (2021) highlight the strengths of deep learning in automating tasks like crack detection, equipment monitoring, and infrastructure assessment, which can lead to more efficient construction processes.

Khallaf & Khallaf (2021) describe CNNs specifically as one of the most widely used deep learning techniques in construction applications. CNNs excel in feature extraction, particularly in processing visual data like images and videos, making them ideal for tasks such as crack detection, equipment monitoring, and worker tracking. CNNs have been used in various studies to automate traditionally manual processes, such as detecting structural cracks in concrete and tracking construction equipment (Cha et al, 2017). While CNNs significantly improve efficiency and accuracy in these areas, there are also challenges, including the need for large datasets and the risk of false positives or negatives in image classification tasks (Khallaf & Khallaf, 2021).

2.6 Knowledge Graphs

Graph databases represent data as nodes (entities) and relationships (connections between entities), enriched with properties (key-value pairs) to store relevant information (Hitzler et al., 2007). These databases excel in handling complex queries involving relationships, offering superior performance and flexibility compared to traditional relational databases (Robinson et al., 2013). Graph database models organize data and schema in graph formats, manage data via graph-oriented operations, and enforce data consistency through various integrity constraints (Angles, 2008).

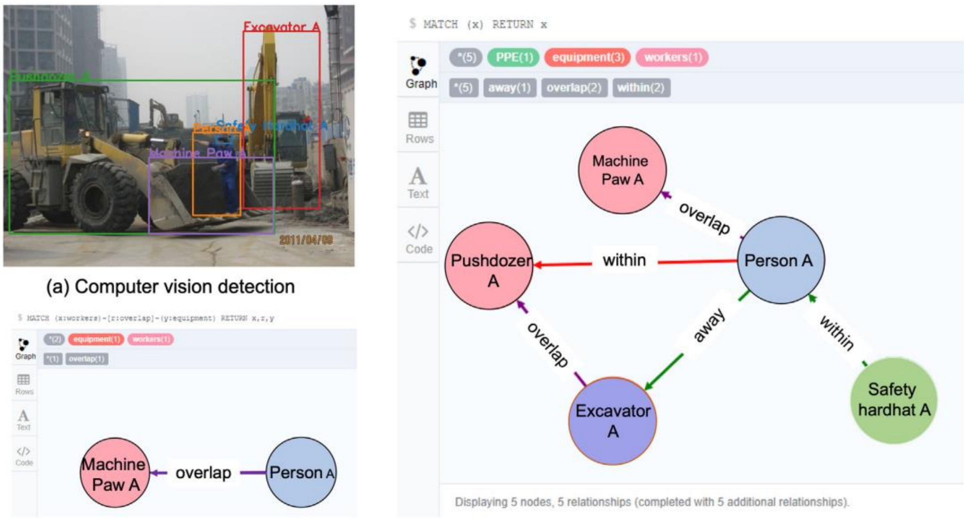
A knowledge graph is a structured semantic network composed of nodes representing concepts or specific entities such as people, places, or organizations, and edges that depict relationships between them. By integrating fragmented concepts and entities, knowledge graphs form comprehensive repositories that enhance the management and understanding of information (Yan et al., 2018).

Resource Description Framework (RDF) graphs require a strict node-relationship structure using statements expressed as triples provided by a schema (Segaran et al., 2009). Labeled Property Graphs (LPG), alternatively, offer complete flexibility: each node can be labeled uniquely, and nodes with identical labels can possess different properties. RDF graphs are especially valuable for ensuring interoperability among different systems, making them ideal for integrating and querying data from multiple sources in a standardized manner (Needham & Hodler, 2019).

Since Labeled Property Graphs (LPG) are largely schema-free, they provide practical advantages such as qualifying instances of relationships and representing data that closely mirrors real-world logical models. This flexibility enables them to adapt more easily to the changing and evolving data structures, making it well-suited for monitoring progress and resources on a construction site (Pokorný, 2015).

In the AEC industry, graph databases can significantly enhance various processes. For project management, nodes represent different entities such as projects, tasks, and resources, while relationships capture dependencies between tasks, resource allocations, and team members (Pauwels et al., 2017). El-Diraby et al. (2005) proposes a logic to model the graphs for Building Information Modeling (BIM) elements, in which, nodes denote components of a building (walls, doors, windows) and stakeholders (architects, contractors), with relationships illustrating connections like component dependencies, workflows, and stakeholder interactions. Additionally, Tserng et al. (2012) proposes a logic for supply chain management, in which nodes include suppliers, materials, construction sites, and logistics providers, and relationships depict the flow of materials, supplier dependencies, and delivery schedules.

Another application of knowledge graphs specifically to construction sites was proposed by Fang et al. (2020), who presented a framework that combines computer vision with ontologies to identify safety hazards on construction sites. The knowledge graph leverages computer vision to detect entities (e.g., workers, equipment) and extracts spatial relationships to infer potential hazards. The system allows real-time detection of risks, such as falls from heights, through an ontology model that adapts to changing safety regulations, improving hazard identification beyond standard computer vision approaches. Figure 5 illustrates the reasoning process of the identification of unsafe conditions using the knowledge graph, demonstrating a query matching a worker's position overlapping with equipment, indicating unsafe behavior.



(a) Hazard identification by reasoning (b) Data modeling

Figure 5: The reasoning of unsafe conditions with knowledge graphs (Fang et al., 2020)

The knowledge graph information can be queried and manipulated with CYPHER notation, a powerful query language for the graph databases, designed specifically for expressing graph patterns and data retrieval in a concise and efficient manner (Robinson et al., 2013; Needham & Hodler, 2019).

The information about productivity of the building elements stored and queried in these graphs can then be utilized in data analytics and machine learning models analyse the characteristics of building elements provides valuable insights into construction delays. By examining the aforementioned factors, one can identify patterns and outliers affecting construction timelines. For instance, machine learning algorithms can correlate complex data from multiple sources to pinpoint whether specific materials perform poorly in certain weather conditions or if particular team configurations lead to inefficiencies. This comprehensive analysis enables targeted interventions to improve overall project performance (Bilal et al., 2016). Such comprehensive evaluation enables targeted interventions to optimize construction processes (Bock & Linner, 2016).

2.7 Productivity Calculation

The general productivity rate for building elements involves dividing the total work output to complete the construction of said element by the total time necessary to achieve this result. These productivity rates are usually assessed in terms of a geometric dimensions (e.g. m2 or m3) divided by the man-hours necessary to perform this geometric unit. The general productivity rate formula can be expressed as presented below (Peurifoy et al., 2011):

$$Productivity\ Rate = \frac{\text{Total Work Output}}{\text{Total Time}}$$

The total work output varies depending on specificities of each type of building element. For reinforced concrete columns (circular or square), productivity is based on concrete volume and reinforcement density, considering formwork complexity and concrete placement efficiency (Peurifoy et al., 2011). Metallic column productivity involves fabrication, transportation, and installation, with key variables including column weight, installation height, and worker skill (Peurifoy et al., 2011).

Reinforced concrete wall productivity depends on wall area, thickness, formwork system, rebar density, design complexity, and curing time (Peurifoy et al., 2011). Drywall productivity is measured in square meters per hour, influenced by drywall type, area coverage, cuts, and joints, with efficiency enhanced by pre-cut panels and minimized seams (Peurifoy et al., 2011). Brick wall productivity, calculated by bricks laid per hour, is affected by brick type, mortar, mason skill, wall height, pattern, and working conditions (Peurifoy et al., 2011).

Finally, reinforced concrete slab productivity is determined by slab area and thickness, considering formwork system, concrete placement method, and reinforcement density, with guidance from detailed tables and examples (Peurifoy et al., 2011).

Chapter 3

Related Work

This chapter discusses research that forms the basis of this work, covering as-planned and as-performed data collection and processing. Each of them requires different steps and resources and are necessary for productivity assessment and comparison in the context of the proposed methodology. Section 3.1 discusses research necessary for the conceptualization of the as-planned data processing and representation step of the proposed pipeline, while Section 3.2 discusses research necessary for the conceptualization of the as-performed step.

Once these two data sources are integrated in a graph database, comparisons between the planned and performed productivity can be made to assess and recalibrate potential premises deployed in the original estimation of the efforts to execute construction projects.

In the context of Digital Twin Construction, as-performed models capture the real-time progression of construction activities, reflecting the actual performance of tasks as they occur. This contrasts with as-planned models, which outline the intended sequence and methodology of construction tasks, and as-performed models, which represent the completed structure. The integration of as-performed data into digital twins, representing the as-planned models, allows for continuous updates and monitoring, providing insights into deviations and comparison from the plan and enabling more accurate and efficient project management (Sacks et al., 2020).

Then, this Chapter discusses the identified research gap that is being addressed with this thesis in section 3.3.

3.1 As-Planned Data Representation

3.1.1 IFC Model to RDF Graph Conversion

BIM models can integrate various data sources and allow for comprehensive analysis and planning during the project's initial phases (Kavaliauskas et al., 2022). These models can be converted to Industry Foundation Classes (IFC) format, a standardized, digital data schema used for representing building and construction information across various software platforms, allowing for data exchange enhances collaboration and efficiency throughout the project lifecycle (Borrmann et al., 2018).

The further conversion of the IFC model into a Resource Description Framework (RDF) Format is necessary so that the IFC model information can be read as a graph. This can be achieved with the tool IFCtoRDF. RDF is a framework for representing information on the web in a structured, machine-readable format, often used in semantic web technologies (Jing et al, 2023). The conversion of IFC to RDF is significant because it allows the rich, complex data embedded in IFC files to be represented in a semantic, interconnected way that is compatible with web technologies, and BIM to be integrated with other data sources (Bonduel, et al., 2018).

However, for the purpose of productivity assessment, not all the data in the IFC model is necessary. Tools such as SimpleBIM can be used for IFC data wrangling and reduction of the IFC model to only the necessary information for this use case, for optimization purposes (Lennox, 2022). SimpleBIM is an open BIM tool that allows users to standardize, enrich, and manage BIM models, making them suitable for various use cases. This tool is particularly useful for cleaning up, filtering, and organizing IFC. It enables the creation of lightweight models that are easier to manage. This leads to improved efficiency and better project outcomes (Day, 2022).

3.1.2 Knowledge Graph for the As-Planned Model

Ontologies define how and where data is stored in the graph by providing a schema. As-performed data can be stored similarly using a tailored graph meta-model, classified into categories such as human resources, equipment, materials, and environment (Fang et al., 2020). A data model supports structuring the low-level information acquired from the construction site and simplifies obtaining higher-level information at a later stage, which will be driven to the finest level of detail (Zheng et al., 2021).

The conversion of the IFC file with the IFCtoRDF tool is initiated with a conversion into ifcOWL-based RDF graphs and their simplification according to the modular ontologies. The IFCtoRDF converter simplifies the data structure, reducing the number of RDF triples and making the output more concise and easier to query. This approach is aimed at improving usability in Linked Data applications within the Architecture, Engineering, Construction, and Operation (AECO) domain (Bonduel et al., 2018).

Once the IFC data is converted into RDF graph format, it can be used to improve interoperability, querying, and integration with other linked data sources (Bonduel, et al., 2018). Bonduel et al (2018) and Pauwels et al. (2017) also argue that the converted information becomes easily accessible by SPARQL queries, allowing for better performance, the RDF graph can also be linked to other external data sources, such as geographic information, product databases or regulatory data, allowing for the enrichment of the model. The use of standardized ontologies in the RDF graph allows for improved interoperability and easier integration with other datasets in the Linked Data ecosystem, facilitating enhanced collaboration and information sharing.

Figure 6 represents a linked data approach for the construction industry, highlighting the connectivity possibilities between different CAD, Simulation and Render systems, achieved in a semantic web of linked data structure that follows an ontology structure for interoperability (Pauwels et al., 2017).

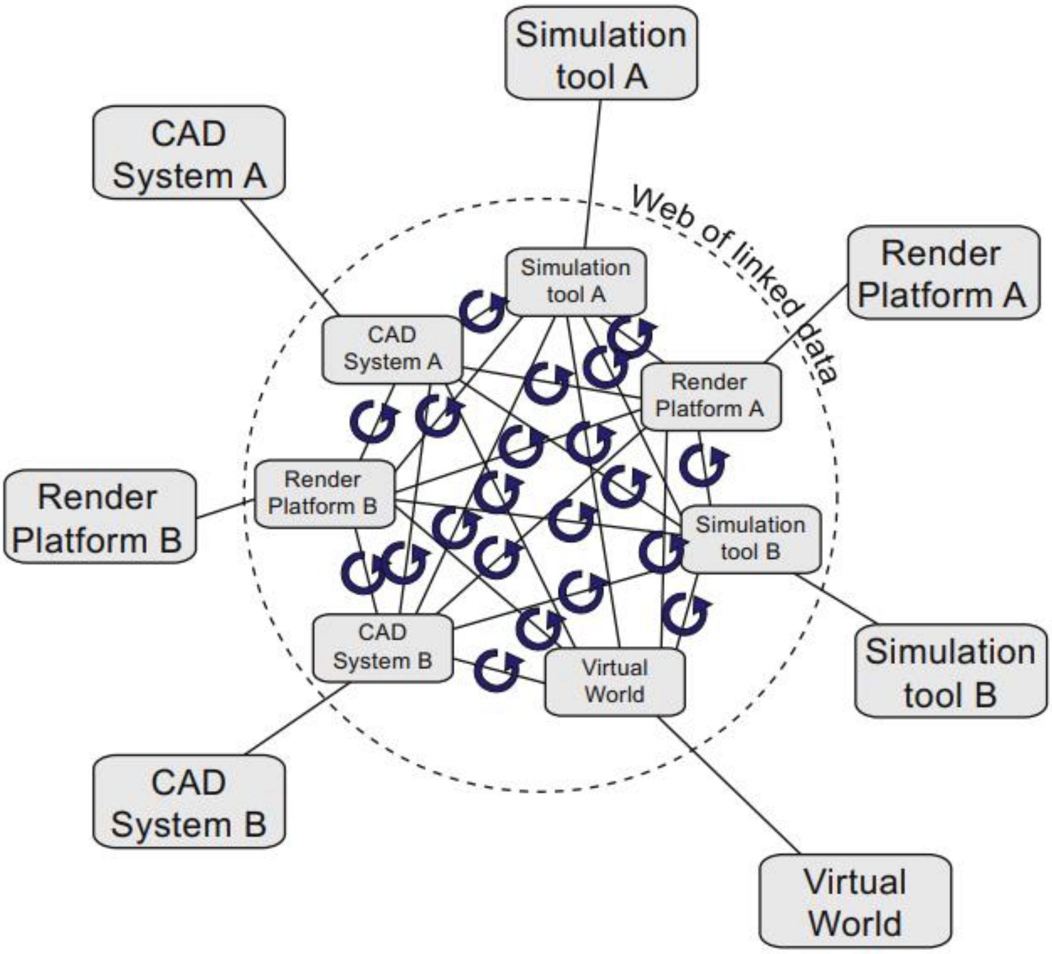


Figure 6: Linked Data Web Representation (Pauwels et al., 2017)

Regarding the benefit raised about the enrichment of the knowledge graph through inclusion of additional information to the nodes, IfcOpenShell is an open-source software library that enables the reading and writing of IFC data, primarily focusing on geometric and semantic information contained within IFC models and can be used for extraction of properties of IFC classes (IfcOpenShell Documentation, 2023). When used in conjunction with the IFCtoRDF tool, IfcOpenShell can enrich RDF graphs by extracting detailed geometric and semantic data from the original IFC model (Dhillon et al., 2014) and associating it with the nodes in the RDF graph that represent built elements (Bonduel et al., 2018). This allows the Linked Building Data (LBD) to incorporate not only topological relationships between building elements but also precise geometric representations such as dimensions, shapes, and spatial relationships (Bonduel et al., 2018). The combination of these tools enhances the RDF graph's richness improving the overall utility of the data for applications in construction management, building analysis, and facility operations.

3.1.3 As-Planned Effort Assessment Through BIM

BIM models can be used to extract quantities of materials, volumes, and other construction-related data from a project. These extracted quantities can then be linked to productivity rates and cost databases to estimate the amount of labor, time, and effort required to build the elements represented in the model (Monteiro & Martins, 2013). Monteiro & Martins (2013) emphasize that accurate quantity take-offs, such as material volumes, areas, and lengths, can be derived directly from 3D BIM models. These quantities can then be linked with productivity rates and unit costs, often stored in external databases, to estimate construction effort in terms of labour, time, and cost.

Many studies have explored the use of BIM for cost estimation in construction projects. For instance, Shen and Issa (2010) conducted a quantitative comparison between BIM-assisted detailed estimating methods and manual estimating approaches across a set of test cases with varying complexity. Their findings showed that BIM-based methods tend to perform better, particularly for entry-level users. On the other hand, Liu et al. (2014) developed a framework that integrates the construction process model with the BIM model to enhance detailed cost estimation. However, they encountered challenges, such as incompatibilities between the BIM-based quantity take-off and subsequent analyses, like cost estimation.

Similarly, Lawrence et al. (2014) proposed an approach that involves mapping BIM objects to cost data, leveraging query languages to connect design parameters directly to cost estimation processes. This technique streamlines the workflow by automating links between design and cost data. Further, Niknam and Karshenas (2015) explored the potential of semantic web technology to integrate cost estimation information provided by different stakeholders, including designers, contractors, and material suppliers. Although ontology-based approaches can reduce human intervention in the estimation process, their implementation requires establishing standardized ontologies and semantic web services within the construction industry. Lastly, Xu et al. (2016) introduced a semantic web ontology-based framework that extracts data from BIM models and uses it to generate the items needed for bills of quantities, which subsequently support cost estimation.

Regardless of the specificities of each of these approaches, all these references seem to agree that the estimation of the necessary effort to execute a building element in a construction project often involve the consideration of the geometry of that element, the material of choice, and the construction methodology adopted for that element (Fazeli et al., 2020).

This information can be extracted from a BIM model, which is typically used to extract detailed geometric and material information for estimating costs and efforts in construction projects. These factors, combined with known productivity rates, allow for estimating man-hours and other resources required for construction (Eastman et al., 2011).

3.2 As-Performed Data Representation

3.2.1 Construction Progress Monitoring

Schlenger et al. (2023) introduce a vision-based approach for automated progress monitoring in construction, specifically targeting cast-in-place shell constructions. In the methodology, fixed on-site cameras are used to capture images, which are then processed by a Convolutional Neural Network (CNN) to classify building elements according to their current status.

The data-capturing system for this approach was described in detail by Collins et al. (2022) and it is summarized here: the system uses crane-mounted cameras for automatic image acquisition. These cameras, placed at strategic positions on cranes, capture images every 30 seconds. The setup includes key components such as a router, Virtual Private Network (VPN), Power over Ethernet (PoE)-Switch, local server, and a remote server for data storage and processing. The system is designed to scale and automate the documentation of the construction process by regularly capturing images from multiple vantage points across the site.

This approach enables the identification of individual operational steps, providing real-time status updates of building elements. The CNN classified the concrete columns identified in the images in four construction phases: not started, rebar, formwork and finished, as illustrated in Figure 7.

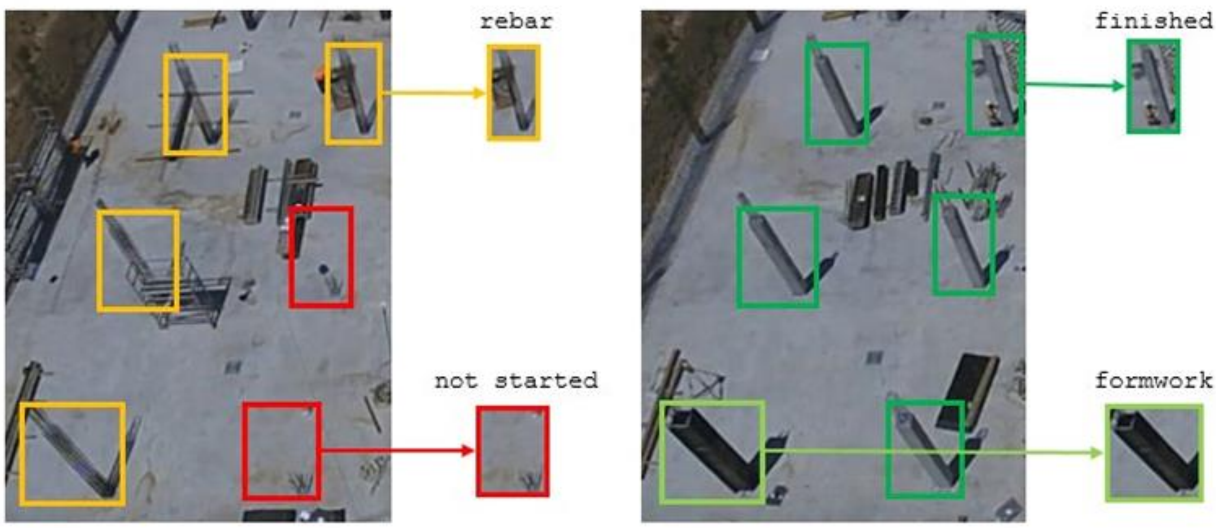


Figure 7: Four construction phases displayed for concrete columns (Schlenger et al., 2023)

The study presented by Schlenger et al. (2023) concludes that the methodology offers a significant improvement over traditional manual monitoring and existing automated methods. By focusing on real-time status updates rather than detailed geometric reconstructions, the approach provides continuous progress monitoring, essential for effective project management in large construction sites. The CNN-based classification system accurately tracked the progress of cast-in-place concrete columns and slabs, accounting for moving objects and other outliers to ensure robust and reliable results.

Diverse other data acquisition techniques have been used in previous research to monitor construction environments. Vähä et al. (2013) showcases a variety of studies regarding potential sensor technologies and robotic applications for automating building construction processes in the construction phase, in uses including prefabrication, on-site assembly and quality control, while Li et al. (2016) focus on the application of Real-Time Locating Systems (RTLS) in construction environments, specifically for improving the efficiency of material and asset tracking on-site.

Sacks et al. (2020) emphasize the unique challenges in construction monitoring, especially due to the dynamic nature of project sites and the need for real-time updates that reflect the as-built state of projects and discuss the integration of multiple data acquisition techniques, such as photogrammetry, laser scanning, and computer vision, to monitor construction progress.

Many researchers, as highlighted by Rehman et al. (2022), utilize photogrammetric reconstruction techniques to monitor construction progress. By extracting geometric features from images taken at various angles, a 3D representation of the site can be generated. Braun et al. (2015) take this further by matching individual points in the reconstructed point cloud to the surfaces of expected building elements, allowing for precise confirmation of their existence. In contrast, Golparvar-Fard et al. (2015) employ a different approach where unstructured images taken by site personnel are used to create point clouds, which are then overlaid on the as-designed BIM model.

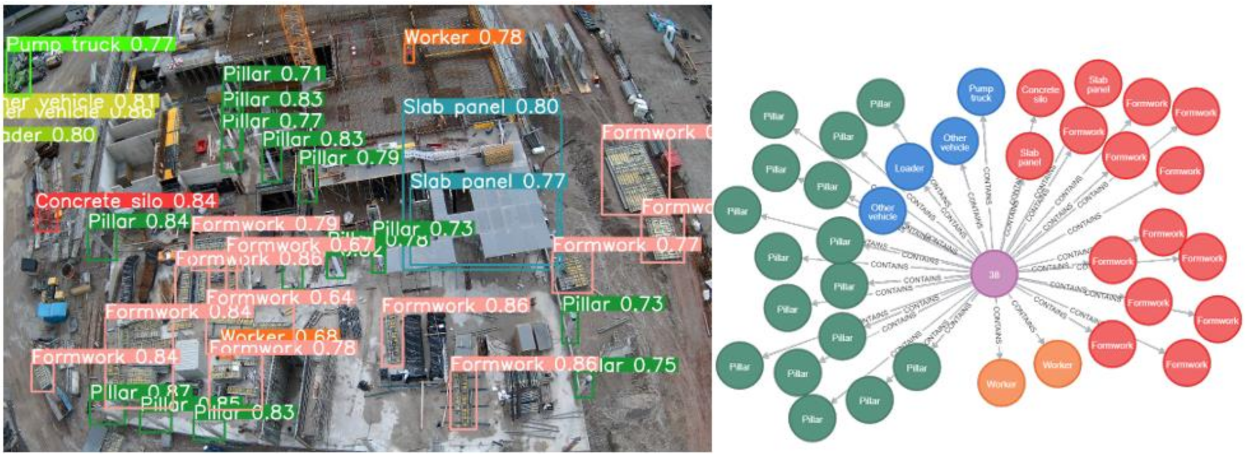
In addition to photogrammetric reconstruction, some researchers focus on analysing individual images or image sequences to monitor construction progress. Fini et al. (2022) use a fixed camera mounted on a tower crane to capture top-down images of prefabricated wooden slab panels. Similarly, Wang et al. (2021) rely on a fixed camera but use neural networks for object detection, instance segmentation, and multi-object tracking to monitor the installation of precast concrete wall elements. Their method aligns the BIM model with reality by identifying wall axes on the horizontal plane. Lastly, Vincke and Vergauwen (2022) use a manual image capture method from various viewpoints to detect the displacement of concrete columns. Unlike the other approaches, they select optimal images to detect deviations in column placement but do not provide a methodology to recognize whether a column exists, assuming its presence.

In general, these studies explore different data acquisition techniques during the construction phase but do not go into detail about integrating as-performed construction data with as-planned data for further comparison and analysis.

3.2.2 Knowledge Graph for the As-Performed Model

Pfitzner et al. (2024) present an approach that integrates object detection with knowledge graphs to enhance productivity in construction. High-frequency images from crane cameras were processed using deep learning for classifying and locating specific on-site objects. This data was then linked into a knowledge graph. This methodology successfully detected construction-related activities, such as working times, identifying process patterns and correlations on construction sites.

Figure 8 showcases an image node, representing all the elements identified on the picture with object detection, such as workers, formwork and pillars, amongst others.



a) Prediction visualized on an image (b) Image node

Figure 8: Processing construction-based image data (Pfitzner et al., 2024)

The study (Pfitzner et al., 2024) concluded that a data-driven digital twin of construction sites could enhance monitoring and management activities significantly. The integration of advanced computer vision techniques and multiple data sources offers a robust framework for real-time construction monitoring. The knowledge graph created helps visualize and analyze complex construction processes.

Another study by Pfitzner et al. (2024) presents an approach for multi-level productivity analysis in construction. It leverages vision-based technologies and neural networks to extract detailed productivity information from images of construction sites. The proposed method is demonstrated through the construction of cast-in-place concrete pillars, highlighting the potential for improved planning and execution of construction projects through data-driven productivity analysis.

For the management and query of the as-performed data, Collins et al. (2022) suggests the implementation of an RDF database, serving as a central repository that links multi-source data from both the design and construction phases. By using established ontologies, such as the Building Topology Ontology (BOT) for building structures and Dublin Core Terms for metadata describing images, the database connects the building elements from the BIM model with the images captured on-site. The RDF graph allows for easy querying using SPARQL, enabling users to extract information, such as the most recent image linked to a particular element (Collins et al., 2022).

This flexible and scalable RDF graph structure, applied in the context of as-performed construction monitoring data collection, provides a foundation for further analysis, such as progress tracking and safety verification, and can be enriched with additional data as needed (Collins et al., 2022).

Knowledge graphs have found many applications in the construction domain. However, only a few papers concentrate specifically on the use of knowledge graphs for storage of construction progress monitoring data (Pan et al., 2021). Rasmussen et al. (2019) demonstrated how the use of semantic web technologies can improve decision-making when working on a construction project with a variety of interconnections by providing the Building Ontology (BOT). In resource management, Pan et al. (2021) designed a computer vision-based video extraction framework to support construction management by detecting processes, tools, and materials.

3.3 Identified Research Gap

The related works highlight the advancements in research regarding the management and manipulation of as-planned data in construction, with a specific focus in graph databases, as well as the handling of as-performed data in construction, including diverse ways to collect this data. However, none of these works propose a methodology that connects the as-planned and as-performed data for further comparison.

The presented research also highlights many use-cases for graph databases in a construction context, but they also don’t explore how this technology can be leveraged to connect and efficiently query both as-planned and as-performed data for optimized decision-making in construction project management.

With this in mind, the research gap that this work aims to address is the development of an end-to-end approach for assessing productivity in construction projects using digital technology. The focus is on creating a graph database-supported data pipeline that correlates as-planned and as-performed information to enhance decision-making and facilitate course corrections in construction project management. This is considered an end-to-end approach because it integrates data collection, query and analysis from both the as-planned phase of the project, as well as the as-performed phase, and use the insights from their analysis to improve decision-making and course correction in the construction phase.

Chapter 4

Methodology

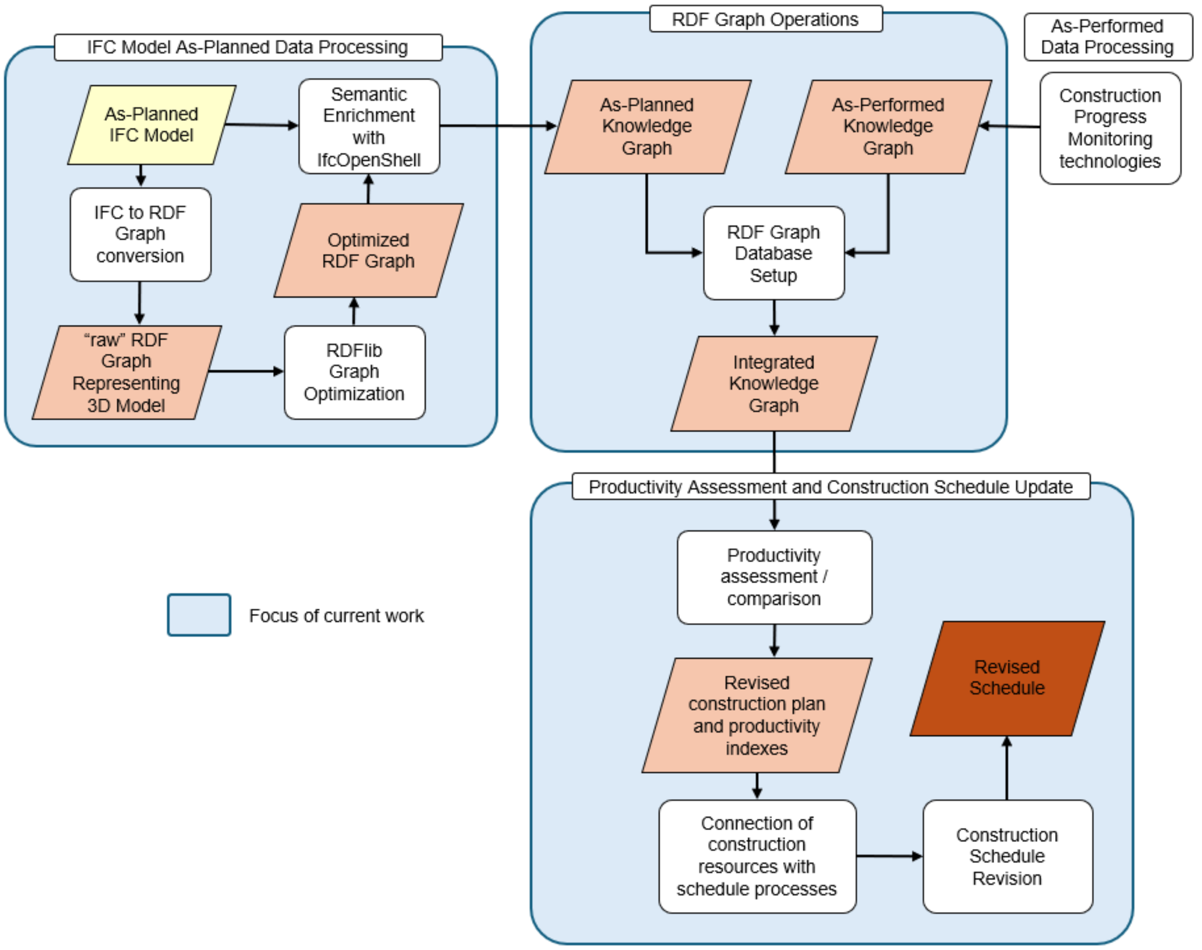


Figure 9: Thesis methodology

The methodology of this research work is structured as presented in Figure 9 and it is initiated with two different pipelines. The first pipeline corresponds to the as-planned data processing, while the second deals with as-performed data processing.

In the as-planned pipeline, the 3D IFC model of the project being utilized is converted into a RDF Graph, which represents the project module in triple subject-relationship-object type elements This graph undergoes several transformations, detailed in this chapter, to form an as-planned knowledge graph, later utilized for productivity assessment and comparison with the as-performed knowledge graph. This step is developed in this work.

The as-performed pipeline represents the process of acquiring information about the construction progress on the site. The method proposed in this chapter builds up on previous work, in which the as-performed data acquisition was further explored (Schlenger et al.,2023 and Pfitzner et al., 2024), as described in Chapter 3. A detailed analysis of as-performed data is beyond the scope of this thesis, and different methodologies for as-performed data acquisition were discussed in Chapter 2. In the scope of this thesis, the methodology focuses on this stage the integration between the as-planned and as-performed data.

These two pipelines then merge into one knowledge graph that contains both the as-planned and the as-performed information, subsequently utilized for productivity assessment and comparison between the planned and performed productivity rates for the built elements. After this comparison, the construction schedule is revised based on the graph data and insights extracted from both as-planned and as-performed information. These steps are further detailed in the following sections.

4.1 IFC Model As-Planned Data Processing

In the as-planned pipeline, the as-planned 3D model data from an IFC format file is processed, simplifying it for query efficiency. The IFC file is converted to a Terse RDF Triple Language (TTL) format to be interpreted as an RDF graph in Neo4j. This RDF graph is optimized, retaining only relevant triples for progress assessment. Nodes representing building elements are enriched with geometric and semantic information using IfcOpenShell, as well as productivity information calculated based on these attributes. At the end of this pipeline, an as-planned knowledge graph is produced, carrying as nodes the building elements from the project, with the necessary geometric, semantic and productivity related information necessary for the subsequent steps. The steps of the as-planned pipeline are broken down in detail in the following sub-sections.

4.1.1 Model preprocessing

The project 3D model is usually presented in an IFC format file. These IFC files usually have geometry and semantic information about diverse elements of the building, however, part of this information is not necessary for the workflow proposed in this work.

Simple BIM or other similar tool is utilized to remove the IFC classes present in the model that wouldn't be used in order to make the information querying process more efficient, once the optimized model is converted to graph format. For example, if the goal is to assess the productivity of wall construction, specifically frame erection, information related to the wall's geometry, materials, and construction method is relevant and should be retained, as discussed in Section 2.7. In this example, information about other domains could be removed, because they're not needed, such as data about electrical wiring, furniture and light fixtures, or the overall finishing elements of the construction or even different building elements that may not be target of this specific progress monitoring.

4.1.2 IFC to RDF Graph conversion

In this step, the optimized IFC file is converted into TTL format in order to be readable as an RDF graph in tools such as Neo4j. Neo4j's capabilities allow for detailed, real-time analysis of construction data, facilitating schedule insights, delay identification, and optimization suggestions (Pokorný, 2015).

The IFCtoRDF framework can be used for this. This conversion enhances data interoperability and integration across different systems by translating IFC entities and relationships into RDF and OWL formats (Pauwels & Roxin, 2016).

It is recommended to group the most common triple relationship types by quantity and assess whether the information they carry is relevant to the use case. If not, removing frequent relationship types with the aid of a script will help reduce the size and complexity of the RDF graph, allowing queries to run more efficiently.

The decision to convert the 3D IFC model into an RDF Graph for the as-planned pipeline was based on the flexibility and scalability offered by linked data technology. RDF graphs allow data to be stored as subject-predicate-object triples, making it easier to represent complex relationships inherent in construction data. This structure supports the integration of diverse data types and is particularly effective for querying and analysing relationships between building elements (Pokorný, 2015).

The choice of RDF graphs also facilitates the combination of multiple datasets, which is critical for this research as it aims to integrate both as-planned and as-performed data into a unified knowledge graph. Neo4j was selected as the graph database for this purpose due to its robust capabilities in handling complex queries and enabling real-time data access.

4.1.3 RDF Graph optimization

If necessary, the RDF graph generated in the previous step can be further optimized through scripting. For this step RDFLib is recommended, a Python library that provides tools to parse, serialize, and query RDF data. With RDFLib, users can create and manipulate RDF graphs, leverage SPARQL queries for data retrieval, and integrate various RDF formats like Turtle, RDF/XML, and N-Triples (RDFLib, 2023).

A custom script parses the TTL file to retain only relevant triples for specific building element types such as IfcWall, IfcSlab, and IfcColumn. This step significantly reduces the number of triples, maintaining only those necessary for productivity analysis. For example, in the case of IfcColumn, the script only maintains tripes that either had IfcColumn as a subject or as an object and would further categorize objects and subjects related to this class. This step also reduces the complexity of the RDF Graph, allowing for more efficient querying.

This premise is based on the understanding that an RDF graph represents the relationship between entities, if there is no connection between a specific triple and the IFC element that we are interested in, it is likely that this triple does not provide relevant information to further categorize this IFC element, and it is not necessary for this specific use case.

In the example of the IfcColumn, triples such as IfcPositiveInteger_List(subject) → IfchasContents(predicate) → IfcPositiveInteger(object), or IfcLengthMeasure_List(subject) → IfchasContents(predicate) → IfcLenghtMeasure(object) occur at least once per instance of every single built element, and don’t necessarily provide any geometric of semantic information necessary for the productivity calculation, which means they can be removed from the graph for this specific use case.

4.1.4 IfcBuiltElement properties extraction

The conversion to TTL process does not transfer the geometric and semantic properties from these classes to the nodes, therefore, this information needs to be added in an additional step, considering that data related to the geometry and material of the instances are necessary for the calculation of the necessary efforts to build them.

These additional geometric and semantic properties should be extracted from the IFC file using scripting. For this step IfcOpenShell is recommended, an open-source library for parsing and processing IFC files (IfcOpenShell, 2023). These properties are added to the RDF graph nodes to support detailed productivity calculations. A script reads the necessary geometric and semantic data from the IFC file, updates the corresponding nodes in Neo4j, including this information as properties for the nodes.

4.1.5 Planned productivity

Productivity rates are calculated based on the specific characteristics of each building element type, such as geometry and construction requirements. These rates are added as properties to the RDF graph nodes by the utilization of geometric and semantic properties added to the nodes in step 4.1.4 as input, enabling detailed productivity assessments and further analysis.

The general productivity rate formula is expressed by the quotient between the total work output and total time to execute the activity, but the specificities about the total work output naturally derive from the nature of the built element being constructed, generally including factors such as geometry, density and construction requirements, amongst others, these specificities were discussed in section 2.7.

In this step, productivity rates are calculated specifically to each IFC Built Element target of the use case that is being performed. These productivity rates are usually assessed in terms of a geometric dimensions (e.g. m2 or m3) divided by the man-hours necessary to perform this geometric unit.

With these productivity rates, the necessary time in man-hours for construction of each element can be calculated considering the existing geometric and semantic properties of each element and become a new property added to each node in the building element knowledge graph, representing each instance of this element.

This new property is then added according to which type of built element each instance corresponds to (in case of columns, if there are made of concrete or steel, for example). The calculation is done for each instance and the result is inserted as a property "Effort (man-hours)" in each node with a CYPHER query. The steps described in section 4.1.5 are graphically represent in Figure 10 below.

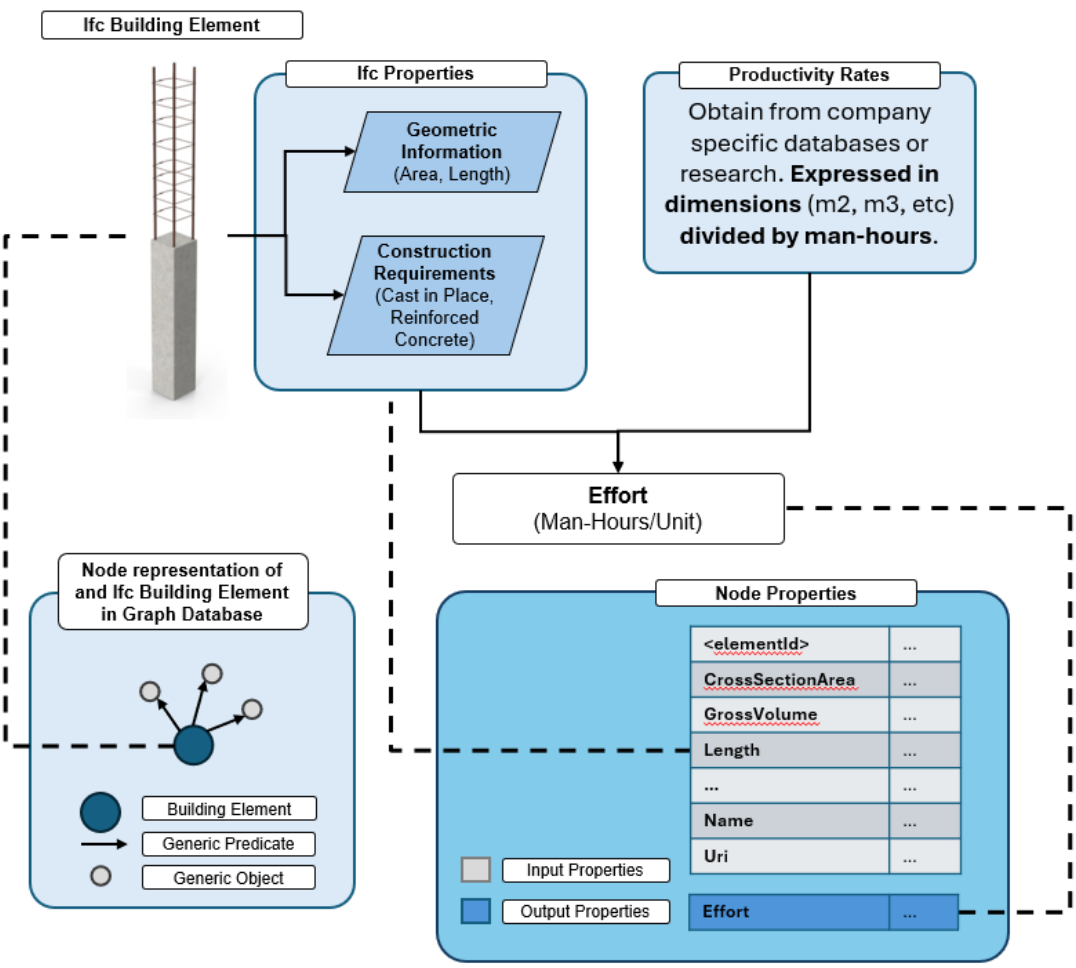


Figure 10: Planned Productivity Calculation with BIM and RDF Graphs

The selection of which properties from the specific building element are relevant for the calculation will depend on the nature of the element and how their productivity rate is usually measured. For example, for reinforced concrete columns, references present productivity rates are calculated for the rebar (CBIC, 2017), formwork (Moselhi et al., 2020) and concrete pouring (RSMeans, 2020) production steps and depend on gross volume, for the rebar and concrete pouring steps, and surface area of the element, for the formwork step, as presented in Table 2.

Table 2: Productivity rates for IfcColumn

Type	Reinforced Concrete
Rebar	18,2 kg (of rebar) / man-hour or 0,12 m3 (reinforced concrete) / man-hour (CBIC, 2017).
Formwork	1,1 m2 / man-hour (Moselhi et al., 2010)
Concrete Pouring	1,0 m3 / man-hour (RSMeans, 2020)

However, a different building element require different geometric and semantic information for its calculation. The information about preferred or recommended construction methods can be obtained from the company's own construction history or as an academic reference. The construction method will dictate the necessary properties that need to be considered from the nodes representing the instances of a specific construction element.

4.2 As-Performed Data Integration

The as-performed pipeline represents the process of acquiring information about the construction progress on site. Methods of data acquisition were discussed on section 2.1.2, and they are compatible with the proposed methodology, as long as acquired data can be converted to an as-performed knowledge graph following the same ontology as the as-planned knowledge graph, generated with the as-planned pipeline presented in section 4.1.

The as-planned knowledge graph generated from the steps presented in section 4.1 have triples that correlate each instance of the built elements part of the scope with a node called IfcGloballyUniqueId, which contains a property named hasString, with a string representing a unique ID to identify each instance. The process to generate the as-performed knowledge graph should also include these triples and this unique identifier, because this is the property that will be matched between the two data sources to merge information between them. Figure 11 illustrates this process.

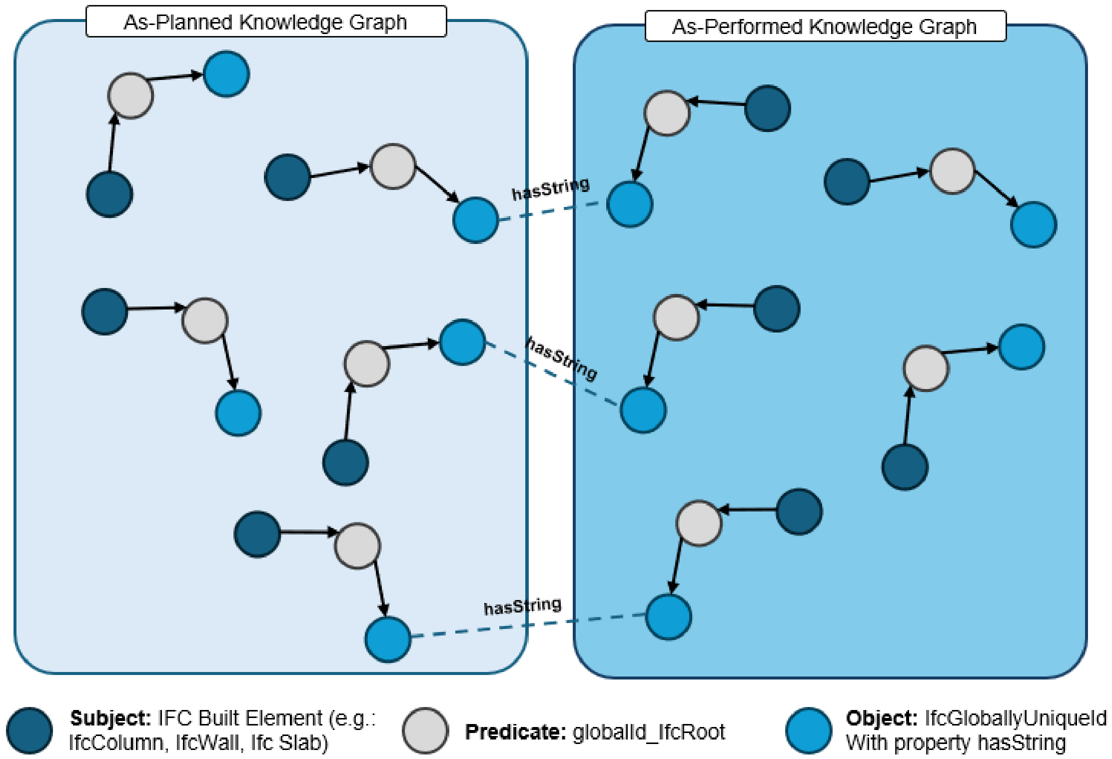


Figure 11: As-planned and as-performed knowledge graph integration

When every instance is matched through the property hasString of the node IfcGloballyUniqueId, the information between the as-planned and as-performed version of the same instance can be compared, specially the “Effort (man-hours)”. The comparison between the as-planned and as-performed results will show if the element was constructed faster, slower, or exactly as planned.

4.3 RDF Graph Operations

In this phase, the calculated productivity rates for each building element type included in their respective nodes and resulted from the steps described in sections 4.1, for as-planned data, and 4.2, for as-performed data, are finally compared. Both as-planned and as-performed information is combined in the same knowledge graph for further comparison to evaluate construction progress and identify potential deviations from the planned estimates. The data collected for each of these elements can be used for further analysis and identification of potential factors influencing on the productivity.

4.3.1 Productivity assessment

This step involves comparing the as-planned effort in man-hours for each building element with its as-performed equivalent. Both these properties can be queried on Neo4j using CYPHER notation.

The comparison of the planned and performed efforts is then enriched with the analysis of other characteristics of these building elements, also captured through the extraction of information from the IFC model or the construction site, such as geometry, material, location, month of the year and weather conditions in which the element was built, proximity to other elements, team involved in the construction and other factors that may have influence in the performance.

As discussed in Chapter 2, with enough information about productivity and characteristics of the building elements, this data can be utilized in data analytics and machine learning models to identify insights into construction delays and potentially improve overall project performance.

This analysis can also provide insight into the potential need to revise the productivity rates originally adopted for the planning phase. If the analysis pinpoints a factor that impacts on this index and wasn't considered in the original estimation, a new rate can be proposed for the remaining activities of the project that considers potential factors identified with this analysis.

The as-planned and as-performed properties is then combined in one single node per element, which allowed for the comparison between planned and performed necessary efforts in terms of man-hours. A new attribute named "Variation" is calculated to assess the quantitative different between the as-planned and as-performed efforts. This attribute is calculated as follows:

$$\text{Variation} = \frac{(\text{AsPerformedEffort}) - (\text{AsPlannedEffort})}{\text{AsPlannedEffort}}$$

A variation of 0% signifies that the as-performed efforts are exactly as predicted in the as-planned estimation. A positive variation represents an underestimation of the necessary efforts, and a negative variation represents an overestimation of the necessary efforts during the planning phase.

4.3.2 Construction plan revision

The construction short-term and long-term plan is then revised considering recalibrated index and efforts resulted from the previous step. The plan should be revised also taking into account other insights there weren't originally considered in the planning and were identified during the comparison promoted by the methodology, insights such as unsuitable material performance in adverse weather, ineffective team configurations, or logistical challenges.

If the methodology has been used in previous similar projects in the same company before, the previous data can also be used to support decision making and course-correction.

4.3.3 Construction schedule update

With aspects of the construction plan revised, the construction schedule needs to reflect these changes. The impacts assessed in the previous step are taken into consideration and also presented in the updated version of the schedule, to be compared with the baseline schedule. For example, if with the previous step it was assessed that the walls that were constructed so far took on average 25% longer to be concluded than what the baseline schedule predicted because of a factor that wasn't originally considered, the updated schedule should update the duration of the remaining walls to consider this factor. This way, the current planning can reflect the newly discovered insights, when applicable. Figure 12 illustrates this process.

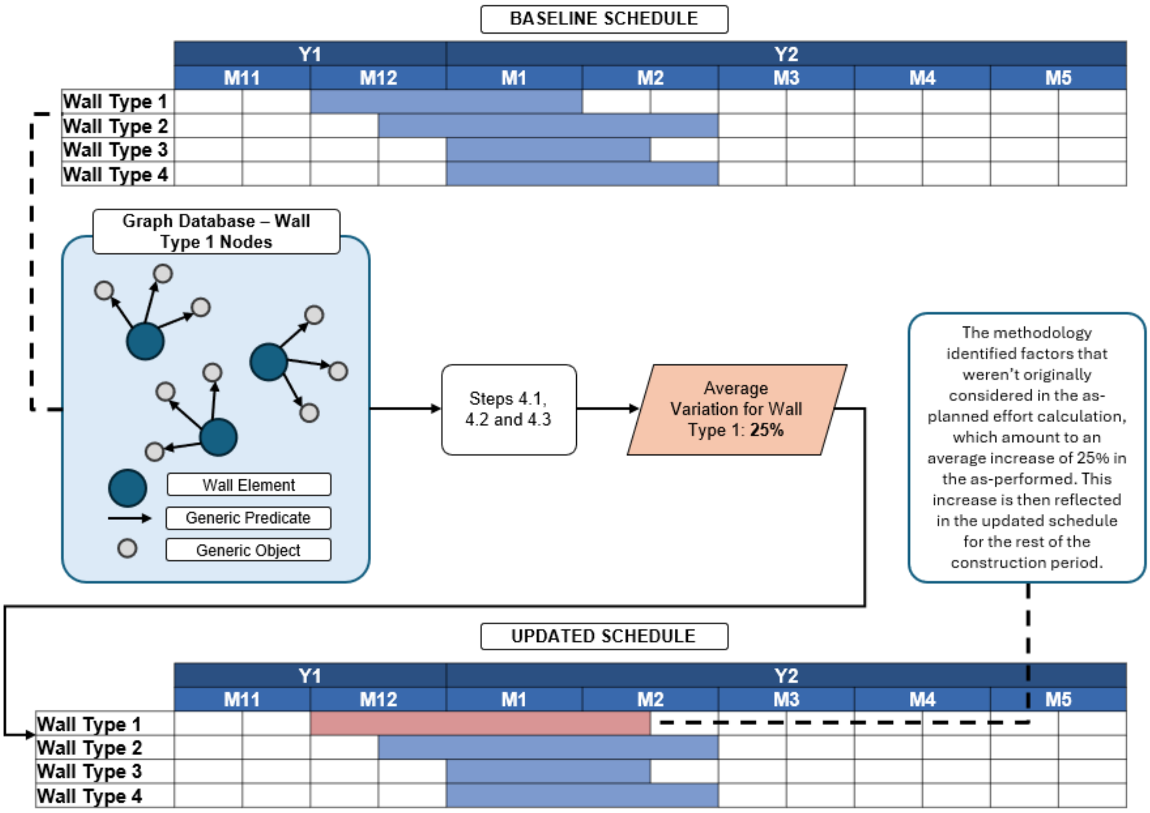


Figure 12: Schedule Update

However, precise schedule optimization techniques and automated connection with the as-planned and as-performed databases are out of the scope of this thesis. The goal of this section is to simply point out the possibility of this complementary step, which could be further explored in future work.

Chapter 5

Case Study

The case study was conducted utilizing as-performed information from a construction project in Munich, Germany. The construction site corresponds to a scientific laboratory building. The construction period was from April 2022 to August 2022. The building contains four storeys and a ground floor area of 3200 square meters. Figure 13 illustrates the site during the construction process.

Figure 13: On-site environment of case study building

The steps present in the as-planned pipeline, as described in section 5.1, were performed utilizing the 3D IFC model of this project and are described in the following subsections. The steps corresponding to the as-performed pipeline consider data acquired from earlier work done by Pfitzner et al. (2024).

Finally, an RDF graph database was set up by integrating the as-planned and as-performed data, originating from the two pipelines. This graph was utilized for productivity assessment and comparison between the planned and performed productivity rates for the built elements. Figure 14 presents the BIM model of the project, which is the object of the case study.

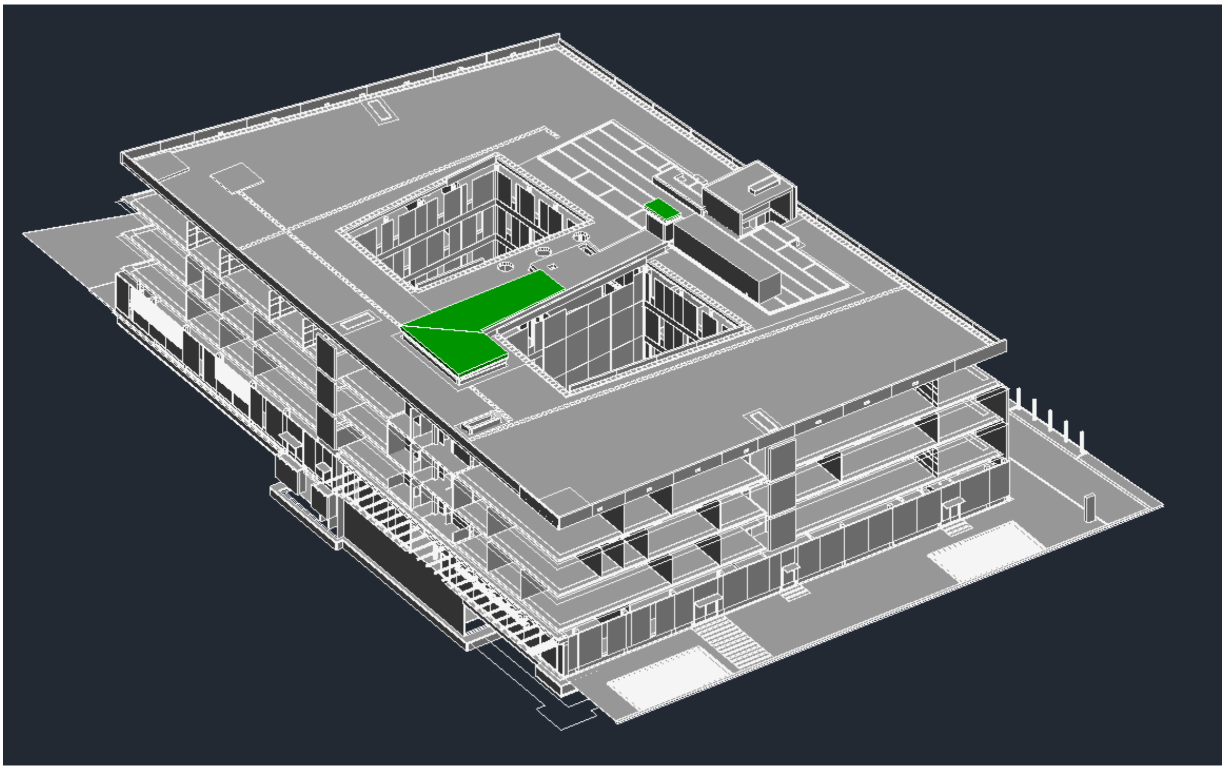


Figure 14: Case study building BIM model on Revit Viewer

5.1 IFC Model As-Planned Data Preprocessing

5.1.1 Model preprocessing with SimpleBIM

The project model was provided in an IFC format file containing geometric and semantic information about various building elements. However, some of this information was unnecessary for the workflow proposed in this research. Therefore, SimpleBIM was used to remove irrelevant IFC classes, optimizing the model for more efficient querying once it was converted to graph format.

Table 3 illustrates the object classes that were preserved on the simplified model and the object classes that were removed for not being relevant for this use case, as well as the number of instances of each class. This process reduced the file size to half of its original size. The filtering criteria here involved, as discussed in Chapter 4, removing disciplines that were unrelated to the frame erection of the main built elements (Column, Wall, Plate, Roof, Slab, Stair), and preserving objects that may provide information about the geometry or location of these built elements.

Table 3: SimpleBIM IFC model simplification

Preserved Object Classes	Removed Object Classes
Beam (151), Building (1), Building Element Proxy (807), Building Storey (7), Column (304), Covering (882), Curtain Wall (347), Door (317), Element Assembly (4), Model Information (1), Plate (3434), Project (1), Roof (18), Site (1), Slab (499), Stair (52), Wall (1476), Window (175)	Distribution Port (33), Flow Terminal (275), Footing (40), Furniture (28), Member (14), Railing (38), Space (491), Stair Flight (15), System Furniture Element (9), Transport Element (14)

5.1.2 IFC to RDF Graph conversion with IFCtoRDF

IFCtoRDF was used to convert models in the IFC format into Linked Building Data using Semantic Web technologies. This tool was utilized on the optimized IFC file, output of the last step, to convert it into an TTL format readable by graph database tools such as Neo4j. The conversion generated a TLL file with 5.1 million triples, even though the input was the IFC file optimized on the last step.

This large number of triples represented various relationships between entities in the 3D model. There were 497 different types of relationships. However, the nine types shown in Table 4 (approximately 2% of the total) accounted for around 4.5 million triples, or 90% of the entire TTL file, none of which provided relevant information for the use case explored in this study.

Table 4: Most frequent triple types identified in the TTL representation of the 3D Model

Subject_Type	Predicate	Object_Type	Count
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcPositiveInteger_List	https://w3id.org/list#hasContents	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcPositiveInteger	1110879
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcPositiveInteger_List	https://w3id.org/list#hasNext	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcPositiveInteger_List	876331
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcLengthMeasure_List	https://w3id.org/list#hasContents	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcLengthMeasure	793679
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcLengthMeasure_List	https://w3id.org/list#hasNext	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcLengthMeasure_List	505259
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcLengthMeasure_List_List	https://w3id.org/list#hasContents	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcLengthMeasure_List	294383
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcLengthMeasure_List_List	https://w3id.org/list#hasNext	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcLengthMeasure_List_List	279339
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcIndexedPolygonalFace_List	https://w3id.org/list#hasContents	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcIndexedPolygonalFace	228257
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcIndexedPolygonalFace	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#coordIndex_IfcIndexedPolygonalFace	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcPositiveInteger_List	228257
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcIndexedPolygonalFace_List	https://w3id.org/list#hasNext	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcIndexedPolygonalFace_List	226953

The triples presented in Table 3 don't carry any geometric and semantic information for the building elements, necessary to calculate the efforts for their completion. For example, the relationship IfcPositiveInteger_List (subject) → hasContents (predicate) → IfcPositiveInteger (object) represents a connection where a list of positive integers (subject) contains individual positive integer values (object). Essentially, this relationship allows for the specification of multiple positive integer values within a structured list, ensuring that all values within the list are positive integers, as indicated by the

IfcPositiveInteger datatype. The triples with other lists as subjects (e.g. IfcLength-Measure_List, IfcLenghtMeasure_List_List, IfcIndexedPolygonalFace_List) follow a similar logic (buildingSMART, IFC4 ADD2, 2016).

Considering this fact, another optimization was done to only maintain triples who carry relevant information for the productivity related to the building elements who are being assessed in this study, and only the 10% remaining relevant triples were kept on the resulting file.

5.1.3 RDF Graph optimization with RDFLib

For this phase, three of the built elements of this project were selected, namely IfcWall, IfcSlab and IfcColumn, which, for the purpose of monitoring the frame erection of the building, they are the most relevant elements. The remaining steps of the pipeline were applied to each of these built element types.

In order to simplify and optimize the TTL file, maintaining only the necessary triples to perform this use case, a Python script using the RDFLib library was developed to parse through the whole original file. The parsing process preserved only triples that, in the case of IfcColumn, for example, either had IfcColumn as a subject or as an object, and would further categorize objects and subjects related to this class. The same process and criteria were applied to IfcWall and IfcSlab.

In a second step, the Python script enhanced the new outputted RDF graph representing one of the classes by integrating relevant triples from related to IfcGloballyUniqueId from the original RDF graph. The presence of nodes specifying the IfcGloballyUniqueId of each instance of the class is important for the next step of the process. This script's detailed functionality and the complete code are included in Appendix A for further reference and reproducibility.

After running the script for IfcWall, IfcSlab and IfcColumn, a new simplified TTL file were generated for each of them, with a much smaller number of triples, as presented in Table 5, which summarizes the triple count and the number of instances for each of those classes. The complete list of triple relationship types for these three classes is presented in table form in Appendix B. The total number of remaining triples after this step is of approximately 60 thousand, which is drastically smaller than the original 5.1 million original triple count.

Table 5: Triple and instance count for IfcWall, IfcSlab and IfcColumn

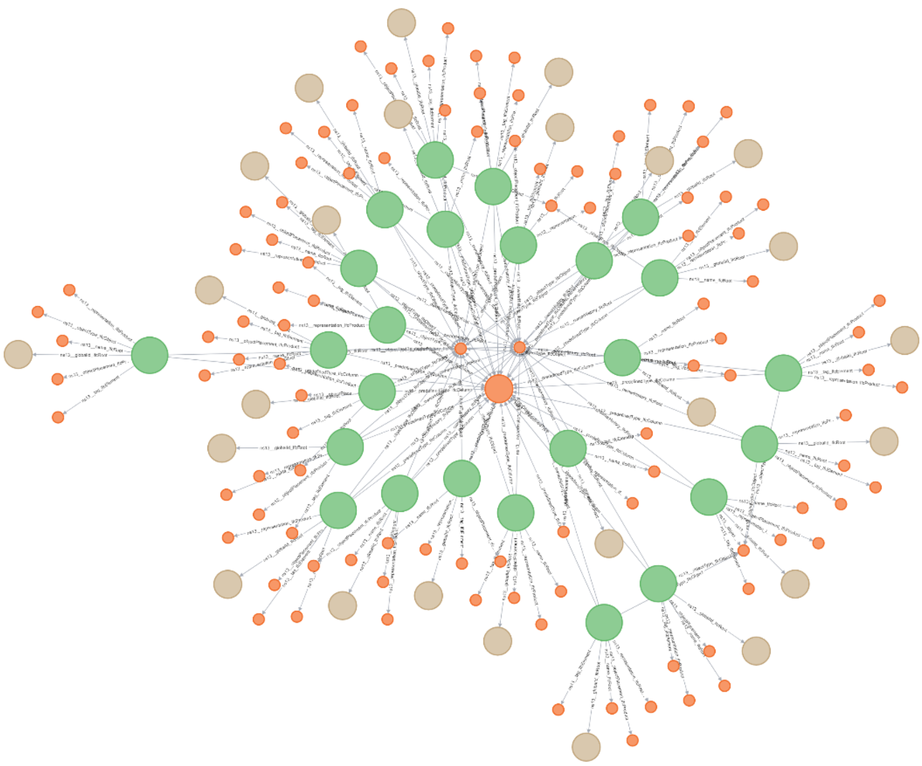
Building Element	Triple Count	Number of Instances

IfcWall	34.908	1476
IfcSlab	14.950	499
IfcColumn	9.220	304

Figure 15 presents a sample of 200 nodes and relationships from the IfcColumn TTL file, displayed on Neo4j, highlighting many of the relationships in which IfcColumn is either subject or object, including their connection with IfcGloballyUniqueID. This unique identifier will be useful in the next step for connecting further information extracted from the original IFC file to the final class-specific graph.

```
MATCH (n)
WHERE any(prop in keys(n) WHERE toString(n[prop]) CONTAINS 'IfcColumn')
OPTIONAL MATCH (n)-[r]->(m)
OPTIONAL MATCH (n)-[rel:ns13__globalId_IfcRoot]->(o)
RETURN n, r, m, rel, o
LIMIT 200;
```

(a) Neo4j query for IfcColumn



(b) Node Graph for IfcColumn

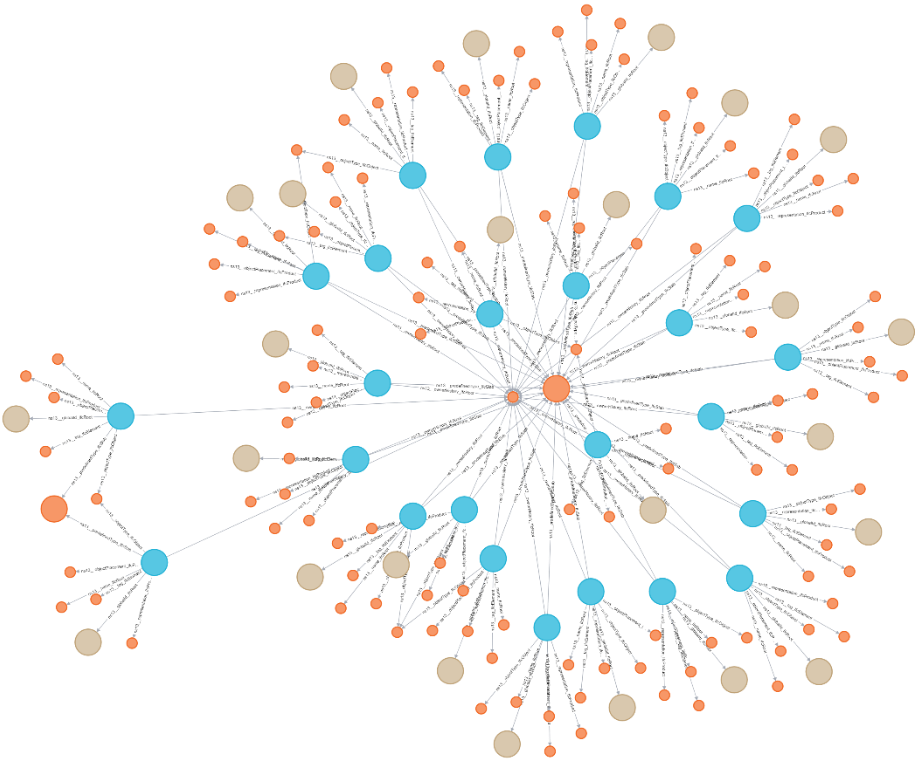
(c) Node Labels for IfcColumn

Figure 15: Neo4j query representing a sample of the IfcColumn nodes and their relationships

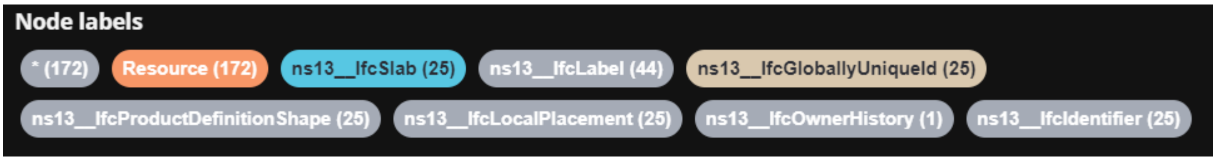
The graph was also generated for the building elements IfcSlab and IfcWall and are represented in Figures 16 and 17. In general, a lot of the triples and nodes presented in each of those two elements types are similar to the logic of IfcColumn, however, each of them possess particularities related to their nature, for example IfcSlab triples and nodes need to represent if they are floor, landing or roof types, and IfcWall triples and nodes need to demonstrate the relationship between a Wall element and their windows and doors, which impacts in their net volume.

```
MATCH (n)
WHERE any(prop in keys(n) WHERE toString(n[prop]) CONTAINS 'IfcSlab')
OPTIONAL MATCH (n)-[r]→(m)
OPTIONAL MATCH (n)-[rel:ns13__globalId_IfcRoot]→(o)
RETURN n, r, m, rel, o
LIMIT 200;
```

(a) Neo4j query for IfcSlab



(b) Node Graph for IfcSlab

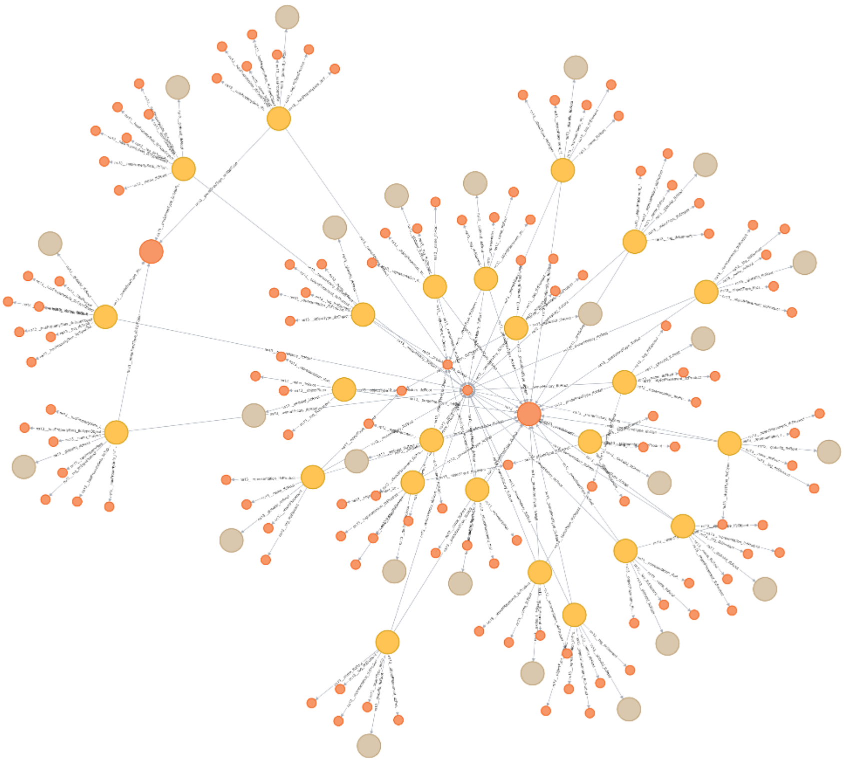


(c) Node Labels for IfcSlab

Figure 16: Neo4j query representing a sample of the IfcSlab nodes and their relationships

```
MATCH (n)
WHERE any(prop in keys(n) WHERE toString(n[prop]) CONTAINS 'IfcWall')
OPTIONAL MATCH (n)-[r]→(m)
OPTIONAL MATCH (n)-[rel:ns13__globalId_IfcRoot]→(o)
RETURN n, r, m, rel, o
LIMIT 200;
```

(a) Neo4j query for IfcWall



(b) Node Graph for IfcWall

Node labels

* (171) | Resource (171) | ns13__IfcWall (20) | ns13__IfcLabel (40) | ns13__IfcGloballyUniqueId (24) | ns13__IfcProductDefinitionShape (20)

ns13__IfcOwnerHistory (1) | ns13__IfcIdentifier (20) | ns13__IfcLocalPlacement (20) | ns13__IfcWallType (4) | ns13__IfcPropertySet (20)

(c) Node Labels for IfcWall

Figure 17: Neo4j query representing a sample of the IfcWall nodes and their relationships

5.1.4 IfcBuiltElement properties extraction with IfcOpenShell

The conversion to TTL process does not transfer most of the geometric and semantic properties from these classes to the nodes, therefore, this information needs to be added in an additional step, considering that data related to the geometry and material of the instances are necessary for the calculation of the necessary efforts to build them. Figure 18 showcases properties from a IfcColumn example, visualized with the aid of BlenderBIM. These properties include geometric information such as CrossSection-Area, GrossVolume, Length, NetVolume and OuterSurfaceArea, as well as other types of information such as IfcLocalPlacement and ObjectType. Finally, an attribute called GlobalId also needs to be extracted, so it is possible to match each instance from the IFC file to the RDF Graph nodes with their IfcGloballyUniqueID.

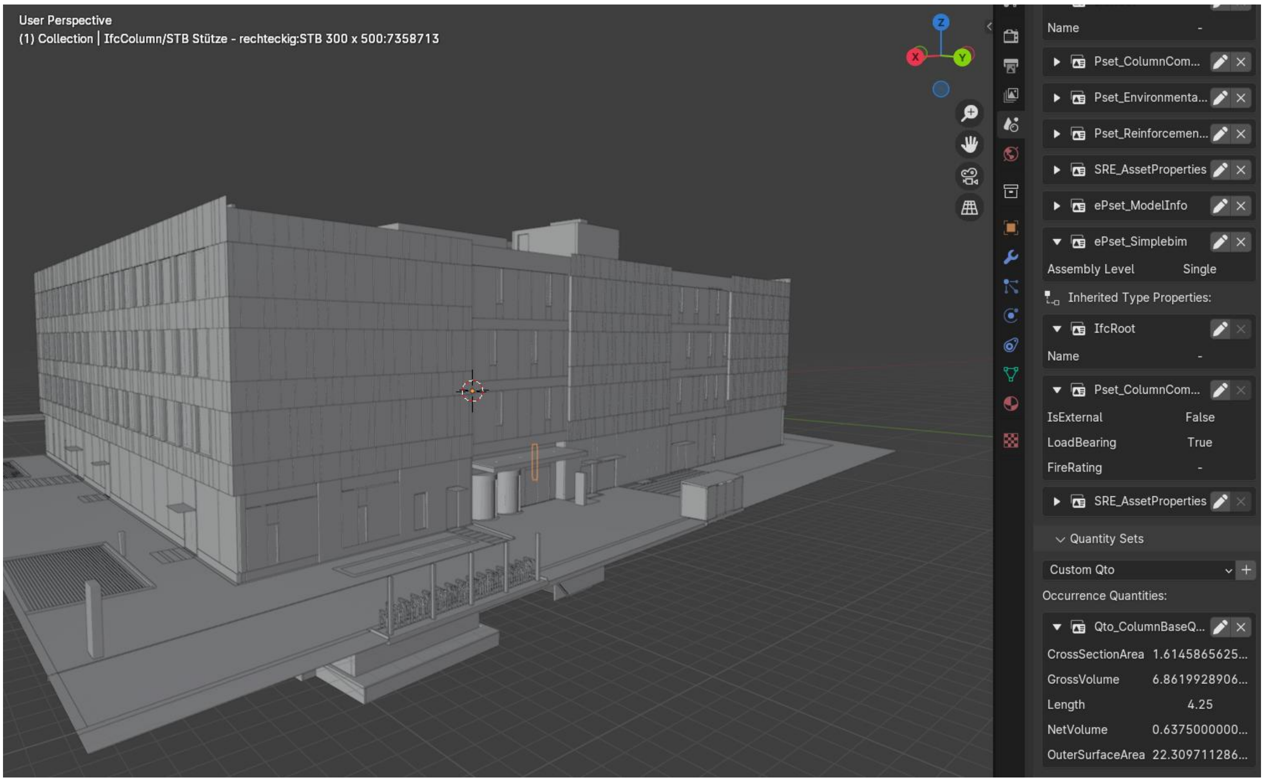


Figure 18: IfcColumn properties on IFC file

Once the necessary properties were identified, a Python script was developed using the IfcOpenShell library to extract specific quantities and properties from built elements of a given type (in this case, IfcColumn) from the IFC file.

The script defined a function that retrieves properties such as IfcGloballyUniqueId, Name, GrossVolume, NetVolume, OuterSurfaceArea, CrossSectionArea, Length and IfcLocalPlacement from the elements. Another function parsed through the specified IFC file, iterated through the elements of the specified type, collected the extracted data into a list and converted this list into a Pandas data frame. The script is presented in Appendix C.

This Python script then reads the Pandas data frame about the IFC instances and updates corresponding nodes in a Neo4j graph database. The core functionality is handled by a function which constructs and executes a Cypher query to match IFC columns based on their globally unique ID. If a match is found in the Neo4j database, the corresponding node is updated.

Figure 19 shows an example of a IfcColumn node in the updated version of the RDF Graph for this built element class. Its node properties tab now lists the additional properties that were included in this step, necessary for further calculations.

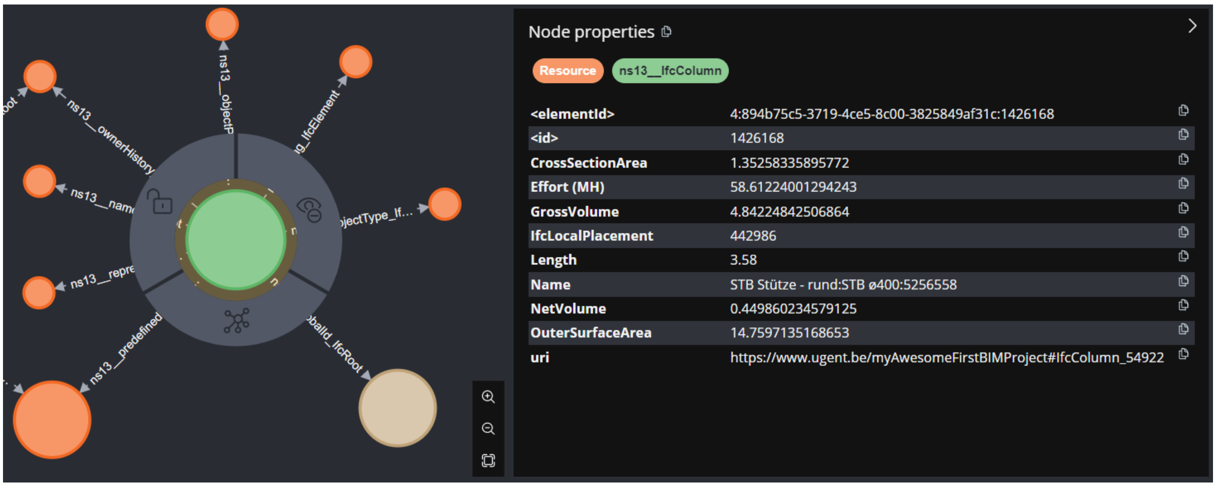


Figure 19: IfcColumn node with additional properties in Neo4j

5.2 As-Performed Data Processing

The as-performed data from the Munich construction site was collected, then processed using a deep learning approach to classify and localize specific on-site objects, then complemented with additional data sources like weather information and presented in knowledge graph format by Pfitzner et al. (2023) as described in Chapter 3. Information about IfcColumn elements specifically were collected and stored in the aforementioned knowledge graph, and included data such as a global ID, starting date and time, finishing date and time and total duration of construction in days, current progress status, bounding box coordinates and storey by Pfitzner et al. (2023), for further integration and comparison with the as-planned data.

However, depending on the employment of a different methodology for as-performed data acquisition, or the employment of different parallel methods, more information about the construction progress and construction site can be collected and added to each construction element, to enrich the data oriented decision-making process.

Planned and performed data were integrated into a single knowledge graph by matching each IfcColumn as-planned instance with its as-performed counterpart through the hasString property of their IfcGloballyUniqueId node.

5.3 RDF Graph Operations

5.3.1 Productivity assessment

As-planned productivity calculation

The general productivity rate formula is expressed by the quotient between the total work output and total time to execute the activity, but the specificities about the total work output naturally derive from the nature of the built element being constructed, generally including factors such as geometry, density and construction requirements, amongst others, as discussed in section 2.7.

In this step, productivity rates are calculated specifically for IfcColumn, which is the element with existing as-performed information for comparison, as mentioned in Chapter 3.

The construction steps for the columns are summarized in the Table 6 below for the two main general types of columns, in reinforced concrete and in steel. The table summarizes productivity rates and the families that are part of each of those types for this specific IFC model file utilized in the case study.

Table 6: Productivity rates and related families for IfcColumn in the case study

Type	Reinforced Concrete
Rebar	18,2 kg (of rebar) / man-hour or 0,12 m3 (reinforced concrete) / man-hour (CBIC, 2017).
Formwork	1,1 m2 / man-hour (Moselhi et al., 2010)
Concrete Pouring	1,0 m3 / man-hour (RSMeans, 2020)
Families	STB Stütze - rechteckig: STB (250 x 250, 250 x 700, 300 x 300, 300 x 400, 300 x 500, 350 x 350, 350 x 700, 350 x 900, 400 x 400, 450 x 450, 500 x 800). STB Stütze - rund: STB (ø 270, ø 350, ø 400, ø 450, ø 500).
Type	Steel
Position and Alignment	1,5 columns / man-hour (RSMeans, 2020)
Bolting	0.75 column / man-hour (CBIC, 2017).
Welding	3 m / man-hour (CBIC, 2017).
Families	IPE Stütze:IPE 240, HEM Stütze:HEM 140, HEA Stütze:HEA (160, 200).

With these productivity rates, the necessary time in man-hours for construction of each element can be calculated and become a new property added to each node in the IfcColumn graph, representing each instance. This new property is then added according to which type of column each instance corresponds to (concrete or steel). The calculation is done considering the individual properties of each instance and the result is inserted as a property with the CYPHER queries presented in Figure 20:

```
MATCH (n:ns13__IfcColumn)
WHERE n.Name CONTAINS 'STB Stütze' OR n.Name CONTAINS 'jba_Stütze'
SET n.`Effort (MH)` = n.GrossVolume / 0.12 + n.OuterSurfaceArea / 1.1 + n.GrossVolume / 1.0
RETURN n.Name, n.`Effort (MH)`, n.GrossVolume, n.OuterSurfaceArea;
```

(a) Reinforced Concrete

```
MATCH (n:IfcColumn)
WHERE n.Name CONTAINS 'IPE Stütze' OR n.Name CONTAINS 'HEM Stütze' OR n.Name CONTAINS 'HEA Stütze'
SET n.`Effort (MH)` = 1.5
RETURN n.Name, n.`Effort (MH)`
```

(b) Steel

Figure 20: CYPHER queries for IfcColumn effort definition in Neo4j

Productivity comparison

The information present in both as-planned and as-performed knowledge graphs are matched through the IfcGloballyUniqueId, property present in each of the building elements nodes in both of those graphs. In this case, the match was made for the 20 IfcColumn elements selected from the as-performed graph as a sample.

There was a total of 304 IfcColumn elements in the project, according to the IFC file extraction. However, the as-performed information was extracted considering the construction method for reinforced concrete, which was applied to 251 of the 304 columns. The 20 IfcColumn sample contained the main section shapes for the reinforced concrete columns (circle and square) an the most frequent section areas, being a good representation of the most frequent reinforced concrete column types of the project.

Figure 21 illustrates the final captured images for some of the 20 columns considered for this step of the methodology, demonstrating their final construction stages.

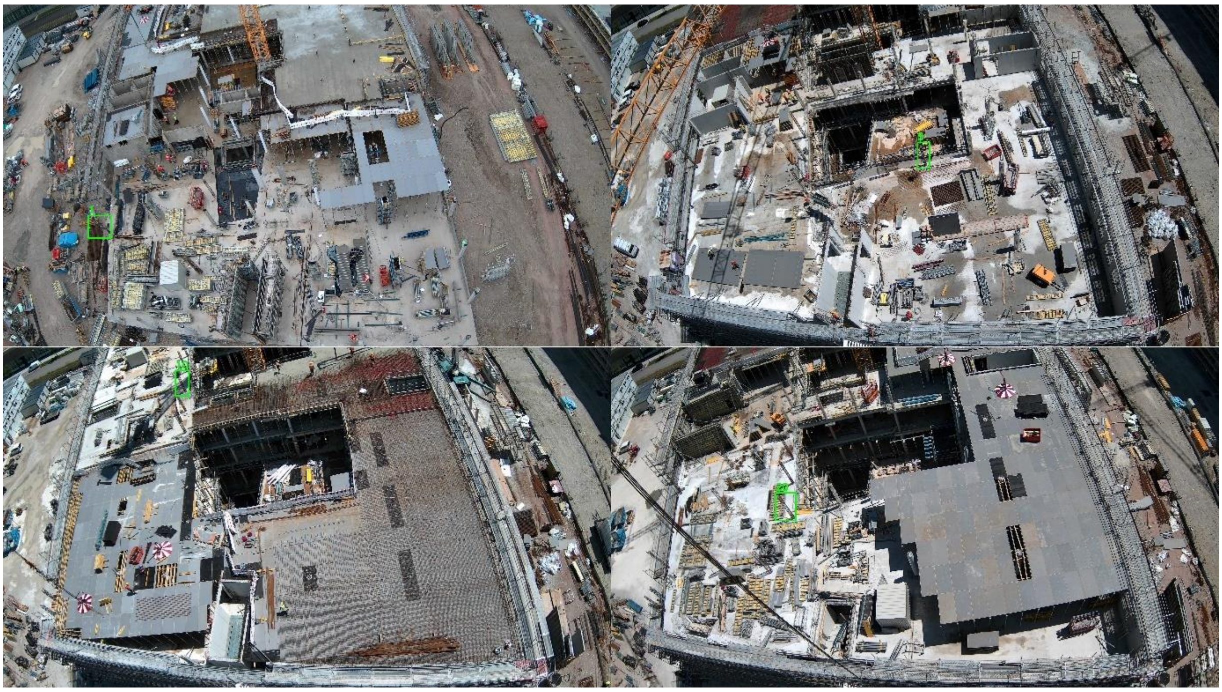


Figure 21: Captured images of some IfcColumn elements

The as-planned and as-performed properties were then combined in one single node per element, which allowed for the comparison between planned and performed necessary efforts in terms of man-hours. Table 7 summarizes the results for each of the 20 IfcColumn elements compared, with a column for the “As-Planned Effort” in man-hours, calculated based on the geometry and information from the BIM model and a column “As-Performed Effort”, calculated by Pfitzner et al. (2023). Finally, the column “Variation” showcases the percentage variation between the as-planned and as-performed efforts and calculated as follows, as presented in Chapter 4:

$$\mathrm{Variation} = \frac{(\mathrm{AsPerformedEffort}) - (\mathrm{AsPlannedEffort})}{\mathrm{AsPlannedEffort}}$$

This initial comparison shows an outliner amongst the columns with a round cross section, “STB Stütze - rund:STB ø400:5267896”, which presented a variation of 121,8% in relation to its as-planned effort. Columns with a square cross section overall required a larger effort than what was originally planned, with average variation of 67.1%, with results ranging from 50.8% to 85,9%, and standard deviation of 18.94%, for columns with different cross section area size. The circular columns presented an average variation of 13,4% between as-planned and as-performed, but its variation values ranged from -56,8% to 121,8%, which amounts to a higher standard deviation of 39,63%.

The results of grouping the columns based on the geometry of its cross section show that the variation results for square columns have a less than half of the standard deviation of the circular columns. At first glance, these results suggest that the future efforts of square columns could be predicted with more precision, despite a higher variation in comparison with the original as-planned calculated effort. However, the effect of other column properties and information in these results were still analysed and considered in this chapter.

Table 7: As-Planned and As-Performed Effort comparison

Name	As-Planned Effort (MH)	As-Performed Effort (MH)	Variation
STB Stütze - rechteckig:STB 300 x 400:6781342	75.3	140.0	85.9%
STB Stütze - rund:STB ø350:5256484	46.3	50.0	7.9%
STB Stütze - rund:STB ø350:5256482	46.3	60.0	29.5%
STB Stütze - rund:STB ø350:5256478	46.3	50.0	7.9%
STB Stütze - rund:STB ø350:5256456	46.3	50.0	7.9%
STB Stütze - rund:STB ø350:5256524	46.3	50.0	7.9%
STB Stütze - rund:STB ø350:5256480	46.3	50.0	7.9%
STB Stütze - rund:STB ø400:5267896	58.6	130.0	121.8%
STB Stütze - rund:STB ø350:5267802	46.3	70.0	51.1%
STB Stütze - rund:STB ø400:5267904	58.6	70.0	19.4%
STB Stütze - rund:STB ø500:5271704	87.4	50.0	-42.8%
STB Stütze - rund:STB ø500:5271702	87.4	80.0	-8.4%
STB Stütze - rund:STB ø500:5270507	87.4	90.0	3.0%
STB Stütze - rund:STB ø400:5267894	58.6	80.0	36.5%
STB Stütze - rund:STB ø350:5267876	46.3	20.0	-56.8%
STB Stütze - rund:STB ø350:5267836	46.3	60.0	29.5%
STB Stütze - rechteckig:STB 250 x 250:6704328	33.2	60.0	81.0%
STB Stütze - rechteckig:STB 250 x 250:6704326	33.2	50.0	50.8%
STB Stütze - rund:STB ø500:5271700	87.4	80.0	-8.4%
STB Stütze - rechteckig:STB 250 x 250:6780537	33.2	50.0	50.8%

Next, the columns were grouped by the shape of the cross-section, and further grouped by the diameter of their cross-section, and the results are presented in Table 8. This was done to assess if the geometric specificities of the elements could influence the construction process, for example, if assembling a square-shaped cross-section would be harder than a circular cross-section, or an element of larger diameter in comparison with a smaller one, based on the shape of the formwork or placement of the rebar.

For the circular columns, the elements with 500 mm diameter were the only type with a lower as-performed effort on average than the as-planned estimation, their variation results were also the most consistent for its category, with the lowest standard deviation of 19.82%. Circular columns with 400 mm diameter presented the highest average variation, 59,2% and also the highest standard deviation, of 54,84%.

The nodes representing these columns carried a series of other properties that were correlated with the variation results in order to analyse potential causal effects. Here’s a description of some of these properties and what kind of insights can be extracted from them with the methodology:

- **Geometric Information:** the nodes have stored information about the geometry of the cross section of the columns, including shape and size. If by grouping the variation results by cross section geometry of cross section size, some patterns are identified, the results may indicate that the geometry has elements influencing in the necessary efforts. Based on this insight, the project manager can investigate further on site as to why the geometry may be a cause.
- **IfcLocalPlacement:** these nodes also had information about the exact placement of the element in the model, which could provide information about how the surrounding area could be affecting productivity. If elements in closer proximity are grouped and patterns in the variation results are identified, the project manager can investigate further on site to identify what factors in that specific proximity may be influencing productivity and reflect this insight in the effort calculation for the rest of the project.
- **Start_Date and End_Date:** these nodes also had the start and finish date and time in which these columns were built, which provides information about the season of the year, and more precisely about the weather conditions in the exact period of execution. By comparing the variation results with the information about the weather of each day, the project manager can identify if rain and temperature have any effect on the variation factor and what is the amount of daily rain and temperature increase and decrease necessary to actually impact activities. Then, these factors can be taken into account for effort calculation for the rest of the project.
- **Storey:** the nodes also had information about the storey in which the element was built, which could also influence productivity if the accessibility of worker and materials to higher storeys would be lower.

The columns were also grouped by the different storeys in which they were built. The results demonstrated that the as-performed effort in the second storey was overall smaller and closer to the as-planned effort than in the third storey, and also had the lowest standard deviation, of 8,81%. This could be due to many different factors, such as the third Storey being difficult to access, or being more disorganized, or having a less-efficient work setup for a good activity flow. The exact cause needs to be assessed on site and the findings incorporated into the new effort calculation.

Table 8: Group analysis of effort for execution of columns

Group	Average As-Planned Effort (MH)	Average As-Performed Effort (MH)	Variation	Standard Deviation
STB Stütze - rund (round cross section)	58.9	65.0	13.4%	39.63%
STB Stütze - rechteckig (rectangular cross section)	43.7	75.0	67.1%	18.94%
STB Stütze - rund:STB ø350 (round cross section)	46.3	51.1	10.3%	29.44%
STB Stütze - rund:STB ø400 (round cross section)	58.6	93.3	59.2%	54.84%
STB Stütze - rund:STB ø500 (round cross section)	87.4	75.0	-14.2%	19.82%
Storey 2 - STB Stütze - rund	46.3	51.7	11.5%	8.81%
Storey 3 - STB Stütze - rund	66.4	73.0	14.5%	50.70%
Storey 3 - STB Stütze - rechteckig	33.2	55.3	60.9%	17.41%

The construction site presented weather reports, which were matched to the Start_Date and End_Date information of each column node to calculate the total amount of millimetres of rain during the whole construction process for each of these elements. Then, a new attribute called “Rain mm x min” was added to these nodes based on this calculation, as shown in Table 9. In this specific case study, the construction period of the evaluated columns was centred in June and July, months with sunny weather in Munich, so, there wasn’t enough variation in the amount of rain to impact production significantly. However, in projects that span over the course of an entire year or more than one year, more information can be collected regarding the impact of weather seasonality in the activities. The goal of calculating and implementing the “Rain mm x min” attribute here was just to highlight the flexibility of the methodology, as well as the interconnectivity with other types of data sources.

Table 9: Analysis of correlation between rain and necessary effort

Name	Start_date	End_Date	Rain mm x min	As-Planned Effort (MH)	As-Performed Effort (MH)	Variation
STB Stütze - rechteckig:STB 300 x 400:6781342	08.04.22	22.04.22	103.8	75.3	140.0	85.9%
STB Stütze - rund:STB ø350:5256484	23.06.22	28.06.22	236.2	46.3	50.0	7.9%
STB Stütze - rund:STB ø350:5256482	23.06.22	30.06.22	236.5	46.3	60.0	29.5%
STB Stütze - rund:STB ø350:5256478	23.06.22	28.06.22	236.2	46.3	50.0	7.9%
STB Stütze - rund:STB ø350:5256456	23.06.22	28.06.22	236.2	46.3	50.0	7.9%
STB Stütze - rund:STB ø350:5256524	23.06.22	29.06.22	236.5	46.3	50.0	7.9%
STB Stütze - rund:STB ø350:5256480	23.06.22	28.06.22	236.2	46.3	50.0	7.9%
STB Stütze - rund:STB ø400:5267896	05.07.22	18.07.22	30.7	58.6	130.0	121.8%
STB Stütze - rund:STB ø350:5267802	05.07.22	12.07.22	30.7	46.3	70.0	51.1%
STB Stütze - rund:STB ø400:5267904	05.07.22	12.07.22	30.7	58.6	70.0	19.4%
STB Stütze - rund:STB ø500:5271704	20.07.22	25.07.22	31.8	87.4	50.0	-42.8%
STB Stütze - rund:STB ø500:5271702	20.07.22	28.07.22	32.3	87.4	80.0	-8.4%
STB Stütze - rund:STB ø500:5270507	20.07.22	30.07.22	41.5	87.4	90.0	3.0%
STB Stütze - rund:STB ø400:5267894	20.07.22	28.07.22	32.3	58.6	80.0	36.5%
STB Stütze - rund:STB ø350:5267876	20.07.22	22.07.22	0.8	46.3	20.0	-56.8%
STB Stütze - rund:STB ø350:5267836	20.07.22	27.07.22	32.3	46.3	60.0	29.5%
STB Stütze - rechteckig:STB 250 x 250:6704328	20.07.22	27.07.22	32.3	33.2	60.0	81.0%
STB Stütze - rechteckig:STB 250 x 250:6704326	20.07.22	25.07.22	31.8	33.2	50.0	50.8%
STB Stütze - rund:STB ø500:5271700	21.07.22	30.07.22	41.5	87.4	80.0	-8.4%
STB Stütze - rechteckig:STB 250 x 250:6780537	21.07.22	27.07.22	32.3	33.2	50.0	50.8%

Based on the insights of this step, the original construction plan is then revised considering the results of the comparison between the as-planned and as-performed data from the previous step. The analysis shows that factors such as the storey, and type of column and the geometry of the column has some influence in how the necessary efforts should be estimated. However, for this case study, only twenty columns were compared, since its main goal was to evaluate the effectiveness of the methodology, and the effectiveness of the steps necessary to collected and combine the information for construction project management decision making.

In a real-life application, it is recommended that this comparison is made for most of the elements, in order to increase the amount of data and extract more robust conclusion from it.

Once conclusions are made about what adjustments need to be made in the effort calculation, the new necessary efforts need to be calculated and the new necessary time for the conclusion of these elements needs to be reflected on the working construction schedule.

Regarding the periodicity in which these adjustments need to be made, it may depend on the specific construction project management strategy. As it was discussed in Chapter 2, Lean Construction principles suggest the Six Weeks Look Ahead practice, in which midterm planning is assessed in weekly meetings to eliminate constraints for activities planned in six weeks in the future. After a few months of data gathering regarding the construction process, the information can be evaluated weekly with this methodology to adapt the construction plan to these insights. Eventually the updated effort calculations performed weekly will reflect the reality of construction enough so that new assessments won’t provide any new insights, and the periodic update will no longer be necessary.

Chapter 6

Discussion and Limitations

Chapter 6 discusses the outcomes of the methodology presented in the previous sections and analyses the findings of the case study in Chapter 5. The focus is on evaluating how effectively the proposed approach addressed the research objectives and the identified research gap.

Additionally, we will consider the limitations encountered during the research, reflecting on areas where the methodology might need further development or where certain challenges were identified.

6.1 Results Discussion

6.1.1 Integration of As-Planned and As-Performed Data

The results underscore the potential benefits and practical applications of using linked data and RDF graphs in combination with construction monitoring methodologies to enhance productivity assessments. The integration of as-planned and as-performed data, as explored in the case study, provides a more dynamic and comprehensive approach to project management. Specifically, the ability to extract and analyze data from the BIM model in combination with the construction progress monitoring information allowed for precise monitoring and evaluation of the construction project's progress, since the graph database structure allows for each individual construction element to be compared in a node-by-node basis.

The comparison between as-planned and as-performed data revealed the possibility of identification of significant insights into discrepancies in the effort calculation and the reality, which facilitated better decision-making and course correction throughout the project's execution and was facilitated by the graph database structure proposed in the methodology.

The use of RDF graphs to store and query these results made the data easily accessible for stakeholders and enabled real-time adjustments to the schedule. The case study indicated that the overall productivity of the construction process can be analyzed with the use of information about the columns stored in the graphs, which allowed for comparisons about the impact of geometries and rainy weather, for example, in the necessary effort to construct these elements.

6.1.2 Potential for Scalability and Versatility

The case study underscores the scalability potential of using linked data and RDF graphs for productivity assessment in construction projects. This is due to the unified structure created with the presented tools, which allows various data sources, such as geometric and environmental factors, to be integrated. This capability is particularly crucial when managing large-scale projects where data points related to different building elements need to be analyzed and cross-referenced. The use of linked data in this study also showed how combining multiple data points, including environmental conditions, geometrical complexities, and project schedules, can lead to a more comprehensive understanding of construction productivity.

With tools like IfcOpenShell for extracting geometric and semantic data, the linked data system can scale across different projects and be adapted for various types of construction projects. So, if a construction company employs this methodology for several projects, for example, conclusions about constructability of specific elements in one project can be extrapolated to future projects if the conditions surrounding the construction of those elements are similar enough.

Moreover, the methodology offers versatility, considering its potential to accommodate additional data sources. By building the RDF graph infrastructure with scalable ontologies, such as BOT and other domain-specific schemas, the system can integrate future enhancements, such as productivity data for more intricate construction processes. This means that as data capture methods evolve, whether through drone monitoring or advanced IoT integration, these can be readily incorporated into the existing framework. This scalability not only applies to the number of building elements being monitored but also to the system's ability to handle various construction scenarios, making it a highly adaptable solution for both small and large-scale projects, as long as the necessary infrastructure is implemented.

6.2 Limitations

6.2.1 Preprocessing Challenges

The methodology encountered challenges during the preprocessing stage of the IFC model for RDF graph conversion. As presented in Chapter 4, the use of tools such as IfcOpenShell and SimpleBIM was essential to optimize the IFC model by removing unnecessary data that would not contribute to the analysis. However, this process required manual effort, a level of expertise, and attention to detail to ensure proper data filtering and prevent errors during the conversion process.

Another major challenge in the preprocessing stage was related to overall efficiency when dealing with larger projects or more complex models than the case study.

For instance, although the methodology worked well for the specific set of building elements used in the case study—primarily IfcColumn—it was not demonstrated how effective the process would be when applied to a broader array of building elements, or intricate architectural features. As the scale and complexity of a construction project increase, so do the demands on data preprocessing if the user wants to monitor several types of building elements, since the user will need to define for each of those element types which triples should be maintained or not and adapt their preprocessing algorithms to this.

Additionally, the need to retain geometric and semantic information relevant to productivity assessments while eliminating redundant data can lead to inefficiencies in managing larger datasets. This limitation points to the need for developing more automated methods to handle preprocessing tasks. Elements with complex and unconventional geometry may also pose a challenge, as determining the correct geometric information needed to assess construction efforts may be less intuitive and straightforward.

6.2.2 Generalizability of Findings

The case study focused primarily on a single construction project in Munich, examining a limited subset of building elements, predominantly columns. While the methodology proved effective for the selected elements, it remains uncertain how well it would perform when applied to different construction tasks or larger, more complex projects.

Even though scalability is possible, the construction tasks that involve more intricate or multi-layered elements might pose challenges for real-time progress monitoring, as these elements are harder to capture accurately using current methods. Moreover, collecting detailed information for each building element on a larger construction site could create additional challenges in terms of data processing and query efficiency, especially when the number of elements grows substantially. A solution for this may be to not perform the as-planned and as-performed comparison in real-time, but in a less frequent periodicity, even though the as-performed data is being collected in real-time.

Another factor affecting the generalizability is the specific nature of the construction elements studied. Columns typically have predictable geometric forms and well-defined construction processes. However, this may not be the case for other elements such as unique architectural features that require custom construction techniques or materials. As such, the productivity rate calculation step of methodology needs to be adapted when applied to different element types or projects with non-standard construction methods.

Additionally, as the scale of the project increases, so does the complexity of managing and querying the RDF graphs, raising concerns about the system's ability to maintain efficiency. Query performance and data management become significant issues when the system has to handle hundreds or thousands of elements simultaneously. For example, if querying the 304 as-planned column elements represented in the case study would amount to a few seconds of query time, this would drastically increase if the methodology was being applied also other building element types such as walls, slabs and beams, and in a much larger project.

6.2.3 Data Acquisition Constraints

The productivity analysis was constrained by the availability of data, meaning that external factors such as subcontractor efficiency, supply chain delays, and unanticipated project changes were not integrated into the model. Expanding the data sources to include these factors would provide a more comprehensive understanding of construction productivity, but it also offers a challenge in terms of how this information can be accurately measured and properly associated with the building elements in the graph.

Additionally, the method's reliance on real-time data presents another challenge. The continuous integration of real-time data, although beneficial, requires advanced infrastructure and reliable internet connectivity, which may not be available on all construction sites. The cost associated with setting up and maintaining such systems may limit their adoption, particularly for smaller projects with fewer resources. A possible solution, as mentioned earlier, is to perform the as-planned and as-performed data integration and comparison at less frequent intervals, even if the as-performed data is collected in real time.

These limitations highlight areas where future research and development are needed to improve the robustness and scalability of the proposed methodology.

6.3 Research Gap Discussion

The identified research gap revolves around the need for an integrated end-to-end framework that can effectively combine as-planned and as-performed data to assess productivity in construction projects. This gap highlights the absence of a standardized methodology that facilitates real-time tracking and comparison of planned versus actual construction progress. The methodology proposed in the thesis, which involves the use of BIM models, IFC to RDF conversion, and data-driven monitoring, is designed to address this challenge. It emphasizes the use of linked data technologies to provide a dynamic and adaptable structure that can integrate different data sources, allowing for continuous updates and progress tracking throughout the project lifecycle.

Throughout the thesis, various aspects of the proposed methodology are tested and validated. For instance, the case study demonstrates how as-performed data is integrated with as-planned BIM models using RDF graphs. This allows for an efficient comparison between the expected and actual progress, and the detection of deviations. The thesis demonstrates that this integration enhances productivity tracking, allowing for real-time adjustments to the project plan. While the framework effectively integrates data and allows for productivity assessment, there are still challenges to generalize this approach for more complex projects, as discussed in this chapter.

Overall, the thesis successfully addresses the research gap by proposing a robust end-to-end methodology. However, it opens new directions for improving the adaptability and scalability of this approach in different construction scenarios, and connecting the insights with an automated construction schedule update.

Chapter 7

Future Work and Conclusion

Chapter 7 summarizes the main conclusions of the research, while also identifying potential directions for future work. The methodology proposed and applied in the case study demonstrated several promising outcomes. However, there are some limitations and challenges that arose during the research process, which point to areas that require further exploration. This chapter will first discuss the directions for future research, followed by the final conclusions drawn from the study.

7.1 Future Work

Based on the limitations discussed in Chapter 6, there are some areas that future research could address to enhance the proposed methodology. One of the key areas for improvement is the automation of the preprocessing stage. As mentioned in Section 6.2.1, the manual preprocessing required for IFC models before they can be converted into RDF graphs was labor-intensive and error prone. Automating this process, or at least semi-automating it, would reduce the time and effort required, thereby making the methodology more scalable and applicable to larger and more complex construction projects. Future work could explore the development of scripts or tools that automatically streamline and optimize the IFC-to-RDF conversion process.

Another area for future work is exploring the applicability of the findings in other construction project types and tasks. As noted in Section 6.2.2, the case study focused primarily on a single construction project involving columns, which limits the generalizability of the results. Further studies could apply this methodology to a wider range of construction elements and in different types of construction projects. Additionally, exploring how other external factors such as subcontractor performance, supply chain delays, or material quality affect productivity could provide a more holistic view of construction performance. Incorporating these factors would make the system more robust and practical for real-world applications.

The integration of the insights from the as-planned and as-performed comparison and analysis into an updated construction schedule can also be explored in future work. This topic can be explored in terms of how to operationalize the process, as well as how to incorporate the variation update in a way that considers the interdependencies between activities and all other factors influencing scheduling.

Lastly, advancing data acquisition technologies is critical. In Chapter 6, limitations were discussed regarding the dependency on image data and the need for better integration of real-time data from diverse sources, including IoT sensors and weather information.

Future research could explore the integration of advanced IoT systems or drone-based monitoring to provide continuous and accurate data. Additionally, optimizing the querying performance of RDF graphs in large-scale projects remains a challenge that needs to be addressed to ensure that the system can handle larger datasets.

7.2 Conclusion

This thesis presented an approach for integrating RDF graphs into construction progress monitoring, using as-planned and as-performed data to assess productivity and identify discrepancies in real-time, or in another periodicity, depending on the project management requirements. The methodology was successfully applied to a case study in Munich, where construction progress for a set of concrete columns was monitored, and insights were gained regarding the factors that influence productivity, such as geometric complexity and environmental conditions. By utilizing RDF graphs, the data collected from the BIM model and progress monitoring systems were made more accessible and queryable, enabling stakeholders to adjust schedules and improve project outcomes based on real-time insights.

A key contribution of this research is the combination of data-driven methodologies with linked data technology to create a comprehensive end-to-end productivity assessment tool. The methodology demonstrates how various data sources can be integrated into a unified system to improve decision-making in construction project management. Furthermore, the potential for scalability and the versatility of the system suggest that this approach could be applied to a wide range of construction projects.

However, some limitations were identified, particularly in the areas of data preprocessing, scalability, and generalization. While the research demonstrated the feasibility of the approach, further investigation is required to streamline the methodology and ensure its applicability across a broader range of construction tasks and environments. Nonetheless, this thesis provides a solid foundation for future work in the field of Digital Twin Construction, demonstrating the potential for linked data and RDF technologies to contribute to construction progress monitoring and productivity assessment.

Appendix A

RDFLib Python Script for Triple Filtering

Phase 1 – RDF Graph Filtering

```
import rdflib
from tqdm import tqdm

# Paths to the input and output TTL files
ttl_file_path = "..."
output_file_path = "..."

# Define namespaces
IFC = rdflib.Namespace("https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#")
INST = rdflib.Namespace("https://www.ugent.be/myAwesomeFirstBIMProject#")

# Class type to filter for (change this variable to filter for different class types)
class_type_to_filter = "IfcWall"  # Change to "IfcSlab" or other class types as needed

# Prefixes
prefixes = """
@base   <https://www.ugent.be/myAwesomeFirstBIMProject#> .
@prefix ifc: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#> .
@prefix inst: <https://www.ugent.be/myAwesomeFirstBIMProject#> .a
@prefix list: <https://w3id.org/list#> .
@prefix express: <https://w3id.org/express#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
"""

def process_chunk(chunk, graph):
    relationships = []
    try:
        graph.parse(data=chunk, format='turtle')
        for subj, pred, obj in graph:
            if (isinstance(subj, rdflib.URIRef) and str(subj).startswith(str(INST)) and class_type_to_filter in str(subj)) or \
               (isinstance(obj, rdflib.URIRef) and str(obj).startswith(str(INST)) and class_type_to_filter in str(obj)):
                relationships.append((subj, pred, obj))
                print(f"Subject: {subj}, Predicate: {pred}, Object: {obj}")
    except Exception as e:
        print(f"Error parsing chunk: {e}")
    return relationships

def filter_ifc_elements(ttl_file_path, output_file_path, class_type):
    # Initialize the RDF graph
    filtered_graph = rdflib.Graph()
    filtered_graph.bind("ifc", IFC)
    filtered_graph.bind("inst", INST)

    # Count total lines for progress bar
    total_lines = sum(1 for _ in open(ttl_file_path, 'r', encoding='utf-8'))

    # Process the file line by line with a progress bar
    chunk = prefixes  # Start each chunk with prefixes
    with open(ttl_file_path, 'r', encoding='utf-8') as file:
        for line in tqdm(file, total=total_lines, desc="Processing TTL file"):
            chunk += line
            if line.strip().endswith('.'):
                # Bind prefixes for each chunk
                temp_graph = rdflib.Graph()
                temp_graph.bind("ifc", IFC)
                temp_graph.bind("inst", INST)

                chunk_relationships = process_chunk(chunk, temp_graph)
                for rel in chunk_relationships:
                    filtered_graph.add(rel)
                chunk = prefixes  # Reset chunk to start with prefixes

    # Write the filtered triples to the new TTL file
    with open(output_file_path, 'w', encoding='utf-8') as out_file:
        out_file.write(prefixes)
        out_file.write(filtered_graph.serialize(format='turtle'))

    print(f"Filtered triples saved to {output_file_path}")

# Run the filter function for the entire file
filter_ifc_elements(ttl_file_path, output_file_path, class_type_to_filter)
```

Phase 2 – IfcGloballyUniqueId Insertion

```
from rdflib.namespace import RDF, OWL
from tqdm import tqdm
import os

# Paths to the TTL files
ifc_wall_ttl_path = "..."
scg_ar_ttl_path = "..."
output_ttl_path = "..."

# Namespaces
IFC = rdflib.Namespace("https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#")
INST = rdflib.Namespace("https://www.ugent.be/myAwesomeFirstBIMProject#")
EXPRESS = rdflib.Namespace("https://w3id.org/express#")

# Prefixes
prefixes = """
@base   <https://www.ugent.be/myAwesomeFirstBIMProject#> .
@prefix ifc: <https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#> .
@prefix inst: <https://www.ugent.be/myAwesomeFirstBIMProject#> .
@prefix list: <https://w3id.org/list#> .
@prefix express: <https://w3id.org/express#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
"""

print("Loading IFC Wall graph...")
# Load the IFC Wall graph
ifc_wall_graph = rdflib.Graph()
ifc_wall_graph.parse(ifc_wall_ttl_path, format="ttl")

print("Collecting all IfcGloballyUniqueId instances...")
# Collect all IfcGloballyUniqueId instances
globally_unique_ids = set()
for subj, pred, obj in ifc_wall_graph.triples((None, IFC.globalId_IfcRoot, None)):
    if obj.startswith(INST):
        globally_unique_ids.add(obj)

print(f"Found {len(globally_unique_ids)} IfcGloballyUniqueId instances.")

# Initialize the SCG AR graph and bind namespaces
scg_ar_graph = rdflib.Graph()
scg_ar_graph.bind("ifc", IFC)
scg_ar_graph.bind("inst", INST)
scg_ar_graph.bind("express", EXPRESS)

print("Loading SCG AR graph...")
total_lines = sum(1 for _ in open(scg_ar_ttl_path, 'r', encoding='utf-8'))

# Function to copy triples
def copy_triples(source_graph, target_graph, subject):
    for s, p, o in source_graph.triples((subject, None, None)):
        target_graph.add((s, p, o))

# Process the SCG AR TTL file in chunks
def process_in_chunks(file_path, chunk_size=100000):
    chunk = prefixes
    line_count = 0
    with open(file_path, 'r', encoding='utf-8') as file:
        for line in tqdm(file, total=total_lines, desc="Reading SCG AR TTL file"):
            chunk += line
            line_count += 1
            if line_count % chunk_size == 0 or line.strip().endswith('.'):
                try:
                    temp_graph = rdflib.Graph()
                    temp_graph.parse(data=chunk, format="ttl")
                    for guid in globally_unique_ids:
                        copy_triples(temp_graph, ifc_wall_graph, guid)
                except Exception as e:
                    print(f"Error parsing chunk: {e}")
                chunk = prefixes
                line_count = 0
        if chunk.strip():  # Process any remaining data in the chunk
            try:
                temp_graph = rdflib.Graph()
                temp_graph.parse(data=chunk, format="ttl")
                for guid in globally_unique_ids:
                    copy_triples(temp_graph, ifc_wall_graph, guid)
            except Exception as e:
                print(f"Error parsing final chunk: {e}")

print("Processing SCG AR TTL file in chunks...")
process_in_chunks(scg_ar_ttl_path)

print("Writing the updated graph to a new TTL file...")
# Write the updated graph to a new TTL file
with open(output_ttl_path, 'w', encoding='utf-8') as out_file:
    out_file.write(ifc_wall_graph.serialize(format='turtle'))

print(f"Updated TTL file saved to {output_ttl_path}")
```

Appendix B

Built Element Classes Triple Types List

Table 10: IfcWall RDF Graph Triple Types List

Subject_Type	Predicate	Object_Type	Count
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelDefinesByProperties	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatedObjects_IfcRelDefinesByProperties	https://www.ugent.be/myAwesomeFirstBIMProject#IfcWall	14423
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelAssociatesClassification	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatedObjects_IfcRelAssociates	https://www.ugent.be/myAwesomeFirstBIMProject#IfcWall	1666
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWall	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWall	1476
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWall	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#predefinedType_IfcWall	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#NOTDEFINED	1476
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWall	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#ownerHistory_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMProject#IfcOwnerHistory_8	1476
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWall	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#objectType_IfcObject	https://www.ugent.be/myAwesomeFirstBIMProject#IfcLabel	1476
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWall	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#name_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMProject#IfcLabel	1476
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelDefinesByType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatedObjects_IfcRelDefinesByType	https://www.ugent.be/myAwesomeFirstBIMProject#IfcWall	1476
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWall	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#globalId_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMProject#IfcGloballyUniqueId	1476
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelContainedInSpatialStructure	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatedElements_IfcRelContainedInSpatialStructure	https://www.ugent.be/myAwesomeFirstBIMProject#IfcWall	1476
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWall	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#representation_IfcProduct	https://www.ugent.be/myAwesomeFirstBIMProject#IfcProductDefinitionShape	1476
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWall	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#objectPlacement_IfcProduct	https://www.ugent.be/myAwesomeFirstBIMProject#IfcLocalPlacement	1476
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWall	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#tag_IfcElement	https://www.ugent.be/myAwesomeFirstBIMProject#IfcIdentifier	1476
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelVoidsElement	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatingBuildingElement_IfcRelVoidsElement	https://www.ugent.be/myAwesomeFirstBIMProject#IfcWall	1149
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWallType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#hasPropertySets_IfcTypeObject	https://www.ugent.be/myAwesomeFirstBIMProject#IfcPropertySet	578
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWallType	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWallType	121
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWallType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#ownerHistory_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMProject#IfcOwnerHistory_8	121
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelDefinesByType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatingType_IfcRelDefinesByType	https://www.ugent.be/myAwesomeFirstBIMProject#IfcWallType	121
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWallType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#tag_IfcTypeProduct	https://www.ugent.be/myAwesomeFirstBIMProject#IfcLabel	121
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWallType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#name_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMProject#IfcLabel	121
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWallType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#globalId_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMProject#IfcGloballyUniqueId	121
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWallType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#predefinedType_IfcWallType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#STANDARD	116
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelAssignsToGroup	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatedObjects_IfcRelAssigns	https://www.ugent.be/myAwesomeFirstBIMProject#IfcWall	9
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcWallType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#predefinedType_IfcWallType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#NOTDEFINED	5

Table 11: IfcSlab RDF Graph Triple Types List

Subject_Type	Predicate	Object_Type	Count
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelDefinesByProperties	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatedObjects_IfcRelDefinesByProperties	https://www.ugent.be/myAwesomeFirstBIMProject#IfcSlab	3982
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlabType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#hasPropertySets_IfcTypeObject	https://www.ugent.be/myAwesomeFirstBIMProject#IfcPropertySet	1487
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlabType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#ownerHistory_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMProject#IfcOwnerHistory_8	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#ownerHistory_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMProject#IfcOwnerHistory_8	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlabType	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlabType	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlabType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#tag_IfcTypeProduct	https://www.ugent.be/myAwesomeFirstBIMProject#IfcLabel	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlabType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#name_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMProject#IfcLabel	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#objectType_IfcObject	https://www.ugent.be/myAwesomeFirstBIMProject#IfcLabel	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#name_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMProject#IfcLabel	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlabType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#globalId_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMProject#IfcGloballyUniqueId	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#globalId_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMProject#IfcGloballyUniqueId	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlab	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlab	499
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelDefinesByType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatingType_IfcRelDefinesByType	https://www.ugent.be/myAwesomeFirstBIMProject#IfcSlabType	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#tag_IfcElement	https://www.ugent.be/myAwesomeFirstBIMProject#IfcIdentifier	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#representation_IfcProduct	https://www.ugent.be/myAwesomeFirstBIMProject#IfcProductDefinitionShape	499
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelDefinesByType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatedObjects_IfcRelDefinesByType	https://www.ugent.be/myAwesomeFirstBIMProject#IfcSlab	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#objectPlacement_IfcProduct	https://www.ugent.be/myAwesomeFirstBIMProject#IfcLocalPlacement	499
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelAssociatesClassification	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatedObjects_IfcRelAssociates	https://www.ugent.be/myAwesomeFirstBIMProject#IfcSlab	499
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlabType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#predefinedType_IfcSlabType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#FLOOR	490
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#predefinedType_IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#FLOOR	490
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelContainedInSpatialStructure	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatedElements_IfcRelContainedInSpatialStructure	https://www.ugent.be/myAwesomeFirstBIMProject#IfcSlab	490
https://www.ugent.be/myAwesomeFirstBIMProject#IfcRelAggregates	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#relatedObjects_IfcRelAggregates	https://www.ugent.be/myAwesomeFirstBIMProject#IfcSlab	9
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#predefinedType_IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#LANDING	6
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlabType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#predefinedType_IfcSlabType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#LANDING	6
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#predefinedType_IfcSlab	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#ROOF	3
https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#IfcSlabType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#predefinedType_IfcSlabType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL#ROOF	3

Table 12: IfcColumn RDF Graph Triple Types List

Subject_Type	Predicate	Object_Type	Count
https://www.ugent.be/myAwesomeFirstBIMProject #IfcRelDefinesByProperties	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# relatedObjects_IfcRelDefinesByProperties	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcColumn	2354
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumnType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# hasPropertySets_IfcTypeObject	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcPropertySet	780
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcGloballyUniqueId	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	https://standards.buildingsmart.org/IFC/DEV/I FC4/ADD2/OWL#IfcGloballyUniqueId	305
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumn	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# ownerHistory_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcOwnerHistory_8	304
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumn	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	https://standards.buildingsmart.org/IFC/DEV/I FC4/ADD2/OWL#IfcColumn	304
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumn	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# predefinedType_IfcColumn	https://standards.buildingsmart.org/IFC/DEV/I FC4/ADD2/OWL#COLUMN	304
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumn	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# objectType_IfcObject	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcLabel	304
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumn	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# name_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcLabel	304
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcGloballyUniqueId	https://w3id.org/express#hasString		304
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumn	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# globalId_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcGloballyUniqueId_965210	304
https://www.ugent.be/myAwesomeFirstBIMProject #IfcRelDefinesByType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# relatedObjects_IfcRelDefinesByType	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcColumn	304
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumn	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# objectPlacement_IfcProduct	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcLocalPlacement	304
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumn	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# representation_IfcProduct	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcProductDefinitionShape	304
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumn	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# tag_IfcElement	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcIdentifier	304
https://www.ugent.be/myAwesomeFirstBIMProject #IfcRelContainedInSpatialStructure	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# relatedElements_IfcRelContainedInSpatialStructure	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcColumn	304
https://www.ugent.be/myAwesomeFirstBIMProject #IfcRelAssociatesClassification	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# relatedObjects_IfcRelAssociates	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcColumn	275
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumnType	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	https://standards.buildingsmart.org/IFC/DEV/I FC4/ADD2/OWL#IfcColumnType	264
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumnType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# ownerHistory_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcOwnerHistory_8	264
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumnType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# predefinedType_IfcColumnType	https://standards.buildingsmart.org/IFC/DEV/I FC4/ADD2/OWL#COLUMN	264
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumnType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# name_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcLabel	264
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumnType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# tag_IfcTypeProduct	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcLabel	264
https://standards.buildingsmart.org/IFC/DEV/IFC4/A DD2/OWL#IfcColumnType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# globalId_IfcRoot	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcGloballyUniqueId_3201127	264
https://www.ugent.be/myAwesomeFirstBIMProject #IfcRelDefinesByType	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# relatingType_IfcRelDefinesByType	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcColumnType	264
https://www.ugent.be/myAwesomeFirstBIMProject #IfcRelAssignsToGroup	https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2/OWL# relatedObjects_IfcRelAssigns	https://www.ugent.be/myAwesomeFirstBIMP roject#IfcColumn	10

Appendix C

IfcOpenShell Python Script for Node Data Enrichment

Phase 1 – IFC Property Extraction

```
import ifcopenshell
import pandas as pd
import os

# Function to extract quantities and additional properties from an IFC element
def extract_quantities_and_properties(element, properties):
    data = {prop: None for prop in properties}
    data["IfcGloballyUniqueId"] = element.GlobalId
    data["Name"] = element.Name

    # Extract IfcLocalPlacement
    if element.ObjectPlacement:
        data["IfcLocalPlacement"] = element.ObjectPlacement.id()

    # Extract IfcIdentifier and IfcLabel
    for definition in element.IsDefinedBy:
        if definition.is_a("IfcRelDefinesByProperties"):
            property_set = definition.RelatingPropertyDefinition
            if property_set.is_a("IfcPropertySet"):
                for property in property_set.HasProperties:
                    if property.Name == "Identifier":
                        data["IfcIdentifier"] = property.NominalValue.wrappedValue
                    elif property.Name == "Label":
                        data["IfcLabel"] = property.NominalValue.wrappedValue

    # Extract quantities
    for definition in element.IsDefinedBy:
        if definition.is_a("IfcRelDefinesByProperties"):
            property_set = definition.RelatingPropertyDefinition
            if property_set.is_a("IfcElementQuantity"):
                for quantity in property_set.Quantities:
                    if quantity.is_a("IfcQuantityVolume"):
                        if "GrossVolume" in quantity.Name:
                            data["GrossVolume"] = quantity.VolumeValue
                        elif "NetVolume" in quantity.Name:
                            data["NetVolume"] = quantity.VolumeValue
                    elif quantity.is_a("IfcQuantityArea"):
                        if "OuterSurfaceArea" in quantity.Name:
                            data["OuterSurfaceArea"] = quantity.AreaValue
                        elif "CrossSectionArea" in quantity.Name:
                            data["CrossSectionArea"] = quantity.AreaValue
                    elif quantity.is_a("IfcQuantityLength") and "Length" in quantity.Name:
                        data["Length"] = quantity.LengthValue
    return data

# Function to extract data from IFC elements and save to Excel
def extract_ifc_data(ifc_file_path, ifc_type, output_file_name, properties):
    # Open the IFC file
    ifc_file = ifcopenshell.open(ifc_file_path)

    # List to store the data
    data = []

    # Iterate through elements of the specified type
    elements = ifc_file.by_type(ifc_type)
    for element in elements:
        extracted_data = extract_quantities_and_properties(element, properties)
        data.append(extracted_data)

    # Convert to DataFrame
    df = pd.DataFrame(data)

    # Save to Excel in the same path as the input IFC file
    output_excel_path = os.path.join(os.path.dirname(ifc_file_path), output_file_name)
    df.to_excel(output_excel_path, index=False)

    print(f"Data saved to {output_excel_path}")

# Example usage for IfcColumn
ifc_file_path = "                                                  /1. Data/SCG_AR (EDITED).ifc"
output_file_name = "IfcColumn_data.xlsx"
ifc_type_to_filter = "IfcColumn"
properties_to_extract = ["IfcGloballyUniqueId", "Name", "GrossVolume", "OuterSurfaceArea", "NetVolume", "CrossSectionArea", "Length",
                         "IfcLocalPlacement", "IfcIdentifier", "IfcLabel"]

extract_ifc_data(ifc_file_path, ifc_type_to_filter, output_file_name, properties_to_extract)
```

Phase 2 – RDF Graph Node Properties Update

```python
import pandas as pd
from neo4j import GraphDatabase

# Load the Excel file
file_path = '...'
df = pd.read_excel(file_path)

# Connect to Neo4j database
uri = "bolt://localhost:7687"
username = "neo4j"
password = "Pockm@rkedch1n!"

driver = GraphDatabase.driver(uri, auth=(username, password))

def update_ifc_column(tx, guid, name, gross_volume, outer_surface_area, net_volume, cross_section_area, length, ifc_local_placement):
    query = """
    MATCH (column:Resource:ns13__IfcColumn)-[:ns13__globalId_IfcRoot]->(guid:Resource:ns13__IfcGloballyUniqueId)
    WHERE guid.ns14__hasString = $guid
    SET column.Name = $name,
        column.GrossVolume = $gross_volume,
        column.OuterSurfaceArea = $outer_surface_area,
        column.NetVolume = $net_volume,
        column.CrossSectionArea = $cross_section_area,
        column.Length = $length,
        column.IfcLocalPlacement = $ifc_local_placement
    RETURN column
    """
    result = tx.run(query, guid=guid, name=name, gross_volume=gross_volume, outer_surface_area=outer_surface_area,
                    net_volume=net_volume, cross_section_area=cross_section_area, length=length, ifc_local_placement=ifc_local_placement)
    if result.single():
        print(f"Match found and updated for ns14__hasString: {guid}")
    else:
        print(f"No match found for ns14__hasString: {guid}")

def main():
    with driver.session() as session:
        for index, row in df.iterrows():
            session.write_transaction(
                update_ifc_column,
                row['IfcGloballyUniqueId'],
                row['Name'],
                row['GrossVolume'],
                row['OuterSurfaceArea'],
                row['NetVolume'],
                row['CrossSectionArea'],
                row['Length'],
                row['IfcLocalPlacement']
            )

if __name__ == "__main__":
    main()
```

Bibliography

McKinsey Global Institute. (2017). Reinventing Construction: A Route to Higher Productivity. McKinsey & Company.

Flyvbjerg, B., Bruzelius, N., & Rothengatter, W. (2002). Megaprojects and Risk: An Anatomy of Ambition. Cambridge University Press.

Deng, F., Cheng, J. C. P., & Anumba, C. (2016). Mapping Knowledge Flow in Green Building BIM-Based Design Processes: An Approach and Its Application. Automation in Construction, 68, 55-68.

Thomas, H. R., & Mathews, C. T. (2016). Construction Productivity: Modeling and Measurement Issues. Journal of Construction Engineering and Management, 142(5), 04015088.

Hegazy, T. Computer-Based Construction Project Management (2002). Available online:https://www.pearson.ch/HigherEducation/Pearson/EAN/9781292027128/Computer-Based-Construction-Project-Management-Pearson-New-International-Edition (accessed on 08 April 2024).

Son, H.; Bosché, F.; Kim, C. (2015): As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey. Adv. Eng. Inform, 29, 172–183. https://doi.org/10.1016/j.aei.2015.01.009

El-Sabek, L.M.; McCabe, B.Y. (2017): Coordination Challenges of Production Planning in the Construction of International Mega-Projects in The Middle East. International Journal of Construction Education and Research, 14, 118–140. https://doi.org/ 10.1080/15578771.2016.1276109

Soibelman, L., & Peña-Mora, F. (2000). Distributed multi-reasoning mechanism to support real-time project control decisions. Journal of Computing in Civil Engineering, 14(4), 207-216. https://doi.org/10.1061/(ASCE)0887-3801(2000)14:4(207)

Soibelman, L., & Kim, H. (2002). Data Preparation Process for Construction Knowledge Generation through Knowledge Discovery in Databases. Journal of Computing in Civil Engineering, 16(1), 39-48.

Navon, R.; Sacks, R. (2007): Assessing research issues in Automated Project Performance Control (APPC). Automation in Construction, 16,474–484. https://doi.org/10.1016/j.autcon.2006.08.001

Sacks, R.; Brilakis, I.; Pikas, E.; Sally Xie, H.; Girolami, M. (2020): Digital Twin Construction: Closing Control Loops in Building and Infrastructure Production Management. Data Centric Engineering. https://doi.org/10.1017/dce.2020.16

Y. Pan, L. Zhang, M.J. Skibniewski (2020): Clustering of designers based on building information modeling event logs, Computer-Aided Civil and Infrastructure Engineering. Eng. 35. 701–718. https://doi.org/10.1111/mice.12551

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

Pfitzner, F., Braun, A., & Borrmann, A. (2024): Object Detection Based Knowledge Graph Creation: Enabling Insight into Construction Processes. Computing in Civil Engineering 2023. https://doi.org/10.1061/9780784485224.023

Cheng, T., & Teizer, J. (2012). Real-time Resource Location Data Collection and Visualization Technology for Construction Safety and Activity Monitoring Applications. Automation in Construction, 34, 3-15. doi: 10.1016/j.autcon.2012.10.017

Asadi, K., & Han, K. (2018). Real-Time Image-to-BIM Registration Using Perspective Alignment for Automated Construction Monitoring. Construction Research Congress 2018, 388-397. doi: 10.1061/9780784481264.038.

Collins, F. C., Pfitzner, F., & Schlenger, J. (2022): Scalable Construction Monitoring for an As-Performed Progress Documentation Across Time. Forum Bauinformatik.

Z. Pan, C. Su, Y. Deng, J. Cheng: A computer vision-based entity extraction framework for updating the architecture, engineering and construction industry knowledge graphs, Autom. Constr. 125 (2021) 103617, http://dx.doi.org/10.1016/j.autcon.2021.103617.

M.H. Rasmussen, M. Lefrançois, P. Pauwels, C.A. Hviid, J. Karlshøj, Managing interrelated project information in AEC knowledge graphs, Autom. Constr. 108 (2019) 102956, http://dx.doi.org/10.1016/j.autcon.2019.102956.

Braun, A. (2020): Automated BIM-based Construction Progress Monitoring by Processing and Matching Semantic and Geometric Data.

Fini, A.A.F, Maghrebi, M., Forsythe, P.J. and Waller, T.S. (2022): Using existing site surveillance cameras to automatically measure the installation speed in prefabricated timber construction. Engineering, Construction and Architectural Management. Vol. 29, no. 2, pp. 573–600, doi: 10.1108/ECAM-04-2020-0281.

Paneru, S., Jeelani, I., (2021): Computer vision applications in construction: Current state, opportunities & challenges, Automation in Construction, 132 (2021) 103940. https://doi.org/10.1016/j.autcon.2021.103940.

Ekanayake, B., Wong, J.K.-W., Fini, P. (2021): Computervision-based interior construction progress monitoring: A literaturereview and future research directions, Automation in Construction, 127 (2021) 103705. https://doi.org/10.1016/j.autcon.2021.103705.

Patel, T., Guo, B.H.W., Zou, Y (2021): A scientometric review of construction progress monitoring studies, Engineering, Construction and Architectural Management. https://doi.org/10.1108/ECAM-10-2020-0799

Z. Ma, S. Liu, A review of 3D reconstruction techniques in civil engineering and their applications, Advanced Engineering Informatics, 37 (2018) pp. 163–174. https://doi.org/10.1016/j.aei.2018.05.005.

Fant, J., Li, Y., Liao, Q., Ren, Z., & Xie, B. (2018): Construction Progress Control and Management Measures Analysis. Smart Construction Research. https://doi.org/10.18063/scr.v0.392.

Wolfe, S., Jr. (2020): Construction Survey: Contractors Waste Time & Get Paid Slowly. Available online: https://www.levelset.com/ blog/2020-report-construction-wasted-time-slow-payment/ (accessed on 10 April 2024).

Howell, A., G. (1999): What is Lean Construction. Seventh Conference of the International Group for Lean Construction.

Ballard, G. (2000): The Last Planner System of Production Control. University of Birmingham, Birmingham, UK.

PBC Today. (2023). What is the Last Planner System? A guide to construction's newest project management methodology. Retrieved from https://www.pbctoday.co.uk/news/digital-construction/construction-technology-news/what-is-the-last-planner-system-a-guide-to-constructions-newest-project-management-methodology/121097/

Sacks., R, Girolami., M and Brilakis., I (2020): Building information modeling, artificial intelligence and construction tech. Developments in the Built Environment, Amsterdam, Netherlands. 100011. https://doi.org/10.1016/j.dibe.2020.100011

Deming, W. (1982): Out of the Crisis. Massachusetts Institute of Technology. https://doi.org/10.7551/mitpress/11457.001.0001

Forbes, L., and Ahmed, S. (2011): Modern Construction: Lean Project Delivery and Integrated Practices. CRC Press. https://doi.org/10.1201/b10260

Hamzeh, F., Zankoul, E., and Rouhana, C. (2015): How can "tasks made ready" during look ahead planning impact reliable workflow and project duration? Construction Management and Economics 33(4), 243–258. https://doi.org/10.1080/01446193.2015.1047878

Sacks., R, Brilakis, I., Pikas, E., Sally Xie, H., Girolami, M. (2020): Construction with digital twin information systems. Cambridge University Press. https://doi.org/10.1017/dce.2020.16

Borrman, A., König, M., Koch, C., Beetz, J. (2018): Building Information Modeling: Technology Foundations and Industry Practice. Springer. https://doi.org/10.1007/978-3-319-92862-3

Soibelman, L., & Peña-Mora, F. (2000). Distributed multi-reasoning mechanism to support real-time project control decisions. Journal of Computing in Civil Engineering, 14(4), 207-216. https://doi.org/10.1061/(ASCE)0887-3801(2000)14:4(207)

Turkan, Y., Bosché, F., Haas, C. T., & Haas, R. (2012). Automated progress tracking using 4D schedule and 3D sensing technologies. Automation in Construction, 22, 414-421. https://doi.org/10.1016/j.autcon.2011.10.003

Bosché, F., & Guenet, E. (2014). Automating the comparison of 4D as-built and 4D as-planned using the Kinect sensor. Automation in Construction, 44, 107-122. https://doi.org/10.1016/j.autcon.2014.04.004

Golparvar-Fard, M., Peña-Mora, F., & Savarese, S. (2011). Integrated sequential as-built and as-planned representation with reality capture technologies for project monitoring and control. Journal of Construction Engineering and Management, 137(12), 1099-1116. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000373

Teizer, J., Allread, B. S., Fullerton, C. E., & Hinze, J. (2010). Autonomous pro-active real-time construction worker and equipment operator proximity safety alert system. Automation in Construction, 19(5), 630-640. https://doi.org/10.1016/j.autcon.2010.02.019

Fang, Y., Teizer, J., Marks, E., & Pérez-López, J. (2016). A framework for developing automated real-time tracking and monitoring system in construction projects. *Journal of Information Technology in Construction*, 21, 139-159.

Braun, A., Tuttas, S., Stilla, U., Borrmann, A. (2018). Bim-based progress monitoring. In A. BORRMANN, M. KÖNIG, C. KOCH, & J. BEETZ (Eds.), Building information modeling - technology foundations and industry practice (pp. 463–476). Springer. https://doi.org/10.1007/978-3-319-92862-3

Braun, A., Tuttas, S., Borrmann, A., & Stilla, U. (2015b). Automated progress monitoring based on photogrammetric point clouds and precedence relationship graphs. In Proceedings of the 32nd International Symposium on Automation and Robotics in Construction and Mining (pp. 274–280). https://doi.org/10.22260/ISARC2015/0034

Bui, N., Dinh, V. T., & Nguyen, L. H. (2022). Internet of Things (IoT), Building Information Modeling (BIM), and Digital Twin (DT) in Construction Industry: A Review, Bibliometric, and Network Analysis. Buildings, 12(10), 1503. doi: 10.3390/buildings12101503.

Boje, C., Guerriero, A., Kubicki, S. and Rezgui, Y. (2020): Towards a semantic construction digital twin: directions for future research. Automation in Construction 114, 103179. https://doi.org/10.1016/j.autcon.2020.103179

Bew, M. (2016): Level 2, level 3 and beyond…. In Eynon, J (ed), Construction Manager's BIM Handbook. John Wiley & Sons, 256 pp. https://doi.org/10.1002/9781119163404

Angjeliu, D., Coronelli, G., Cardani, G. (2020): Development of the simulation model for Digital Twin applications in historical masonry buildings: the integration between numerical and experimental reality. Computer & Structures. https://doi.org/10.1016/j.compstruc.2020.106282

Gerhard, D., Neges, M., Wolf, M., (2020): Transfer of digital twin concepts to the production of precast concrete parts in the construction industry, ZWF Zeitschrift fuer Wirtschaftlichen Fabrikbetrieb 115 58–61. https://doi.org/10.3389/fbuil.2022.864722

Macchi, M., Roda, I., Negri, E., Fumagalli, L. (2018): Exploring the role of digital twin for asset lifecycle management, IFAC-PapersOnLine 51 790–795. https://doi.org/10.1016/j.ifacol.2018.08.415

Yu, Y. (2022): Deep Learning Approaches for Image Classification, College of Engineering and Computer Science, Australian National University, Australia. doi.org/10.1145/3573428.3573691.

Lee, H., Grosse, R., Ranganath, R., Ng, A. (2009): Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th Annual International Conference on Machine Learning. 609–616. https://doi.org/10.1145/1553374.1553453

Krishna, M., Neelima, M., Harshali, M., Rao, M. (2018): Image classification using Deep learning. International Journal of Engineering & Technology 7(2.7):614. https://doi.org/10.14419/ijet.v7i2.7.10892

Rehman, M., Shafiq, M., Ullah, F. (2022): Automated Computer Vision-Based Construction Progress Monitoring: A Systematic Review, Buildings. 12, doi: 10.3390/buildings12071037.

Hussain, M., Bird, J., Faria, D. (2018): A Study on CNN Transfer Learning for Image Classification. In: Advances in Computational Intelligence Systems, pp. 191-202, https://doi.org/10.1007/978-3-319-97982-3_16.

O‘Shea, K., Nash, R. (2015): An Introduction to Convolutional Neural Networks. Department of Computer Science, Aberystwyth University, Ceredigion. https://doi.org/10.48550/arXiv.1511.08458

Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J. (2021): A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. Institute of Electrical and Electronics Engineers. https://doi.org/ 10.1109/TNNLS.2021.3084827

Khallaf, R., & Khallaf, M. (2021). Classification and analysis of deep learning applications in construction: A systematic literature review. Automation in Construction, 129, 103760. https://doi.org/10.1016/j.autcon.2021.103760.

Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering, 32(5), 361-378. https://doi.org/10.1111/mice.12263

Hitzler, P., Krötzsch, M., Rudolph, S., & Sure, Y. (2007). Semantic Web: Grundlagen. Springer-Verlag. http://dx.doi.org/10.1007/978-3-540-33994-6.

Robinson, I., Webber, J., & Eifrem, E. (2013). Graph Databases: New Opportunities for Connected Data. O'Reilly Media.

Angles, R., Gutierrez, C. (2008): Survey of graph database. ACM Computing Surveys, Vol. 40, Issue 1, pp 1–39, https://doi.org/10.1145/1322432.1322433

Yan, J., Wang, C., Cheng, W. (2018): A retrospective of knowledge graphs. Front. Comput. Sci. 12, 55–74. https://doi.org/10.1007/s11704-016-5228-9.

Pokorný, J. (2015). Graph databases: Their power and limitations. In K. Saeed & W. Homenda (Eds.), Computer Information Systems and Industrial Management (pp. 58-69). Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-24369-6_5.

Segaran, T., Evans, C., & Taylor, J. (2009). Programming the Semantic Web: Build Flexible Applications with Graph Data. O'Reilly Media, Inc. https://learning.oreilly.com/library/view/programming-the-semantic/9780596802141/.

Needham, M., & Hodler, A.E. (2019). Graph Algorithms: Practical Examples in Apache Spark and Neo4j. O'Reilly Media, Inc. https://learning.oreilly.com/library/view/graph-algorithms/9781492047674/.

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

El-Diraby, T. E., Lima, C., & Feis, B. (2005). Domain taxonomy for construction concepts: Toward a formal ontology for construction knowledge. Journal of Computing in Civil Engineering, 19(4), 394-406. doi: 10.1061/(ASCE)0887-3801(2005)19:4(394).

Tserng, H. P., Yin, S. Y. L., & Li, S. (2012). Developing a resource supply chain planning system for construction projects. Journal of Construction Engineering and Management, 138(3), 279-289. doi: 10.1061/(ASCE)CO.1943-7862.0000435.

Fang, Y., Park, M.-W., Peña-Mora, F., & Teizer, J. (2020). Knowledge graph for identifying hazards on construction sites: Integrating computer vision with ontology. *Automation in Construction*, 114, 103149. https://doi.org/10.1016/j.autcon.2020.103149

Peurifoy, R.L., Schexnayder, C.J., & Shapira, A. (2011). Construction Planning, Equipment, and Methods. McGraw-Hill Education.

Kavaliauskas, P., Fernandez, J. B., McGuinness, K., & Jurelionis, A. (2022). Automation of Construction Progress Monitoring by Integrating 3D Point Cloud Data with an IFC-Based BIM Model. Buildings, 12(10), 1754. https://doi.org/10.3390/buildings12101754

Jing Jia, et al. (2023). An Automatic Generation Method of Finite Element Model Based on BIM and Ontology. MDPI. Retrieved from MDPI.

Bonduel, M., Oraskari, J., Pauwels, P., Vergauwen, M., & Klein, R. (2018). The IFC to linked building data converter: current status. In Proceedings of the 6th Linked Data in Architecture and Construction Workshop (pp. 34-43). CEUR Workshop Proceedings. Retrieved from http://ceur-ws.org/Vol-2159/04paper.pdf.

Lennox, M. (2022). Simplebim: Working with structured data. AEC Magazine. Retrieved from AEC Magazine

Day, M. (2022). SimpleBIM: Taking back control with IFC. AEC Magazine. Retrieved from AEC Magazine

Fang, W., Wu, C., Liu, R., & Wang, F. (2020). Development of a Cognitive Digital Twin for Building Management and Operations. Frontiers in Built Environment, 8, 856873. https://doi.org/10.3389/fbuil.2022.856873.

Zheng, Z., Lu, Q., Zhang, Z., & Li, H. (2021). COGNIBUILD: Cognitive Digital Twin Framework for Advanced Building Management and Predictive Maintenance. Springer, Cham. https://doi.org/10.1007/978-3-031-29515-7_8.

Monteiro, A., & Martins, J. P. (2013). A survey on modeling guidelines for quantity takeoff-oriented BIM-based design. Automation in Construction, 35, 238-253. https://doi.org/10.1016/j.autcon.2013.05.005

Shen, Z. and Issa, R.R.A. (2010), "Quantitative evaluation of the BIM-assisted construction detailed cost estimates", Journal of Information Technology in Construction, Vol. 15, pp. 234-257.

Liu, H., Lu, M. and Al-Hussein, M. (2014), "BIM-based integrated framework for detailed cost estimation and schedule planning of construction projects", ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, Sydney, Australia, Vol. 31, pp. 286-294.

Lawrence, M., Pottinger, R., Staub-French, S. and Nepal, M.P. (2014), "Creating flexible mappings between Building Information Models and cost information", Automation in Construction, Vol. 45, pp. 107-118.

Niknam, M. and Karshenas, S. (2015), "Integrating distributed sources of information for construction cost estimating using Semantic Web and Semantic Web Service technologies", Automation in Construction, Elsevier B.V., Vol. 57, pp. 222-238.

Xu, S., Liu, K., Tang, L.C.M. and Li, W. (2016), "A framework for integrating syntax , semantics and pragmatics for computer-aided professional practice : with application of costing in construction industry", Computers in Industry, Elsevier B.V., Vol. 83, pp. 28-45.

Fazeli, A., Dashti, M. S., Jalaei, F., & Khanzadi, M. (2020). An integrated BIM-based approach for cost estimation in construction projects. Engineering, Construction and Architectural Management. https://doi.org/10.1108/ECAM-01-2020-0027.

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

Schlenger, J., Vilgertshofer, S., & Borrmann, A. (2023). Process-oriented Progress Monitoring of Cast-in-Place Shell Constructions Based on Computer Vision. Technical University of Munich.

Schlenger, J.; Yeung, T.; Vilgertshofer, S.; Martinez, J.; Sacks, R.; Borrmann, A. (2022). A Comprehensive Data Schema for Digital Twin Construction. Proceedings of the 29th EG-ICE International Workshop on Intelligent Computing in Engineering. doi: 10.7146/aul.455.c194.

P. Vähä, T. Heikkilä, P. Kilpeläinen, M. Järviluoma, E. Gambao, Extending automation of building construction- survey on potential sensor technologiesand robotic applications, Autom. Constr. 36 (2013) 168–178, http://dx.doi.org/10.1016/j.autcon.2013.08.002.

H. Li, G. Chan, J.K.W. Wong, M. Skitmore, Real-time locating systems applications in construction, Autom. Constr. 63 (2016) 37–47, http://dx.doi.org/10.1016/j.autcon.2015.12.001.

Braun, A., Tuttas, S., Borrman, A. and Stilla, U. (2015) A concept for automated construction progress monitoring using BIM-base geometric constraints and photogrammetric point clouds, Journal of Information Technology in Construction. Vol. 20, pp. 68-79, https://www.itcon.org/paper/2015/5.

Golparvar-Fard, M., Peña-Mora, F. and Savarese, S. (2015) Automated Progress Monitoring Using Unordered Daily Construction Photographs and IFC-Based Building Information Models, Journal of Computing in Civil Engineering. Vol. 29(1), doi: 10.1061/(ASCE)CP.1943-5487.0000205.

Wang, Z., Zhang, Q., Yang, B., Wu, T., Lei, K., Zhang, B. and Fang, T. (2021) Vision-based Framework for Automatic Progress Monitoring of Precast Walls by Using Surveillance Videos during the Construction Phase, Journal of Computing in Civil Engineering. Vol. 35, no. 1, doi: 10.1061/(ASCE)CP.1943-5487.0000933.

Vincke, S. and Vergauwen, M. (2022) Vision based metric for quality control by comparing built reality to BIM, Automation in Construction. Vol. 144, https://doi.org/10.1016/j.autcon.2022.104581.

Pfitzner, F., Schlenger, J., & Borrmann, A. (2024). Automated Productivity Evaluation of Concreting Works: The Example of Concrete Pillar Production. In J. Fottner, K. Nübel, & D. Matt (Eds.), Construction Logistics, Equipment, and Robotics (pp. 48-58). Springer, Cham. https://doi.org/10.1007/978-3-031-44021-2_6.

buildingSMART International. (n.d.). *Industry Foundation Classes (IFC)*. Retrieved from buildingsmart.org

Datacubist. (n.d.). SimpleBIM. Retrieved from simplebim.com

Pauwels, P., & Roxin, A. (2016). IFCtoLBD: A Building Data Converter for Linked Building Data. Proceedings of the LDAC Workshop, 2016. Retrieved from https://linkedbuildingdata.net/IFCtoLBD.

RDFLib Documentation (2023). Retrieved from https://rdflib.readthedocs.io/.

IfcOpenShell Documentation (2023). Retrieved from https://ifcopenshell.org/.

Bilal, M., Oyedele, L. O., Qadir, J., Munir, K., Ajayi, S. O., Akinade, O. O., ... & Pasha, M. (2016). Big data in the construction industry: A review of present status, opportunities, and future trends. *Advanced Engineering Informatics, 30*(3), 500-521. https://doi.org/10.1016/j.aei.2016.07.001.

Bock, T., & Linner, T. (2016). Construction Robots: Elementary Technologies and Single-Task Construction Robots. Cambridge University Press. https://doi.org/10.1017/CBO9781139923934.

Moselhi, O., & Khan, Z. (2010). Analysis of Labour Productivity of Formwork Operations in Building Construction. Construction Innovation, 10(3), 286-303. DOI: 10.1108/14714171011060088.

Câmara Brasileira da Indústria da Construção (CBIC). (2017). Manual Básico de Indicadores de Produtividade na Construção Civil, Volume 1 - Relatório Completo. Brasília, DF: CBIC.

RSMeans. (2020). RSMeans Building Construction Cost Data 2020 (78th ed.). Rockland, MA: RSMeans.

Deutsches Institut für Normung e.V. (DIN). (2012). Beton, Stahlbeton und Spannbeton - Teil 3: Bauausführung (DIN 1045-3):