Technical University of Munich

School of Engineering and Design

Chair of Computational Modeling and Simulation

An automated framework for managing design changes in BIM models

Master thesis

for the Master of Science Course Civil Engineering

Autor:	Zaid Memon
Matriculation Number:	[redacted]
Supervisors:	M. Sc Mansour Mehranfar
	M. Sc Jiabin Wu
Examiner:	Prof. Dr.-Ing. André Borrmann

Date of issue:	15. November 2023
Submission date:	15. May 2024

Abstract

BIM is an innovative technology in the AEC industry, providing a digital representation of building projects. Yet, the ever-changing environment of construction projects requires efficient methods for detecting changes within the BIM system. Detecting changes requires the identification and assessment of modifications that take place during the project's progression, guaranteeing precision, uniformity, and effective teamwork among project participants. Identifying changes in BIM models is essential for preserving project integrity, minimizing mistakes, and automating the process saves time, thereby enhancing project timelines. Conventional methods for identifying changes, like visual inspections or manual surveys, are frequently time-consuming and susceptible to human error. Consequently, there is an increasing need for automated change detection techniques that improve the process and boost efficiency in the AEC industry. This paves the way for delving into the importance of change detection in BIM and emphasizes the necessity for creative frameworks to tackle this issue. This thesis aims to create a framework for automated BIM modelling with family creation along with a BIM-based framework for indoor building comparison analysis. The inputs are the BIM Models, which are to be analysed, and the outcome of the study is the detailed representation of the element changes between the models. The proposed method has been tested on different datasets, and the results depict the effectiveness of the proposed method in detecting model changes.

Zusammenfassung

BIM ist eine innovative Technologie in der AEC-Branche, die eine digitale Darstellung von Bauprojekten ermöglicht. Die sich ständig verändernde Umgebung von Bauprojekten erfordert jedoch effiziente Methoden zur Erkennung von Änderungen innerhalb des BIM-Systems. Die Erkennung von Änderungen erfordert die Identifizierung und Bewertung von Modifikationen, die während des Projektverlaufs stattfinden, um Präzision, Einheitlichkeit und effektive Teamarbeit zwischen den Projektbeteiligten zu gewährleisten. Die Identifizierung von Änderungen in BIM-Modellen ist von entscheidender Bedeutung für die Wahrung der Projektintegrität, die Minimierung von Fehlern und die Automatisierung des Prozesses spart Zeit, wodurch die Projektlaufzeiten verkürzt werden. Herkömmliche Methoden zur Erkennung von Änderungen, wie visuelle Inspektionen oder manuelle Überprüfungen, sind häufig zeitaufwändig und anfällig für menschliche Fehler. Daher besteht ein zunehmender Bedarf an automatisierten Verfahren zur Änderungserkennung, die den Prozess verbessern und die Effizienz in der AEC-Branche steigern. Dies ebnet den Weg, um die Bedeutung der Änderungserkennung in BIM zu erforschen und unterstreicht die Notwendigkeit kreativer Rahmenbedingungen, um dieses Problem anzugehen. Ziel dieser Arbeit ist es, einen Rahmen für die automatisierte BIM-Modellierung mit Familienbildung sowie einen BIM-basierten Rahmen für die Analyse von Gebäudevergleichen in Innenräumen zu schaffen. Die Eingaben sind die BIM-Modelle, die analysiert werden sollen, und das Ergebnis der Studie ist die detaillierte Darstellung der Elementänderungen zwischen den Modellen. Die vorgeschlagene Methode wurde an verschiedenen Datensätzen getestet, und die Ergebnisse zeigen die Wirksamkeit der vorgeschlagenen Methode bei der Erkennung von Modelländerungen.

Contents

List of Figures

List of Tables

List of Abbreviations

BIM	Building Information Modeling
FM	Facility Management
AEC	Architecture, Engineering, and Construction
LOD	Level of Development
LOI	Level of Information
API	Application Programming Interface
CAD	Computer-Aided Design
LiDAR	Light Detection and Ranging
LOG	Level of Geometry
MEP	Mechanical, Electrical, and Plumbing
HR	High Resolution
EIR	Employer Information Requirement
AM	Asset Management
VPL	Visual Programming Language
UI	User Interface
TLS	Terrestrial Laser Scanner
VPP	Visual Programming Platform
CSV	Comma Separated Values

1 Introduction

1.1 Overview

Building Information Modeling (BIM) has emerged as a transformative technology in the architecture, engineering, and construction (AEC) industry, providing a comprehensive digital representation of a building or infrastructure project [1]. BIM has existed in the AEC industry for more than 20 years now aiming to make the industry much more efficient and streamlined [2]. Beyond its original function as a tool for three-dimensional design and component use, the use of BIM applications has grown considerably. It's evolved into a versatile tool used for model analysis, clash identification, product selection, and thorough project conceptualization, among other things. BIM is becoming more and more popular as an emerging technology for planning, creating, and managing buildings in different nations. It turns out to be a powerful tool that lets users create virtual prototypes and visual representations of projects before they are really built. However, because of the complex processes involved, BIM requires specific expertise [3].

The requirement for change detection in the BIM ecosystem is found in the dynamic nature of construction projects, where design iterations and revisions are frequent. A study highlighted that BIM is a collaborative platform that allows stakeholders to contribute to and alter the digital representation of the building throughout its entire cycle. Change detection technologies are critical in spotting and assessing these changes, ensuring that project teams are working with the most up-to-date and accurate information. This capacity is especially important in eliminating clashes and conflicts among project components, disciplines, or stages, allowing for a more efficient and error-free building process [4].

Additionally, the BIM ecosystem indicates that the necessity for change detection goes beyond the construction phase. BIM models that have been enhanced with data on building components and systems have proven to be valuable resources for facility management (FM) and operations. Identifying changes in the BIM model after construction helps facility managers plan maintenance, restorations, and future expansions more efficiently. Furthermore, regulatory compliance is becoming increasingly entangled with BIM standards, while efficient change detection guarantees that projects conform to growing industry norms and guidelines, strengthening the built environment's long-term sustainability and flexibility [5].

The construction sector lacks detection of the faults and their intensity found on the building site which leads to a considerable number of financial losses. Therefore, there is a dire need to build a system where errors can be detected and visualized in a comprehensive way. Creating BIM Models is considered an initial step for model preparation. Once the models are prepared, the proposed framework can detect changes in both models in terms of location and parameter changes like thickness, width, etc, in short, both semantic and geometric changes of elements.

1.2 Objectives

The goal of the thesis is to design a framework using a BIM authoring tool to carry out comparative analysis for indoor building environments to ease decision-making regarding the changes between the 'as designed' (planned) and 'as built' (constructed) phase of the project. The thesis aims to solve the following questions:

- What are the minimum requirements for data to carry out comparative analysis of Building Elements?

- How to carry out a comparative analysis between different BIM models at different stages and formulate the results in a comprehensive version?

- How does the proposed framework for change detection contribute to the current BIM industry?

1.3 Structure of the thesis

The thesis is structured as follows:

- Chapter 2 provides information regarding Literature Review, engaging in depth in the fields of BIM, BIM based modeling, comparative analysis, and basis for related technical aspects. This chapter aims to construct a solid theoretical background for the thesis.

- Chapter 3 presents the proposed methodology, explaining the workflow of the proposed approach for comparative analysis, detailing each step to reach the milestone.

- Chapter 4 focuses on the detailed implementation of the steps mentioned in the methodology, including the prerequisites required to perform the analysis, as well as an explanation of the scripts and the idea behind creating them. It also includes the result part, which showcases testing the framework on the models provided by TUM in the form of 2 Datasets (single storey) and a multi-storey model and formulating the output, including both quantitative results and visual interpretation.

- Chapter 5 concludes the thesis by engaging in a comprehensive discussion of the thesis findings and defending the questions proposed at the initial stage of the thesis. Additionally, a few points have been portrayed which could lead to the betterment of the research in the future.

2 Literature Review

2.1 Building Information Modeling

Digitalization has increased production and quality in a variety of industries over the previous decade. Digital tools for planning and controlling structures are increasingly utilized in AEC. However, the uniform application of digital information across the process lags, resulting in information loss, particularly in the form of drawings, and disturbing the flow of built facilities throughout their existence. Relying solely on drawings to manage information limits the possibilities of information technology in project management and construction operations. The human verification necessary for the consistency of distinct technical drawings, which are frequently generated by experts from various disciplines, is a considerable difficulty. If design modifications are not continuously recorded throughout plans, they can result in undiscovered discrepancies, generating additional expenditures during construction. The inadequate information depth of technical drawings also prevents direct usage in downstream applications, mandating manual data re-entry, which contributes to errors and inefficiencies within construction and building management. Extraction of post-construction data gathered from drawings is time-consuming and results in digital loss of data at each exchange step [6].

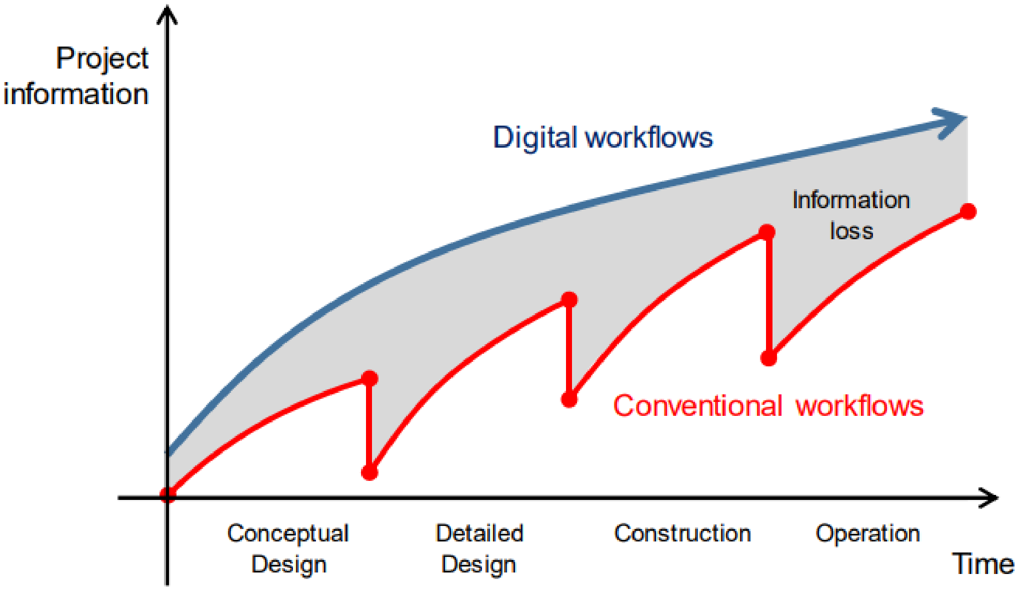


Figure 1 – Loss of information caused by disruptions in the digital information flow [7].

The role of Building Information Modeling then comes into play. The US National Building Information Modeling Standard defines BIM as follows: "*Building Information Modeling (BIM) is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life cycle, defined as existing from earliest conception to demolition. A basic premise of BIM is a collaboration by different stakeholders at different phases of the life cycle of a facility to insert, extract, update or modify information in the BIM to support and reflect the roles of that stakeholder*" [8].

Construction faults are common and contribute to building underperformance. The average dwelling has 2.29 to 28.3 faults, with rectification expenses ranging from 3.23% to 23% of the project budget. Incorrect installation (24%–40%) and missing items (20%–55%) are the most common issues. These are usually found after handover. External walls, partitions, doors, windows, floors, and roofs often have defects. Widespread gaps in building fabric and insufficient insulation installation decrease thermal performance. Common issues include thermal bridging, air leakage, vapor, and air barrier defects. Thermal performance declines due to sealing failure, moisture retention, and the short lifespan of low-quality construction and insulating materials. Building errors can result from design, manufacture, or installation problems. Most building irregularities are caused by design flaws, 30% to 60%. Missing elements during design, production, or construction are another factor. Although rare, building damage can cause flaws. Some contractors tend to take shortcuts or continue with their usual practices instead of adhering to design specifications [9].

BIM provides significant benefits in the AEC business, fundamentally changing project processes. BIM functions as a collaborative platform, improving communication and coordination among project stakeholders throughout the project life cycle [7]. The technology's clash detection capabilities help to reduce construction errors by identifying and solving conflicts early in the design process [10]. This results in more efficient building procedures and better project outcomes. Moreover, parametric modeling in BIM facilitates successful design change management by offering a dynamic representation that can be easily and quickly modified to reflect changes [11]. By ensuring that modifications are smoothly incorporated into the model, this feature lowers the possibility of inconsistent design and improves overall design efficiency. BIM has been linked to time and cost reductions in construction projects in addition to these benefits. BIM helps to improve project efficiency and financial results by reducing errors, preventing rework, and optimizing project timetables [12]. To sum up; BIM transforms the AEC sector by enhancing teamwork, reducing mistakes via clash detection, enabling seamless design modifications, and assisting with time and money savings in building projects.

2.2 Level of Development (LOD)

In the past few years, BIM has become increasingly popular, and many software companies have made a wide range of programs with strong BIM features. BIM started out as a field of study for academics, but it has since grown into a well-established business-standard. During the planning and construction phases of a project, the building model is constantly being improved, going from a concept design to an "as-built" model. The idea regarding LOD has gotten a lot of support in the AEC field to document how dynamic the building design process is.

LOD is an important standard tool that indicates what kind of numerical data (Level of Information, or LOI) and physical information (Level of Geometry, or LOG) you need at different points in a project. This idea not only describes the information's breadth but also its age and dependability. Because building projects naturally move forward, LOD and some design steps go hand in hand. LOD ensures consistency and openness as the architectural model grows, making it easier for project partners to talk to each other [6].

Basically, the development of BIM from its academic roots to industry standards can be summed up by the addition of powerful software tools and the legalization of the LOD concept. This framework is important for getting through the complicated and iterative process of building design because it has both physical and informational parts. It also helps projects go better in the AEC business by making it easier for people to work together and reducing uncertainty. There are six common LODs set by the US-American BIMForum [13]. They are 100, 200, 300, 350, 400, 200. While LOD 100 and LOD 200 show the most basic level of detail with rough geometric representations of building elements, LOD 300 and above models must be used to evaluate design changes because they contain accurate geometry and precise dimensions that are needed for the framework to work accurately.

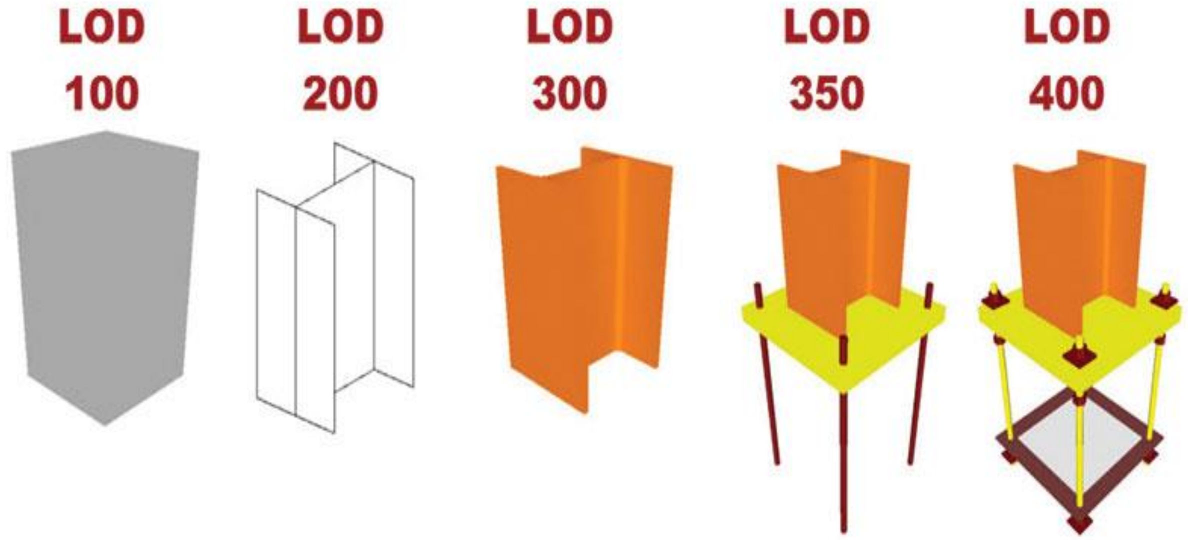


Figure 2 – Characterization of Level of Development (LOD) [14].

2.3 Digital BIM Models

In the realm of BIM, a "digital model" pertains to a virtual depiction of a building or infrastructure project. This digital model covers different aspects of the project, such as its geometry, spatial relationships, materials, components, and associated metadata. The digital model acts as a centralized and collaborative platform for stakeholders to design, visualize, analyze, and manage the project throughout its lifecycle.

The digital model exists primarily in two states: “as designed” and “as built”. The difference between "as-designed" and "as-built" not only relates to the time period between the completion of modeling and construction drawings but before the occupants move in and start using the building. In simple terms, the differences between "as-designed" and "as-built" arise from the stages where design choices are implemented in practical terms, such as procurement, construction, and commissioning [9].

2.3.1 As Designed Model

The as-designed model is a complete representation of the initial design intent for a building or infrastructure project. It is usually created during the early stages of the project lifecycle, specifically in the design phase. This digital model provides a comprehensive overview of the building's geometry, architectural elements, structural systems, MEP (Mechanical, Electrical, Plumbing) systems, and other relevant components. Collaboration among stakeholders, such as architects, engineers, designers, and clients, takes place around this model, enabling the development and improvement of the project's design concept. In addition, the model produced serves as an essential reference tool. It allows stakeholders to visualize the project's proposed appearance, simulate its performance, conduct a thorough analysis, and make well-informed decisions prior to starting construction.

2.3.2 As Built Model

The as-built model offers a precise digital depiction of the actual state of a building or infrastructure project once construction is finished. It is created by capturing and integrating data collected throughout the construction process, including any changes, modifications, or variations from the original design. This model accurately represents real-world conditions, taking into account variations in dimensions, material quantities, installed components, and spatial relationships. Advanced technologies like laser scanning, photogrammetry, and reality capture are widely used to ensure the as-built model's accuracy and completeness. Once established, this model becomes a crucial asset for facility management, operations, and maintenance tasks once the project is transferred to the owner or operator. Through a thorough comparison of the as-built model and the as-designed model, stakeholders can identify and analyze any discrepancies, verify the quality of construction, evaluate adherence to design specifications, and utilize the knowledge gained to enhance future design iterations or renovation projects [34].

2.4 BIM Comparative Analysis

The process of finding and judging changes or differences that happen in a BIM project over time is called BIM comparative analysis. It involves looking at different versions of a BIM model or piece of data to find and show any changes, updates, or problems. Finding changes in BIM is important for keeping things accurate, making sure everything is the same, and encouraging everyone to work together well throughout the lifecycle of a building project. Architectural, structural, and MEP (Mechanical, Electrical, and Plumbing) parts from the BIM model may be used in this process. In the past, progress was checked by either building workers looking at it visually or professional surveys using total stations and leveling tools, which took a lot of time and money. With 3D laser scanning, you can record "as-built" information quickly and accurately on-site using a number of different point cloud modeling methods. There is a big problem with this method, though: it needs purposeful study of 3D point clouds, computerized help with interpretation, and ways to look at metric tolerances in engineering [15].

2.4.1 Important Components of BIM Comparative Analysis

During the design phase, it's crucial to track any alterations made to the plan, structure, or components of a building. This ensures architects and designers stay informed about design iterations, facilitating a streamlined process. Construction updates are essential for monitoring changes implemented during the construction phase, whether additions or eliminations of elements, ensuring adherence to the original plans. Clash detection is vital for identifying conflicts or clashes that may arise when different components intersect, aiding in conflict resolution. Progress tracking involves monitoring construction progress by comparing BIM models at various stages, enabling stakeholders to assess adherence to the project's timeline effectively. Lastly, facility management changes involve identifying updates or alterations in the BIM model related to facility management, such as machinery changes, maintenance requirements, or operational adjustments, ensuring efficient facility management post-construction.

2.4.2 Need for BIM Comparative Analysis

When compared with examples of real projects, the need for comparative analysis is highlighted. For instance, at the Art Museum in Denver, prior to construction, a total of over 1200 members interference were reviewed using BIM in the assembly of steel members. Without a BIM tool, it would have been very difficult to review the clashes in such a complex structure. The usage of BIM not only streamlined the construction process but also saved around $0.4 million. A similar project at the V6 Engine Plant of General Motors used BIM to review 3000 to 4000 interferences and rectification, which not only saved around 3-5% of overall expenditures but also helped in completing the project five weeks quicker than projected [16].

Similar to projects like the Trenue Tower in South Korea, the projects used BIM to check designs and found over 40 drawing flaws and 30 conflicts. A similar kind of project, the Vietnam Plant of Samsung Engineering, where roughly 470 conflicts were detected in different building components. This not only helped in avoiding the necessity for reconstruction but also saved millions of dollars [16]. There exists such kind of multiple examples that clearly state the importance of BIM comparative analysis.

2.5 BIM and Facility Management

Facility Management's (FM) primary goals are to continuously demonstrate value and improve productivity. As defined by ISO 41013, a facility is "a grouping of assets constructed, installed, or established to fulfill the requirements of an entity, whether it be individuals or an organization." These resources include utilities, real land, infrastructure, machinery and systems, and specialized services [17].

A critical and essential component of any project combining FM with BIM is change management. Two approaches might be used to support these changes: a "general step-by-step" approach or the execution of a "niche project with follow-up." In the first, a specific application area inside the company is defined in order to present and verify the benefits and results of the suggested modification. This method approaches the transition-transformation cycle as if it were a short-term experiment. Implementation timescales are often shortened, and enthusiastic staff are sought to support the specialty project. The aim is to ascertain concrete outcomes from the project's implementation and record insights gained to enable the initiative's ensuing rollout across the entire organization. The latter strategy entails comprehensive planning and company-wide process reengineering. Here, seminars, well-run practical learning sessions, and the definition of "as is" and "to be" engineering process models are essential. Process reengineering techniques are closely associated with training, which directs and supervises the entire implementation process step-by-step. The second strategy, which focuses on the 'niche project with suggestions' change management strategy, is examined in the remaining sections of this paper. It is less common because of the difficulties with tackling numerous ambiguities in Employer Information Requirements (EIRs) and their corresponding workflows, as well as the difficulties in upfront planning [18].

When implementing any BIM endeavor, change is a necessary component. Change management is essential when owners switch from traditional methods, such as asset management (AM) and/or FM, to BIM-supported methods because of the associated organizational and technological changes. On the other hand, little is known about the dynamics of change or how best to manage it in this situation.

2.6 Automated BIM Modeling

Digital BIM models are the basis of the method; they keep all the project information in one place and make it easier for everyone to work together. Automation, a word usually linked to how well robots work, is being used more and more for digital jobs in the BIM framework. Adding artificial intelligence to this idea has made it even better by letting different BIM jobs be done automatically. Even though setting up AI software may take a long time, the time saved is well worth it. AI can quickly handle boring and repeated tasks, giving BIM experts more time to work on more important tasks and making the project more efficient overall [35]. When BIM technology is added to these models, it makes new ideas possible because it speeds up the process, makes it more accurate, and lets people from different fields work together. It means that when a builder or designer changes geometry or a building element in one view, like a plan, section, or elevation, the BIM tool instantly adjusts and syncs the change with all other views. It is possible to work with building parts, make conceptual models, make objects and geometries, and use computational and parametric design in 3D BIM. With the three-dimensional view, the people on the project team may be able to work together better and model and fix structural problems that affect everyone. The BIM Model stores all the important data and information, which makes it easier to solve problems in the future [19]. To sum up, 3D BIM creates a solid foundation for dealing with future issues that may arise in the planning and building process. It does this by improving modeling speed and accuracy and promoting group problem-solving.

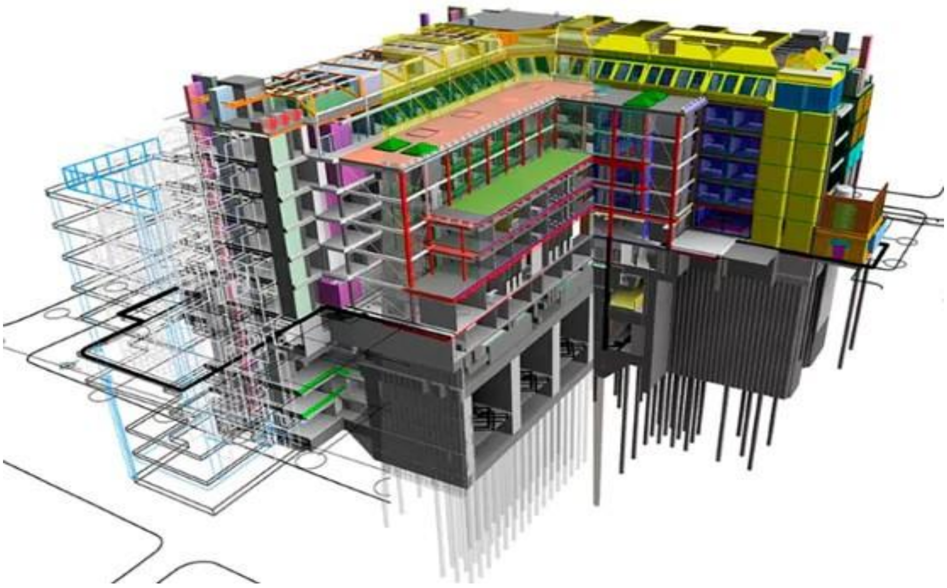


Figure 3 – Visualization of BIM 3D Model [20].

Creating a BIM model manually involves placing and detailing each element of the building model within BIM software. This approach empowers practitioners with extensive control over all aspects of the building's design, enabling the creation of personalized solutions that meet specific project needs. Yet, it can be quite time-consuming to model everything manually, especially for intricate or extensive projects, and the manual input of data raises the risk of human mistakes. Professionals must have a strong understanding of both the BIM software and architectural or engineering principles to develop precise and dependable models.

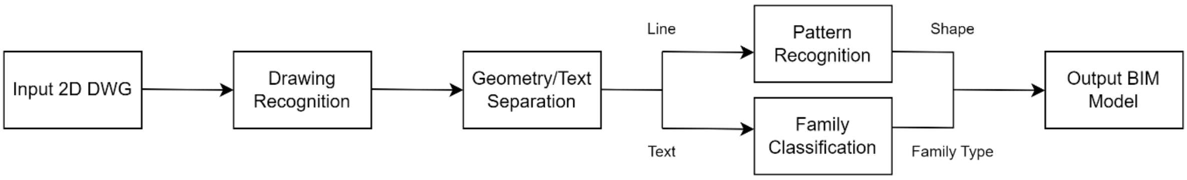


Figure 4 – General workflow of Manual BIM Modeling.

On the contrary, automated BIM modeling uses VPL to establish a series of rules and parameters that can generate or adjust the building model automatically. This method is very effective, enabling the quick creation of models and simplifying updates or modifications throughout the entire model. It guarantees a uniform implementation of design guidelines and criteria. After a script is created, it can be repurposed or modified for different projects, boosting efficiency for recurring tasks and intricate data handling. Although there is a learning curve associated with VPLs, they excel in handling intricate geometry and data-driven design processes, proving to be a valuable asset in contemporary BIM workflows.

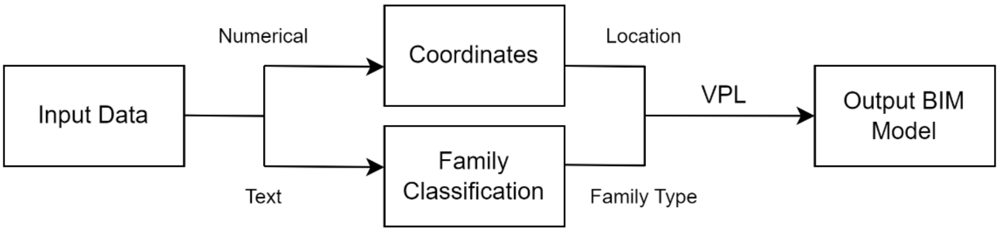


Figure 5 – Workflow of Automated BIM Modeling proposed in this thesis.

Practically, a hybrid technique can be utilized. Automated processes manage complex or repetitive parts of the model, while manual work is dedicated to unique, intricate, or project-specific elements. This collaborative method combines the efficiency of automated modeling with the precision and customization of manual modeling. This leads to an enhanced, precise, and top-notch BIM process that is flexible to various project requirements.

2.6.1 Benefits of BIM Models in the AEC domain

Below mentioned are the potential benefits of using digital BIM Models in AEC:

1. Improved Accuracy: 3D modeling services guarantee extremely precise and ideal data for building projects, removing the need for rework, and offering accurate information for high-quality construction sites [20].

2. Reduced Risks and Costs: Large budgets are frequently involved in construction projects. 3D BIM modeling gives precise project cost estimates and enables comprehensive planning ahead of time. Conflict avoidance also reduces delays and rework, which saves money and time [20].

3. Pre-Construction Visualization: It is advantageous for stakeholders to see the project before actual work starts. An exact depiction is provided by the 3D BIM model, enabling any necessary early corrections and revisions [20].

4. Communication: 3D BIM modeling allows each team member to view the project, promoting communication and discussion, in contrast to traditional construction approaches where employees work in isolation. This cooperative strategy guarantees lucidity, diminishes misunderstandings, and minimizes conflicts [20].

5. Improved Efficiency: By increasing productivity and efficiency and lowering waste and costs, 3D modeling services optimize construction projects. Time is saved, and important building deadlines are met when paper drawings are eliminated, and the use of virtual models increases. It also guarantees a safe and regulated work environment and encourages productive teamwork and communication [20].

2.6.2 Parametric BIM Modeling

Algorithms in a parametric digital model ensure that changes automatically update related information based on established parameters. Parametric design is created based on the values of parameters, with equations being employed to define the connections between the different shapes [36]. Parametric modeling involves incorporating constraints and variable parameters [37]. Conceptual parametric design is a powerful tool for designers and architects as it allows for real-time modification and exploration with no effort from the user [36]. Parametric design prioritizes the design process over the finished product, in contrast to traditional CAD design. It enables the automatic derivation and description of multiple design variations. Currently, parametric techniques are being employed in various additional domains, including infrastructure development, modular building, bionic construction, and lightweight construction. Architects, when engaging in parametrization, assume the role of developers who create information chains to describe relationships. Parametric BIM modeling automates updates to design components, reduces human error, and accelerates the updating process as framework circumstances change, as opposed to manual alterations. Drawings generated from the BIM model provide rapid layout regeneration following each modification. Parametric models, once created, can be enhanced for future projects, enabling planners and engineers to automate repetitive procedures. Even individuals with minimal programming skills can generate parametric models using visual scripting tools [21]. Implementing parametric modeling concepts to self-elevating maritime platforms, they created full-scale constrained models, which significantly decreased the burden on design personnel [38]. When parametric design methods were implemented for self-erecting platforms, they generated the main body of the platform as well as its numerous component sections in a brief period of time. This made it easier to make alterations to the model and significantly sped up the design process [39]. The aforementioned works provide a strong theoretical base for the parametric design of BIM models.

Parametric BIM Modeling is basically divided into three parts:

- Object-based Parametric Modeling.
  (Related software: AutoCAD Architecture, Nemetschek Allplan, Graphisoft ArchiCAD)
- Geometry-based Parametric Modeling.
  (Related software: Autodesk Revit, Graphisoft ArchiCAD, TEKLA Structure)
- Parametric BIM Modeling.
  (Related software: Siemens NX, Autodesk Inventor, Generative Components, Grasshopper)

2.7 Visual Programming Languages (VPLs)

Visual programming is a type of coding language that allows users to graphically depict operations. A visual programming language, as opposed to standard text-based programming languages, allows developers to express processes in human-readable language. The capabilities of visual programming technology in use impact the amount of the transition from visual to traditional programming [23].

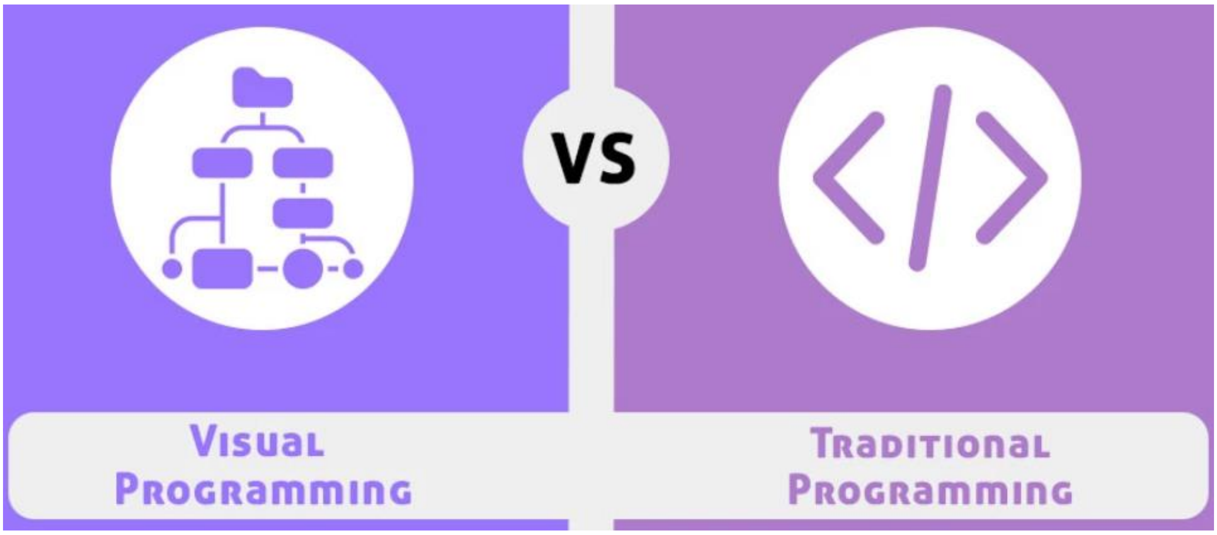


Figure 6 – Scripting method comparison between traditional and visual programming [22].

Visual development and conventional development are two unique techniques for creating software. Traditionally, code is written in a programming language that relies on text, such as C++, Java, or Python. This method needs a thorough understanding of the language, syntax, and core programming principles such as data structures and algorithms. Visual development, on the other hand, uses visual components such as flowcharts and diagrams to represent programming processes, with environments frequently incorporating drag-and-drop interfaces. The instrument and its intended users influence the decision between visual and traditional development. It can boost professional developers' productivity or allow people with no programming experience to create software. Overall, the fundamental difference is that traditional development focuses on actual coding, whereas visual development focuses on the design and overall organization of the software [24].

2.7.1 Consolidating Visual Programming with BIM

Visual programming in parametric design aids architecture and engineering. First, it automates design chores like creating comprehensive construction documents or building component schedules. This automation saves engineers and architects time and effort, allowing them to focus on innovative design initiatives. Visual programming allows professionals to generate bespoke models for each project's needs. Parametric design allows architects to create proprietary algorithms that generate façade designs based on inputs. Customization improves design adaptability and responsiveness to project needs [25].

Visual programming allows for faster construction of complicated and realistic models than traditional methods. Parametric models that account for window size, insulation thickness, and HVAC system performance can help designers assess how design choices affect energy usage. This speeds design exploration and informs decision-making. Visual programming thereby enhances design workflows, creativity, and efficiency in architecture and engineering [25].

Table 1 – Visual scripting tools with corresponding language [26,27,28,29].

Name	Complying Software	Company	Programming language
Dynamo	Revit	Autodesk	C#, Python
Allplan Visual Scripting	Allplan	Nemetschek	Python
Grasshopper	Rhinoceros	Open Source	C#, Python, C++
PyFlow	Standalone	Open Source	Python

Dynamo is a powerful visual scripting plugin for Autodesk Revit that allows users to use the Revit API (Application Programming Interface) at an unparalleled level. It connects with Revit effortlessly, boosting computational building information modeling (BIM) interactively, although it should be emphasized that it is not a standalone product. The fact that Dynamo for Revit does not require prior programming skills is a remarkable feature. Users can enter lines of code and use nodes to design algorithms. Revit Dynamo substantially improves project design processes and operations with an intuitive interface and a selection of pre-made scripting libraries [34].

2.8 Current Research for change detection using point clouds

The suggested method provides a thorough way to compare a Building Information Model (BIM) with a point cloud that displays the inside of a building. There are two main steps in the process: Point Classification and Registration and Comparison. The goal of both is to find and show differences between the computer model and the real building [31].

The point cloud is broken down into "existing" (0) and "new" (1) points in the Point Classification step. Existing points match areas in the BIM, and new points show parts of the building's interior that aren't in the BIM. The ranking is based on the point-surface distance, which is the straight-line distance between each point and the BIM surface that is closest to it. It is marked as existing (0) if the distance is less than a certain level and the point's orthogonal projection falls within the surface boundary. If not, it is marked as new (1). The noise in the point cloud and the mistake in registration between the point cloud and the BIM are used to set the threshold [31].

The BIM and point cloud are aligned in a shared coordinate system in the Registration and Comparison step. This makes it easier to compare them in more detail. This process helps find differences between the BIM and the real interior of the building by pointing out places where the model doesn't match up with reality. The point classification method is very important in this step because it lets you see that new elements in the point cloud are made up of groups of points labeled as Type 1, while old elements in the BIM only have points of Type 0. Figuring out the point-to-surface distance not only helps with labeling points, but it also tells you how far away each point in the point cloud is from its related element in the BIM [31].

Figure 8 shows the results of the fake dataset that show how well the point classification based on point-surface distance works. In the point cloud, groups of points that are far apart show that the building has changed a lot since the BIM. Some points connected to new elements are very far away (up to 2.5m), while others are very close to the BIM surfaces they are connected to.

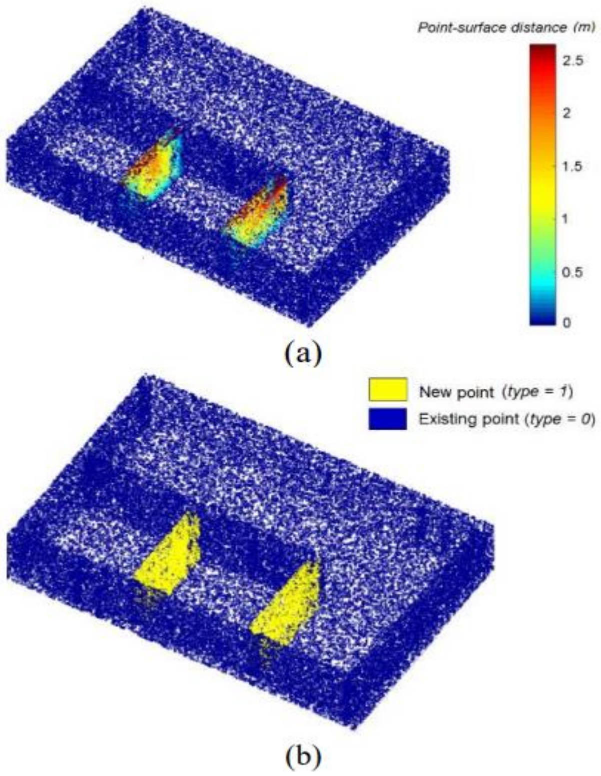
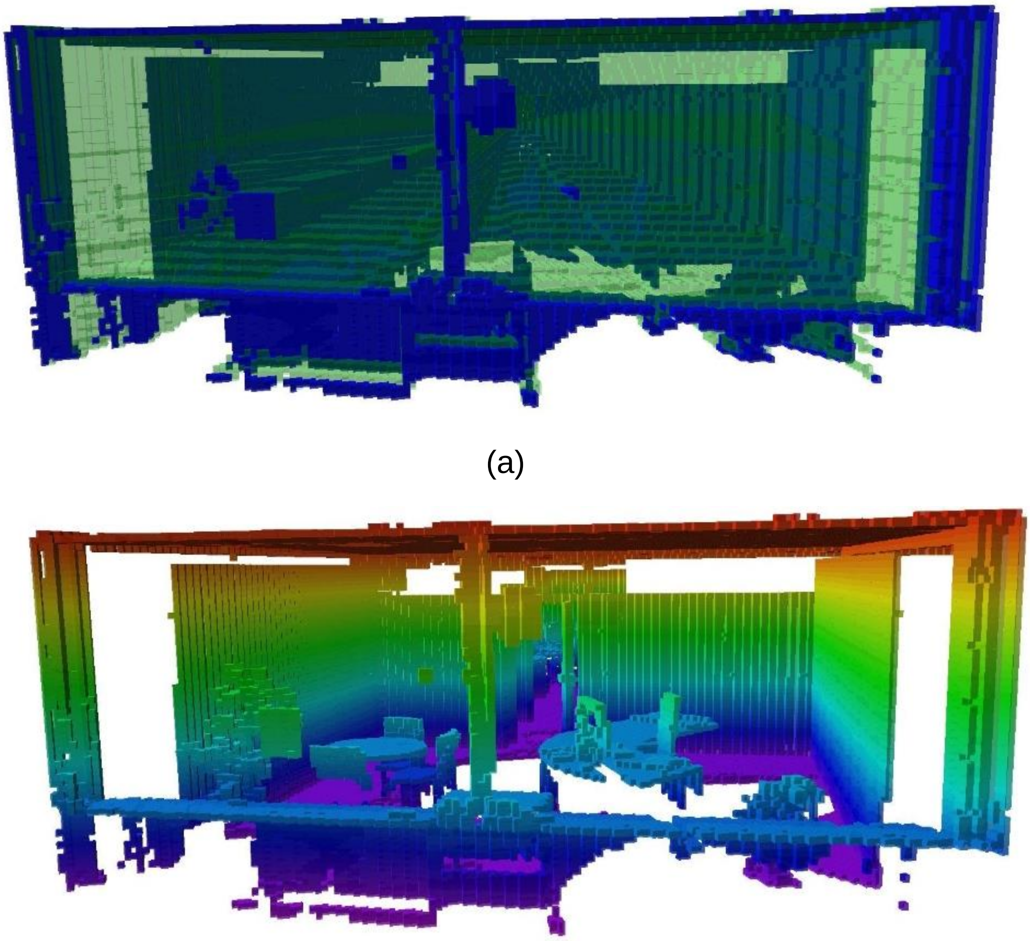


Figure 7 - Comparison results of the dataset: (a) the point cloud colorized according to point-surface distances; (b) the point cloud with the result of point classification [31].

Then, putting points into two groups: existing (blue) and new (yellow) makes the differences even more clear. Existing points show parts that are already in the BIM, while new points show the presence of new walls that were not in the BIM at first. This way of classifying really brings out the differences between the fake dataset and the BIM, especially in places where new parts of the building's interior have been added [31].

In conclusion, the suggested method offers a structured and thorough way to compare BIM and point cloud data, which lets you find and see differences between how a building looks digitally and how it actually is. This helps make sure that building models are correct and improves them, which makes architectural, and construction plans more accurate.

Another approach for indoor comparative analysis is using the TLS. During the initial phase of automatic indoor change detection, the researchers utilized an indoor terrestrial laser scanner (TLS) point cloud. This point cloud consisted of 31 individual scans that were organized in an octree structure. By leveraging the scanner's positions, a ray-tracing algorithm was employed to differentiate between three states within the octree cells. Cells containing points were classified as occupied, while cells with at least one ray passing through were labeled as free. The state of the remaining cells was considered unknown. This structure allows for the examination of the model's alignment with a previous version and the analysis of scenarios where additional changes occur, and their general locations are identified. In order to monitor these changes, a cost-effective device equipped with a range sensor, like Google Tango, could be utilized. The octree structure enables convenient updates to the relevant model sections by modifying the state of the corresponding voxels as required [32].



(a)

(b)

Figure 8 - Octovis visualization of occupancy map: a) green: free, blue: occupied, transparent: unknown octree volumes; b) color-coded height map [32].

2.9 Research Gap

An efficient BIM-based framework is needed to determine changes between ‘as designed’ and ‘as built’ BIM Models with respect to geometric and semantic data. Many existing research focus primarily on detecting changes during the construction phase, however, there still lies a gap along the approach for long-term monitoring and maintenance of buildings which can enhance the overall project lifecycle management. There exist few approaches for comparing BIM Models, but the industry requires automated, quick and robust frameworks and more sophisticated algorithms that can effectively identify and analyze changes in a more dynamic way. Few methods have been discussed in section 2.8, however, the drawback with the present methods is that it doesn't include semantic information about the changes that have been found, like clutter, building elements, utilities, and so on. There exist some approaches where VPL and BIM authoring tools are used to access model changes, however, these approaches are semi-automatic, using VPL only to extract data from models and analyzing the data in spreadsheets. Moreover, these approaches are only capable of highlighting change classification and not capable of detecting any new elements or detecting elements deleted. The framework presented in this thesis is automatic and capable of detecting any new or deleted elements thereby filling the aforementioned gaps.

Visualization is crucial for effective communication and decision-making in massive construction projects, however, the construction industry encounters considerable communication obstacles, such as a lack of enthusiasm for proactive communication due to conflicting interests and a tendency to prioritize personal gains over information sharing. In addition, the industry's hierarchical management structure leads to prolonged and distorted message transmission, while inconsistent information standards impede effective communication among the many professionals and firms involved in projects. Also, the industry's historical dependence on labor-intensive practices and limited information integration has contributed to the slow adoption of advanced communication technologies, which further intensifies these challenges [33]. To overcome these challenges, it is important to take proactive steps such as encouraging effective communication, simplifying management systems, implementing uniform information standards, and encouraging the use of cutting-edge technologies to improve coordination and avoid conflicts, and enhance the overall quality of construction projects. To solve the problem, the thesis proposes intuitive and interactive visualization techniques like 3D overlays or color-coded Models helping stakeholders to comprehend and respond to alterations during project evolution.

3 Methodology

3.1 Proposed Method

To solve the above discussed problem, a framework to detect model changes is proposed as shown in Figure 10. Step 1 is to prepare 3D BIM Models which are carried out using BIM Modeling Software. Step 2 is to compare BIM Models which are carried out using Visual Programming Platform (VPP). In order to make the run time quicker and to carry out comparison as per the element of user's choice, individual scripts for each element have been prepared. Step 3 involves result interpretation, which includes color-coded grading using VPL and visualization of result in a BIM tool, as well as detailed, easily readable numerical result exported into a database or a spreadsheet.

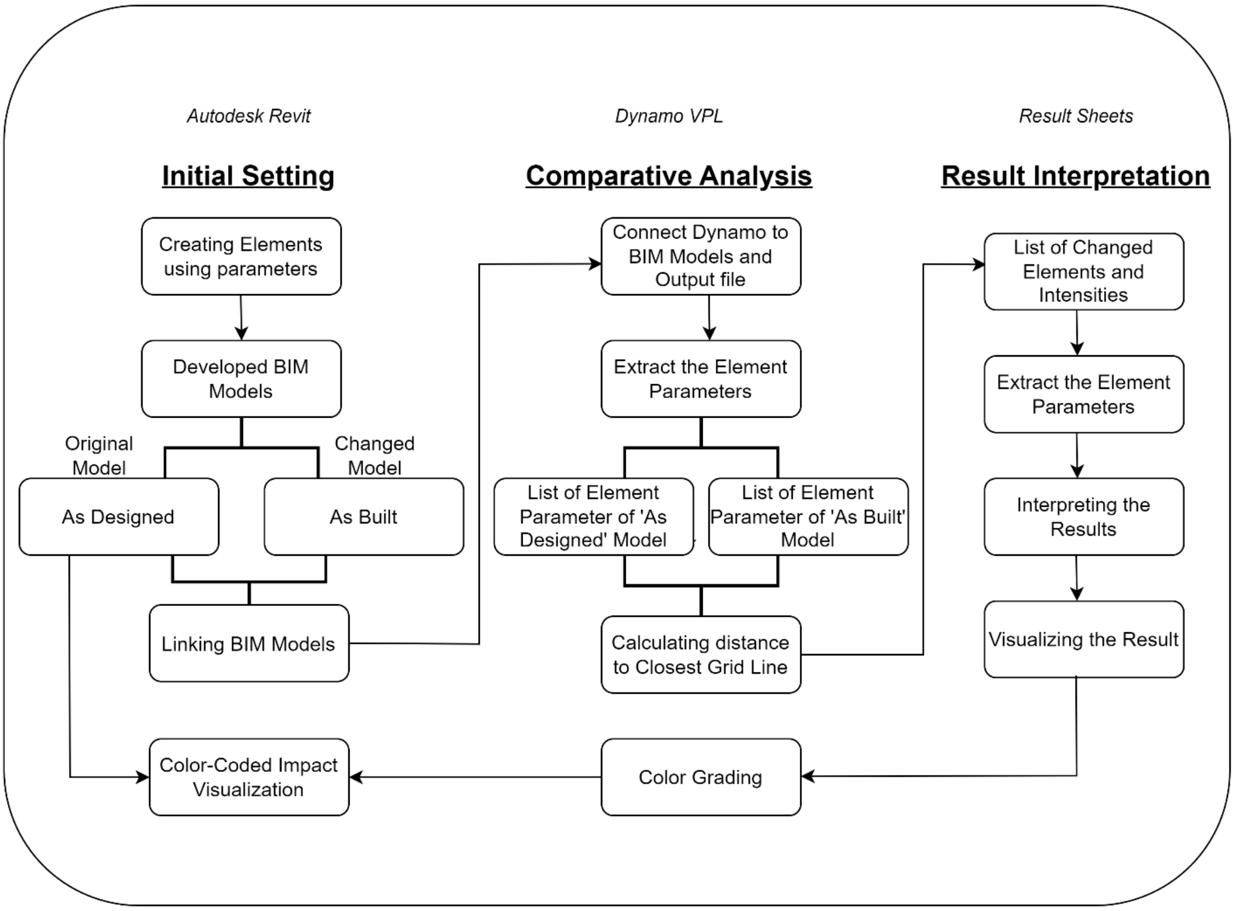

Autodesk Revit
Dynamo VPL
Result Sheets
Initial Setting
Comparative Analysis
Result Interpretation
Creating Elements using parameters
Developed BIM Models
Original Model
Changed Model
As Designed
As Built
Linking BIM Models
Connect Dynamo to BIM Models and Output file
Extract the Element Parameters
List of Element Parameter of 'As Designed' Model
List of Element Parameter of 'As Built' Model
Calculating distance to Closest Grid Line
List of Changed Elements and Intensities
Extract the Element Parameters
Interpreting the Results
Visualizing the Result
Color Grading
Color-Coded Impact Visualization

Figure 9 – BIM-enabled framework proposed in this thesis for detecting model changes.

3.2 BIM Model Generation

Leveraging the power of visual programming scripts, the process of generating building elements is streamlined and automated, significantly enhancing efficiency and productivity. By harnessing user data from spreadsheets, the scripts seamlessly translate design specifications into tangible elements within the Building Information Modeling (BIM) environment. As illustrated in Table 2, various types of elements, including walls, columns, doors, and windows, can be effortlessly generated with precision and accuracy.

Furthermore, to cater to diverse user needs and accommodate specific project requirements, additional scripts have been developed to facilitate the automatic creation of family element instances. This functionality empowers users to swiftly instantiate custom family elements tailored to their unique design preferences and project specifications. Moreover, the integration of Python script library adds an extra layer of functionality, allowing for the creation of plug-ins in the form of pushbuttons. These pushbutton plug-ins provide users with intuitive and user-friendly interfaces for executing individual scripts, further streamlining the process, and enhancing usability.

In addition to the core functionalities, the VPP ecosystem also offers supplementary scripts designed to address evolving user requirements. For instance, users have the flexibility to create new family instances dynamically as needed, ensuring adaptability and responsiveness to changing project demands.

Table 2 – Input for automated generation of building element (Wall).

Start x-coordinate	Start y-coordinate	End x-coordinate	End y-coordinate	Height	Level	Name	Elevation

The Table above is used to fill in the input data, the user must fill in the x, y, and z coordinates for the element. In the case of walls, both start and end point coordinates whereas for doors, windows, and column only the coordinate of the location point. Moreover, height, level, and the elevation of the element must be entered. The name in the Table indicates the family type name of that element.

3.3 Grid-Lines Creation

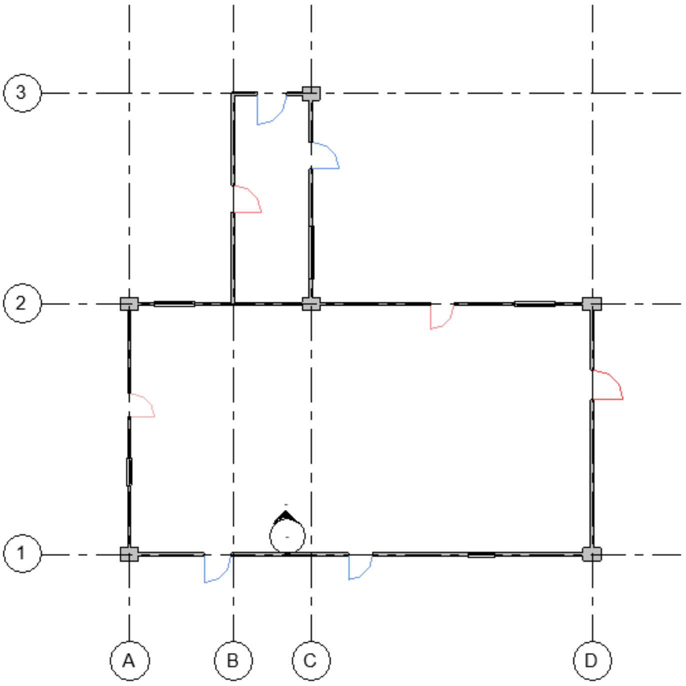


Figure 10 – View of generated Grid Lines.

The grids must be generated in the ‘as designed’ model. It is important to note that grids from the ‘as designed’ will be read to carry out further analysis.

The final step is to link the ‘as built' model with the ‘as designed' model such that both models lie exactly one above the other.

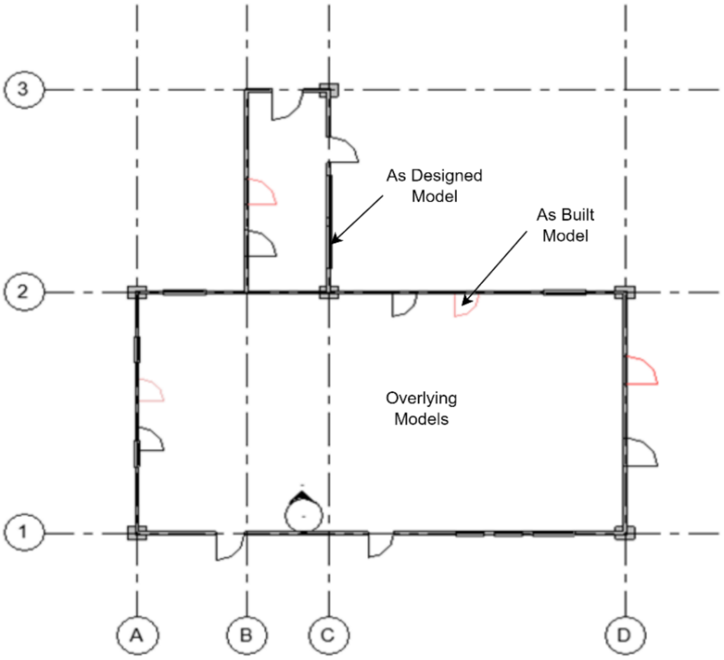


Figure 11 – Linking BIM Models using in-built Revit tool.

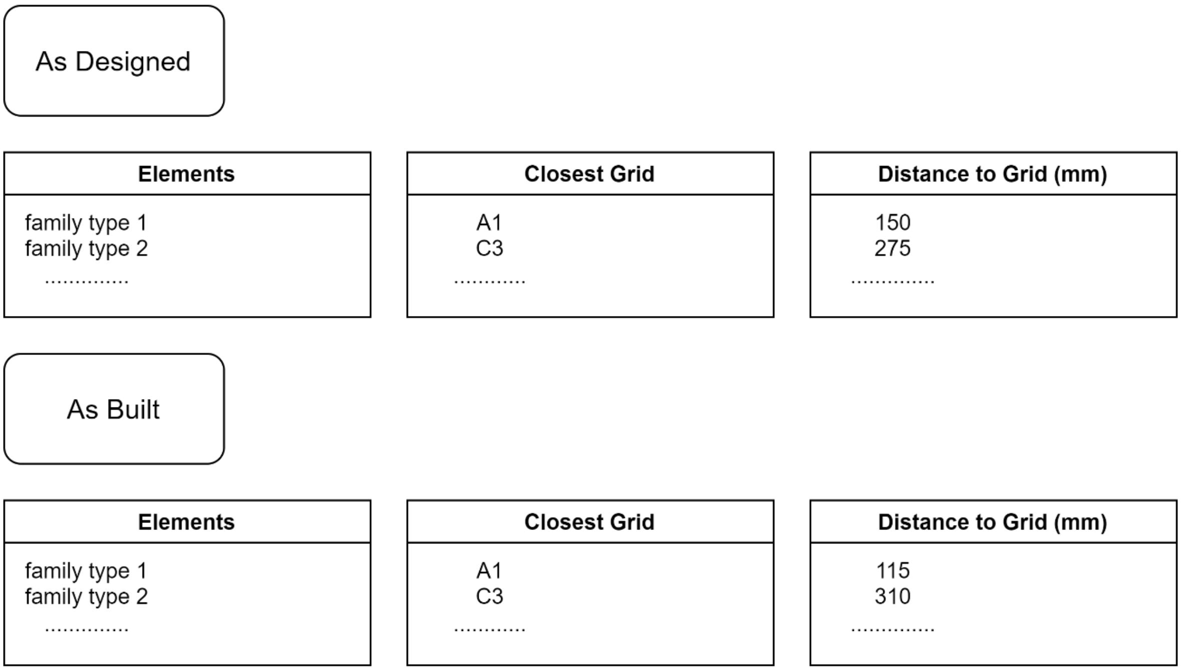
Using the ‘Manage Links’ tool of Revit, the ‘as built’ model can be linked to the ‘as designed’ model such that both models overlap each other as shown in Figure 12. The reason to make it overlap is to the fact that the analysis for the ‘as built’ model will be through the grids generated in the ‘as designed’ model.

3.4 Comparative Analysis

After the initial generation of the grids, the next step involves calculating the coordinates of points that connect these grids along the X and Y axes. These points are then assigned names corresponding to the connecting grids; for example, the connection between Grid ‘A’ and Grid ‘1’ is denoted as ‘A1’. Subsequently, the system checks the locations of building elements to identify the closest grid lines. This meticulous process ensures that the "as built" model aligns precisely with the "as designed" model, necessitating the use of identical grid points to calculate distances to specific elements accurately.

Automation is key to streamlining this process, achieved seamlessly through a visual programming script activated by pushbuttons. A detailed breakdown of the script's functionality and implementation can be found in the 'Implementation' section of the thesis. The resulting data includes the type of element, its corresponding nearest grid point, and the distance to that grid, presented comprehensively in Fig. 16.

The distance from the element to the grid is recorded foin both models and the deviation is considered as a model change. This method ensures precision and consistency in comparing elements between the "as designed" and "as built" models. The outcome will include both geometric and semantic information which means that the location of the elements will be detected for both models as well as the cross-section (width and height) of that particular element will also be identified, if there has been any changes in semantics, deviation for width and height have been included in the output Table.



As Designed

Elements	Closest Grid	Distance to Grid (mm)
family type 1	A1	150
family type 2	C3	275
....	....	....

As Built

Elements	Closest Grid	Distance to Grid (mm)
family type 1	A1	115
family type 2	C3	310
....	....	....

Figure 12 – Output example from comparative analysis for different BIM models.

3.5 Result Interpretation

In addition to detailing the element type for both the "as designed" and "as built" models, the results encompass vital information regarding their spatial relationship to the grid system. This includes identifying the nearest grid line and precisely measuring the distance to that grid, essential for assessing alignment accuracy. The Status part within the output serves as a crucial indicator, flagging whether an element has undergone any movement from its originally designated position. Moreover, the analysis extends beyond mere positional data, delving into deviations in both coordinates and cross-sections. These deviations are meticulously calculated, shedding light on any discrepancies between the intended design and the constructed reality. Furthermore, to gauge the degree of congruence between the models, the concept of similarity is introduced, utilizing tolerances of 5cm, 10cm, and 20cm for deviation. This holistic approach ensures a comprehensive evaluation of the fidelity and precision of the "as built" model relative to its "as designed" counterpart, facilitating informed decision-making and quality assurance in construction projects.

Following this, a visualization technique employing color grading is introduced to enhance user experience and expedite decision-making processes. Illustrated in Figure 14, this method employs a spectrum of colors to convey varying degrees of alteration within the elements.

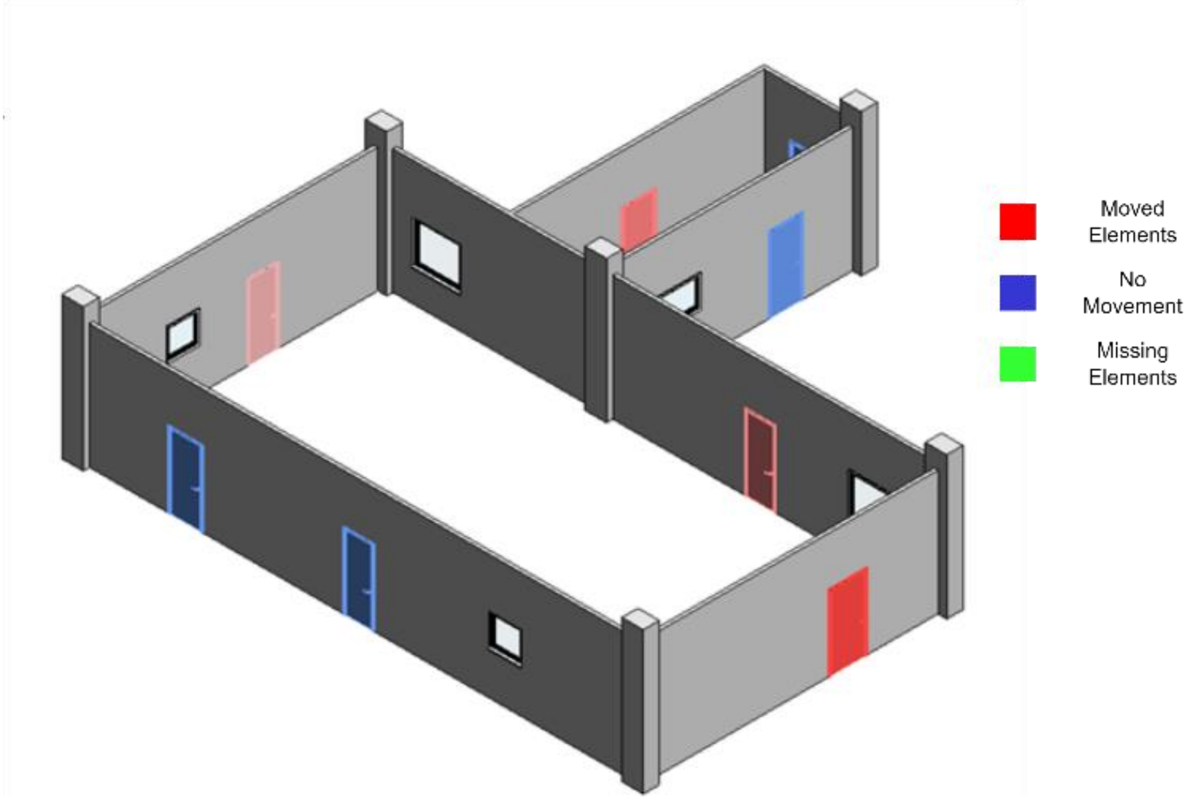


Figure 13 – Visualization examples for doors post comparative analysis in Revit.

3.6 Quantitative evaluation of Model Changes

When evaluating the resemblance between components in a model structure, it is necessary to have a metric that can provide a quantitative evaluation of these components. In this proposal, we present a method that defines completeness, correctness, and accuracy of datasets.

Completeness: As per equation (1), the numerator indicates the number of common elements between as-designed and as-built model (Nc) and the denominator of the fraction is the total number of elements in as-designed model (Nd) [40].

$$M_{comp} = \frac{Nc}{Nd} \tag{1}$$

Correctness: As per equation (2), the numerator indicates the number of common elements between as-designed and as-built model (Nc) and the denominator of the fraction is the total number of elements in as-built model (Nb) [40].

$$M_{corr} = \frac{Nc}{Nb} \tag{2}$$

Accuracy: As per equation (3), the numerator indicates the number of moved elements (Nm) which have shown movement less than a particular threshold limit (r) and the denominator of the fraction denotes the total number of elements (N). Here, thresholds of 5, 10, and 20 centimeters have been considered [40].

$$M_{\mathrm{Acc}} = \frac{\mathrm{Nm<r}}{N} \tag{3}$$

These ratios provide valuable insights into the overall coherence and consistency of the model, indicating the degree to which elements maintain their original positions. Higher accuracy ratios imply a higher level of uniformity and alignment among elements, while lower ratios may suggest inconsistencies or spatial discrepancies within the model. This approach enables efficient comparison and analysis of different model variations, which aids in making informed decisions and improving design processes.

4 Implementation and Results

4.1 Tools used for analysis

One of the most crucial steps in analysis involves software setup and dynamo package installation before it undergoes further modeling or analysis. Revit 2022 is used for visualization, PyRevit is used to create extensions in Revit, and Dynamo 2.10.1.4002 is used for creating visual programming scripts. The user has to download the following packages in order to test the framework error free: Clockwork, Crumple, Data-Shapes, GeniusLoci, Orchid, Springs, and WombatDynamo, archilab, Quasar, and Rhythm.

4.2 Data Sets

This section of the thesis presents an analysis of the outcomes obtained by applying the proposed framework for comparative analysis as explained in Chapter 3. The methodology has been tested and validated on two distinct datasets acquired from the academic institution of the Technical University of Munich.

4.2.1 Case Study

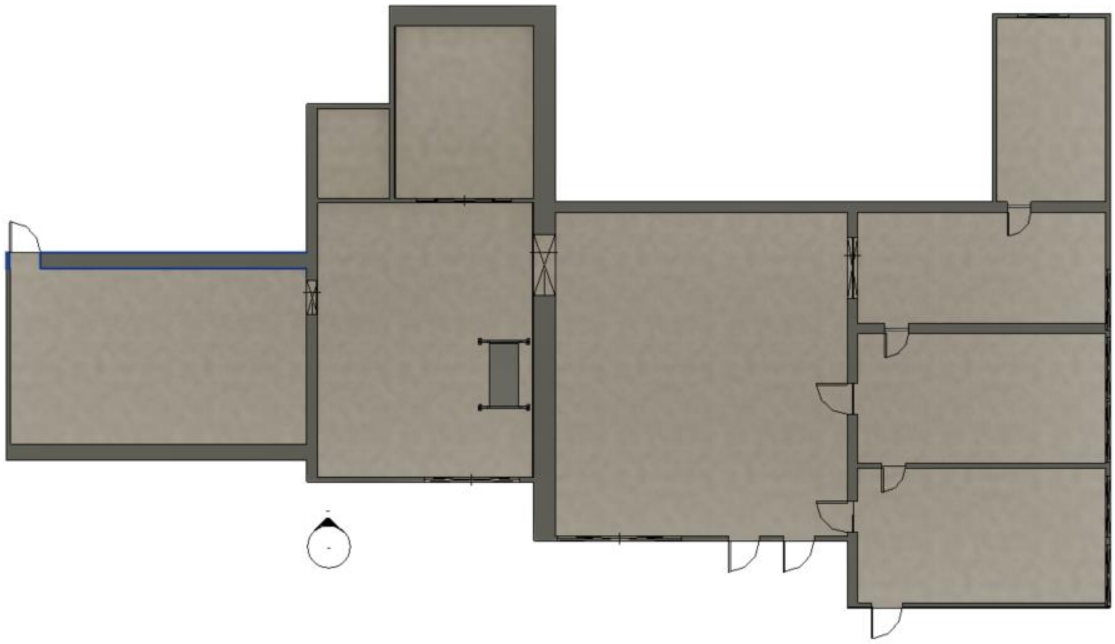


Figure 14 - Plan Model of Data Set 1.

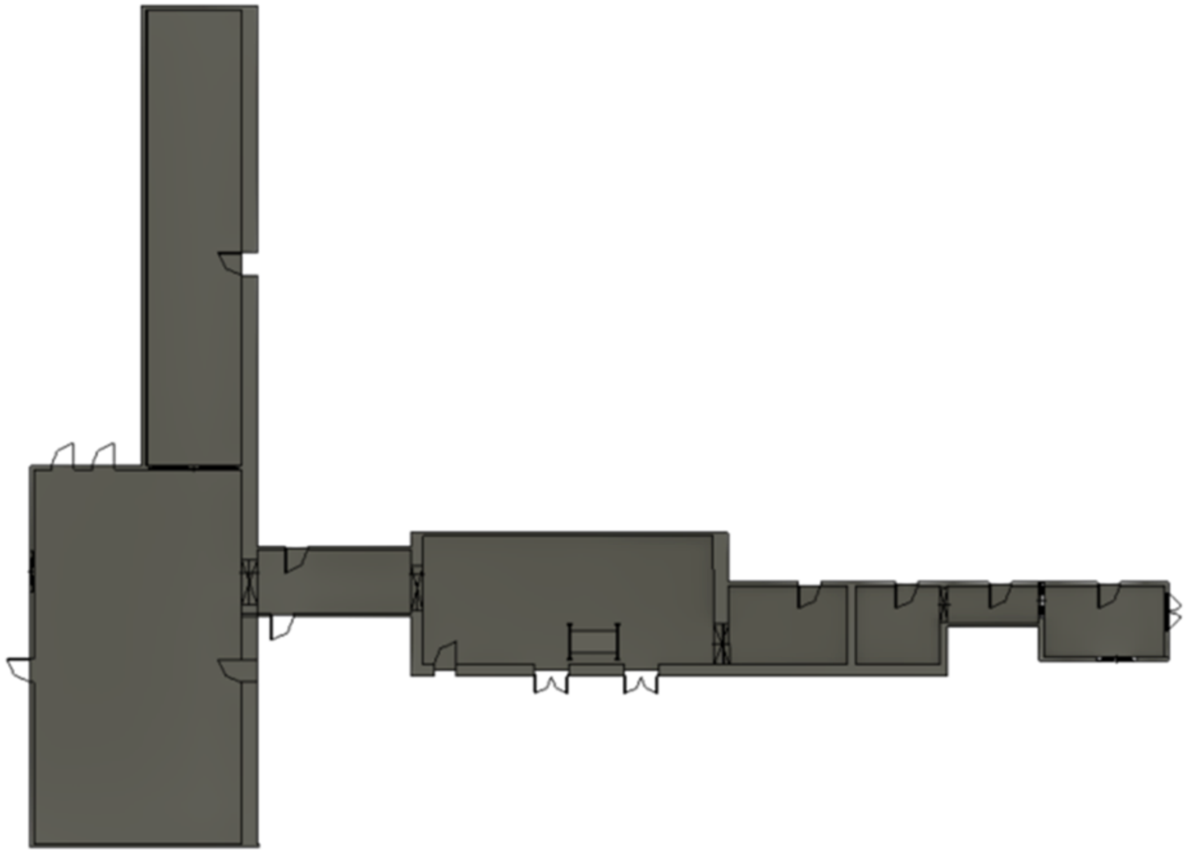


Figure 15 - Plan Model of Data Set 2.

Figures 15 and 16 above provide visual representations of two different BIM Models to be used for the analysis. Additionally, the Table below represents essential quantitative parameters for each dataset, including number of rooms, doors, windows and the total length and width of the models.

Table 10 – Data Sets Information.

Data Sets	No. of Rooms	No. of Doors	No. of Windows	Length (m)	Width (m)
Data Set 1	9	15	6	32.565	17.549
Data Set 2	8	19	1	33.747	45.693

4.2.2 Data Set 1

The dataset is provided by TUM consisting of 9 distinct rooms. Considering the model provided by TUM as ‘as designed‘ model and making few geometric changes purposely in the model and using it as ‘as built‘ model. There is no access to the ‘as built‘ model, therefore some modifications are made deliberately. This particular setup is required to carry out the analysis. Furthermore, the dataset covers all information regarding the building elements. Figures 17 and 18 represent the ‘as designed‘ and ‘as built‘ model for Dataset 1.

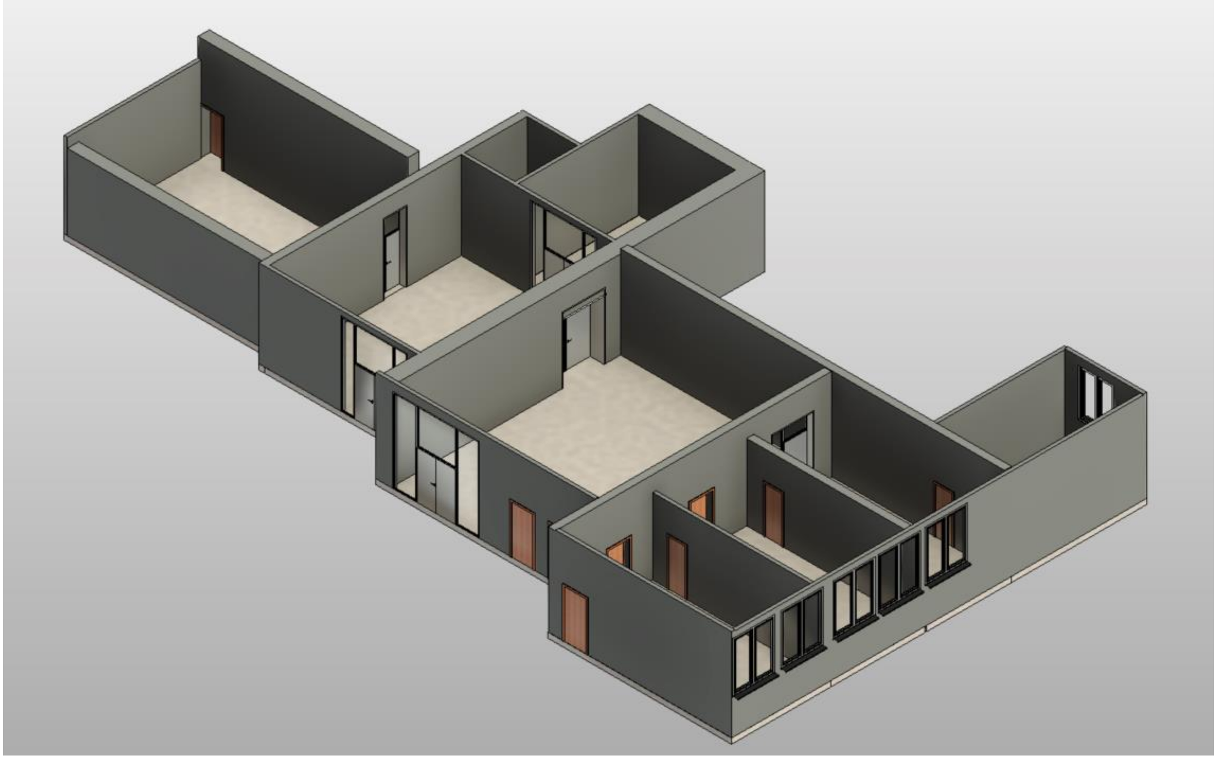


Figure 16 - Dataset 1 (as designed model).

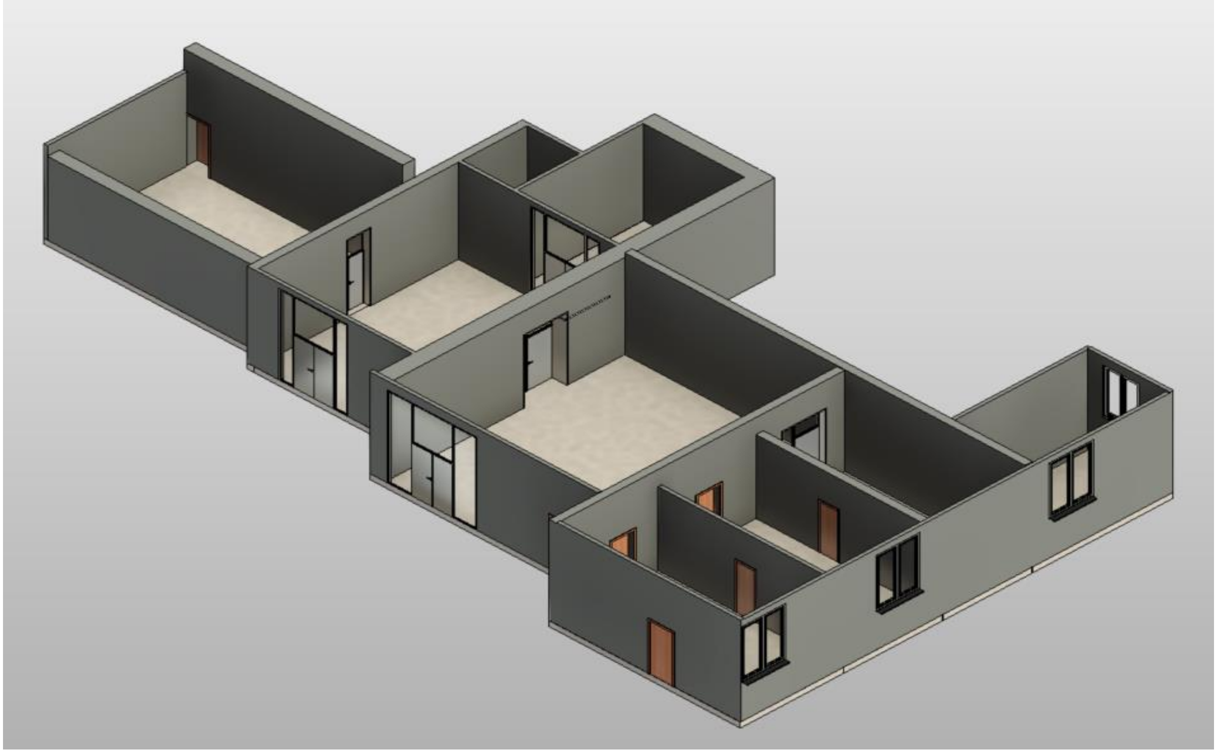


Figure 17 – Dataset 1 (as built model).

4.2.3 Data Set 2

The dataset is provided by TUM consisting of 8 distinct rooms. Considering the model provided by TUM as ‘as designed‘ model and making few geometric changes purposely in the model and using it as ‘as built‘ model. There is no access to the ‘as built‘ model, therefore some modifications are made deliberately. This particular setup is required to carry out the analysis. Furthermore, the dataset covers all information regarding the building elements. Figures 19 and 20 below represent the ‘as designed‘ and ‘as built‘ model for Dataset 2.

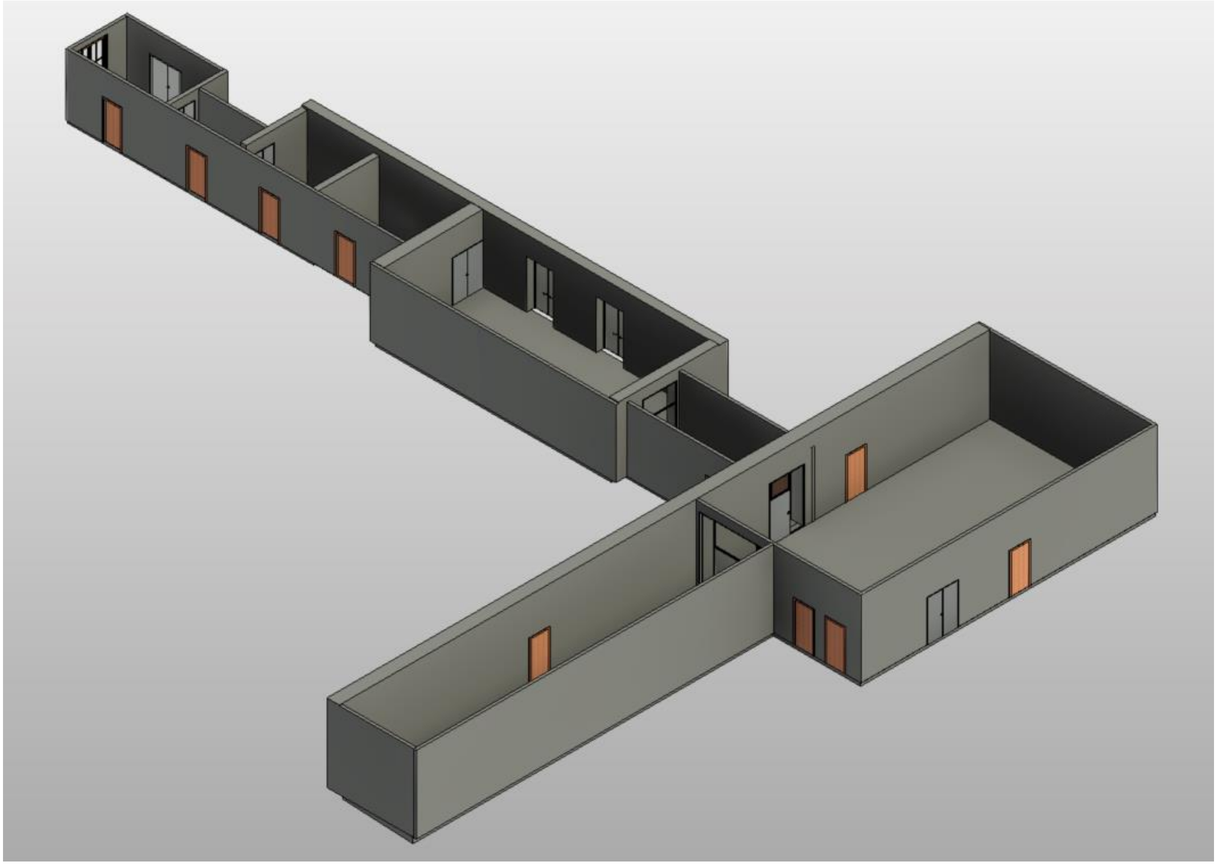


Figure 18 - Dataset 2 (as designed model).

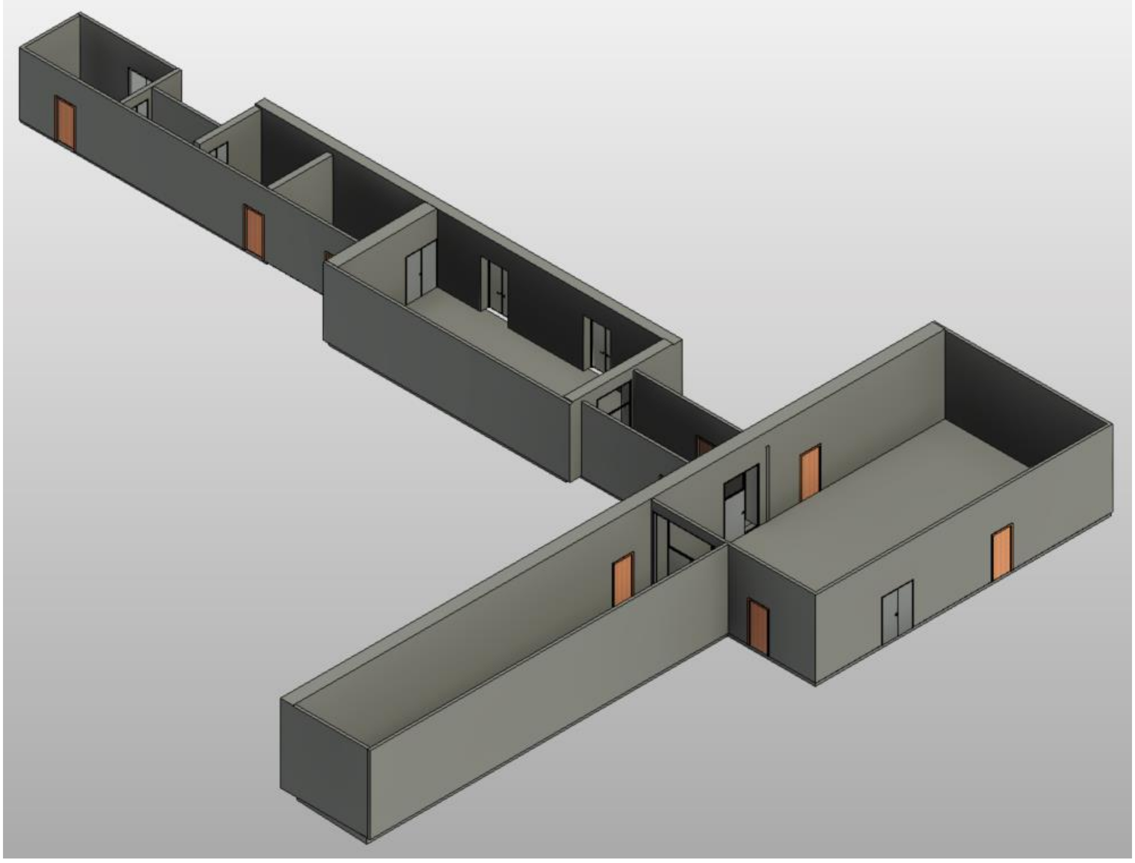


Figure 19 – Dataset 2 (as built model).

4.3 Dynamo Scripts Creation

4.3.1 Creating Family Instances

Wall:

- Input:

Table 3 – Input for creating wall instances.

Name	Thickness (mm)
wall-100	100
wall-200	200
wall-300	300
wall-400	400
wall-500	500

The input Table has the option to provide the name and thickness of the wall. The name is not dependent on the thickness and can be changed according to preference. There is no restriction on the number of instances. The wall types used in Wall-Ret_Conc can be easily modified in the script using a pulldown list. The dynamo code is prepared such that when run, a pop-up input box appears where the user can select the path to the input file. The scripts then runs and wall instances are automatically created. For reference, the entire script has been added in the Appendix.

Column:

- Input:

Table 4 – Input for creating column instances.

Width	Depth	Name
300	350	Col.300x350
400	450	Col.400x450
350	400	Col.350x400
350	450	Col.350x450

The input Table has the option to provide the width and depth of the column. The name has nothing to do with the dimensions, it can be renamed as per choice and there is no limitation on the number of instances. Any default column type can be chosen as a host and there are no limitations to the number of instances. The code takes the input and column instances are created. For reference, the entire script has been added in the Appendix.

Door:

- Input:

Table 5 – Input for creating door instances.

Width	Height	Name
400	800	D.400x800
400	820	D.400x820
350	750	D.350x750
500	1000	D.500x1000

The input Table has the option to provide the width and height of the door. The name has nothing to do with the dimensions, it can be renamed as per choice and there is no limitation on the number of instances. Any default door type can be chosen as a host and there are no limitations to the number of instances. The code reads the input provided by the user and creates door instances when run. The entire script can be referred in the Appendix.

Window:

- Input:

Table 6 – Input for creating window instances.

Width	Height	Name
400	800	W.400800
400	820	W.400820
350	750	W.350750
500	1000	W.5001000

The input Table has the option to provide the width and height of the window. The name has nothing to do with the dimensions, it can be renamed as per choice and there is no limitation on the number of instances. Any default window type can be chosen as a host and there are no limitations to the number of instances. The parameters (width and height) of the window is read by the code and window instances are generated. For more information, the script can be viewed in the Appendix.

4.3.2 Generating Building Elements

Wall:

- Input:

Table 7 – Input for Wall generation.

Start x-coordinate	Start y-coordinate	End x-coordinate	End y-coordinate	Height	Level	Name	Elevation
1000	1000	10000	1000	3000	Level 0	wall-300	0
10000	1000	10000	10000	3000	Level 0	wall-300	0
10000	10000	1000	10000	3000	Level 0	wall-300	0
1000	10000	1000	1000	3000	Level 0	wall-300	0
1000	1000	10000	1000	3000	Level 1	wall-300	3000
10000	1000	10000	10000	3000	Level 1	wall-300	3000
10000	10000	1000	10000	3000	Level 1	wall-300	3000
1000	10000	1000	1000	3000	Level 1	wall-300	3000
1000	1000	10000	1000	3000	Level 2	wall-300	6000
10000	1000	10000	10000	3000	Level 2	wall-300	6000
10000	10000	1000	10000	3000	Level 2	wall-300	6000
1000	10000	1000	1000	3000	Level 2	wall-300	6000

Table 7 is being used to collect the input data which includes placement points and the types of walls to be placed. For walls both coordinates for start and end point are necessary along with the height and family type of wall. These inputs are read by the code and 3D walls are automatically created. To refer, the total script has been mentioned in the Appendix section.

Column:

- Input:

Table 8 – Input for Column generation.

X-coordinate	Y-coordinate	Height	Level	Start Elevation	End Elevation	Name
1000	1000	3000	Level 0	0	3000	Col.300x350
10000	1000	3000	Level 0	0	3000	Col.400x450
10000	10000	3000	Level 0	0	3000	Col.350x400
1000	10000	3000	Level 0	0	3000	Col.350x450
1000	1000	3000	Level 1	3000	6000	Col.300x350
10000	1000	3000	Level 1	3000	6000	Col.400x450
10000	10000	3000	Level 1	3000	6000	Col.350x400
1000	10000	3000	Level 1	3000	6000	Col.350x450
1000	1000	3000	Level 2	6000	9000	Col.300x350
10000	1000	3000	Level 2	6000	9000	Col.400x450
10000	10000	3000	Level 2	6000	9000	Col.350x400
1000	10000	3000	Level 2	6000	9000	Col.350x450

Table 8 is being used to collect the input data which includes placement points and the types of columns to be placed. The coordinates and height are read by the code to place the column at the user defined location. The family name can be chosen by the user and the script has the capability to read and understand it. After having all those inputs, columns are generated in 3D. To have a deeper understanding, the code can be referred to the Appendix section.

Door and Window:

- Input:

Table 9 is being used to collect the input data which includes placement points and the types of Door to be placed. After having the input data ready, the script creates doors and windows in 3D. Since doors and windows are both instances of wall, the host wall is detected automatically by the scripts and the scripts is therefore similar for wall instances (doors and windows). For more information, the script can be referred to as in the Appendix section.

Table 9 – Input for Door and Window generation.

x-coordinate	y-coordinate	z-coordinate	Level	Name
4000	1000	0	Level 0	D.400x800
10000	4000	0	Level 0	D.400x820
4000	10000	0	Level 0	D.350x750
1000	4000	0	Level 0	D.500x1000
4000	1000	3000	Level 1	D.400x800
10000	4000	3000	Level 1	D.400x820
4000	10000	3000	Level 1	D.350x750
1000	4000	3000	Level 1	D.500x1000
4000	1000	6000	Level 2	D.400x800
10000	4000	6000	Level 2	D.400x820
4000	10000	6000	Level 2	D.350x750
1000	4000	6000	Level 2	D.500x1000

4.4 Grid Generation

- Script

To generate grids, it is important to detect all walls and columns in the model. The scripts is prepared such the required elements are detected and sorted, for instance, if two elements are too close (500 mm), they are merged into a single grid to avoid congestion in the plan. The nodes 'Element.GetLocation' and 'Grid.ByLine' have been used for generating the grid. However, there is still a need to make it comprehensible.

Using Vector function, the horizontal and vertical grids are separated and named in alphabetical and numerical order respectively with the node 'Element.SetParameterByName'. For more information, the entire script has been added in the Appendix.

- Test

The nomenclature of the analysis is dependent on the Grids as the element location in two models will be compared using the Grids. Figures 21 and 22 showcase the grids created automatically for both Datasets.

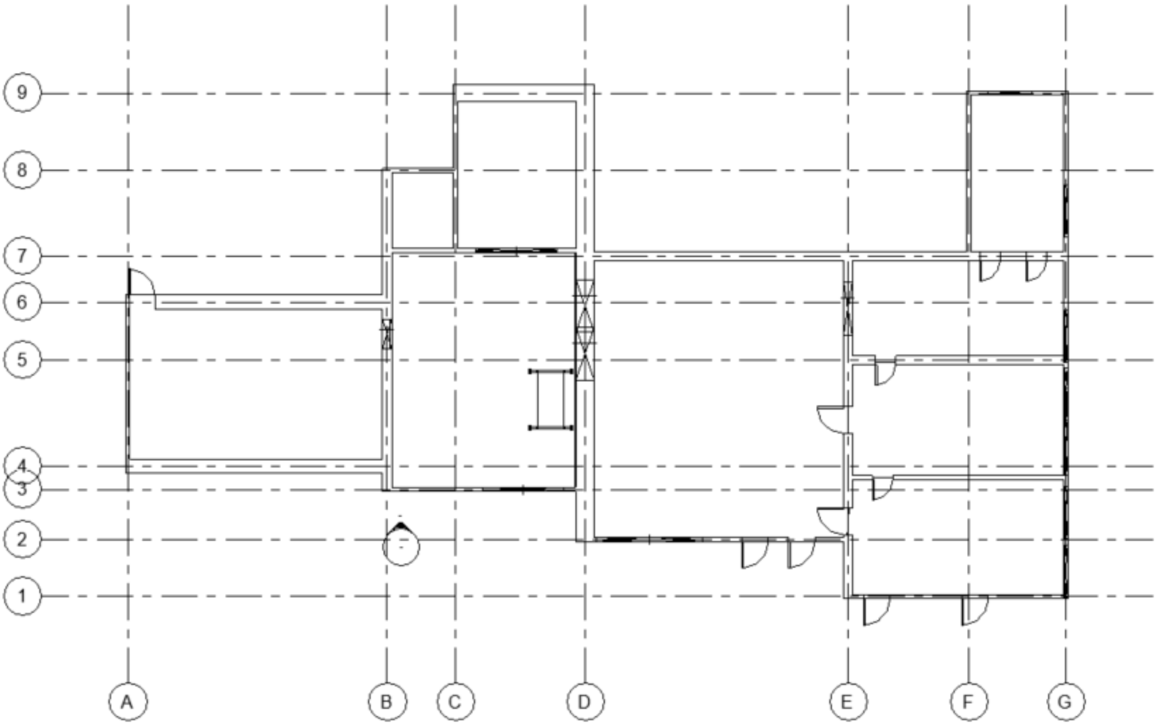


Figure 20 – Grid Creation for Dataset 1.

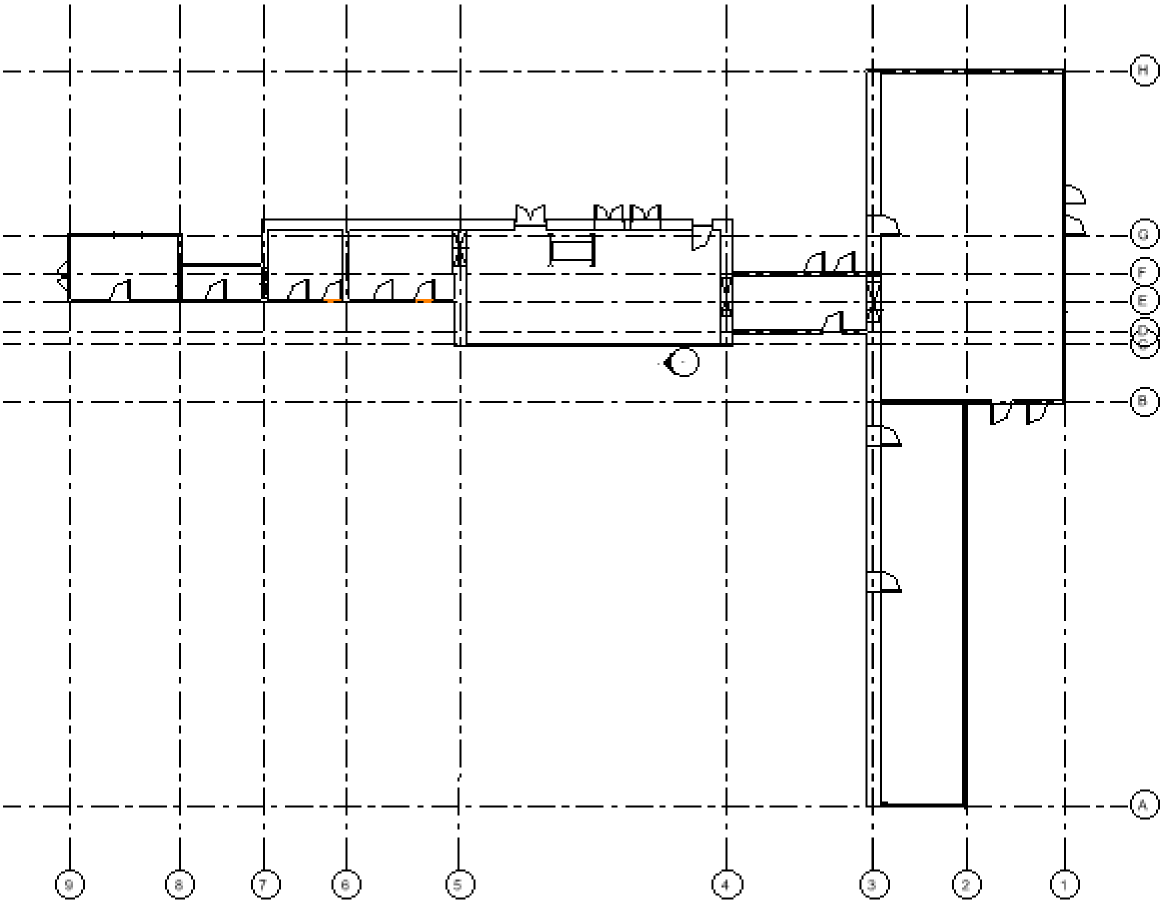


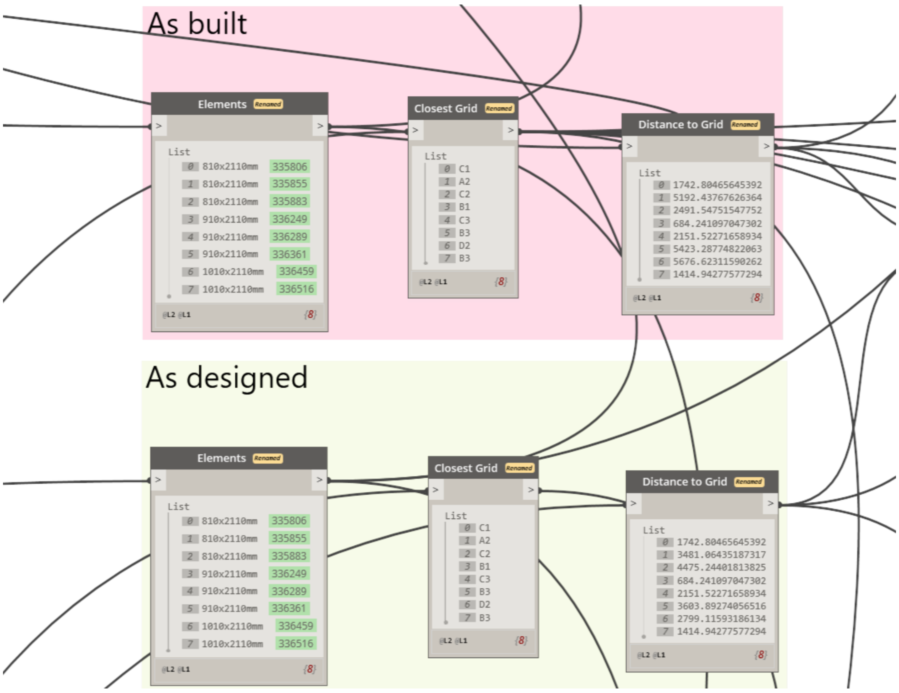
Figure 21 – Grid Creation for Dataset 2.

4.5 Comparative Analysis

- Script

In order to carry out comparative analysis of ‘As designed’ and ‘As built’ Models, the user has to link the ‘As built’ model with the ‘As designed’ model. When the script is run, the elements from both models are read by the script as shown in Figure 23 and simultaneously the location of elements are calculated with the corresponding closest grid point.

As shown in Figure 23, the elements from both models are compared and the results in terms of Element Type, Closest Grid name and Distance to Grid are computed.



As built – Elements:

```
List
  0 810x2110mm  335806
  1 810x2110mm  335855
  2 810x2110mm  335883
  3 910x2110mm  336249
  4 910x2110mm  336289
  5 910x2110mm  336361
  6 1010x2110mm  336459
  7 1010x2110mm  336516
@L2 @L1                {8}
```

As built – Closest Grid:

```
List
  0 C1
  1 A2
  2 C2
  3 B1
  4 C3
  5 B3
  6 D2
  7 B3
@L2 @L1    {8}
```

As built – Distance to Grid:

```
List
  0 1742.80465645392
  1 5192.43767626364
  2 2491.54751547752
  3 684.241097047302
  4 2151.52271658934
  5 5423.28774822063
  6 5676.62311590262
  7 1414.94277577294
@L2 @L1            {8}
```

As designed – Elements:

```
List
  0 810x2110mm  335806
  1 810x2110mm  335855
  2 810x2110mm  335883
  3 910x2110mm  336249
  4 910x2110mm  336289
  5 910x2110mm  336361
  6 1010x2110mm  336459
  7 1010x2110mm  336516
@L2 @L1                {8}
```

As designed – Closest Grid:

```
List
  0 C1
  1 A2
  2 C2
  3 B1
  4 C3
  5 B3
  6 D2
  7 B3
@L2 @L1    {8}
```

As designed – Distance to Grid:

```
List
  0 1742.80465645392
  1 3481.06435187317
  2 4475.24401813825
  3 684.241097047302
  4 2151.52271658934
  5 3603.89274056516
  6 2799.11593186134
  7 1414.94277577294
@L2 @L1            {8}
```

Figure 22 – Comparative Analysis script part 1.

The next part of the script as shown in Figure 24 includes visualization, the user can identify objects as per color coding. A dark red color denotes significant alterations, highlighting areas where substantial changes have occurred between the "as designed" and "as built" models. Conversely, a lighter shade of red signifies minor alterations, indicating areas where deviations from the original design are less pronounced. Elements that remain unchanged from the initial design are represented in a blue shade, indicating their stationary status. Additionally, to facilitate easy identification of missing elements in the "as built" model, these are visually depicted in a distinct green color within the "as designed" model. This intuitive visualization scheme not only provides a comprehensive overview of the degree of alteration but also aids in pinpointing areas requiring attention or further investigation. By leveraging color coding, users can swiftly discern critical information and make informed decisions, thereby streamlining the assessment and validation process of the construction project.

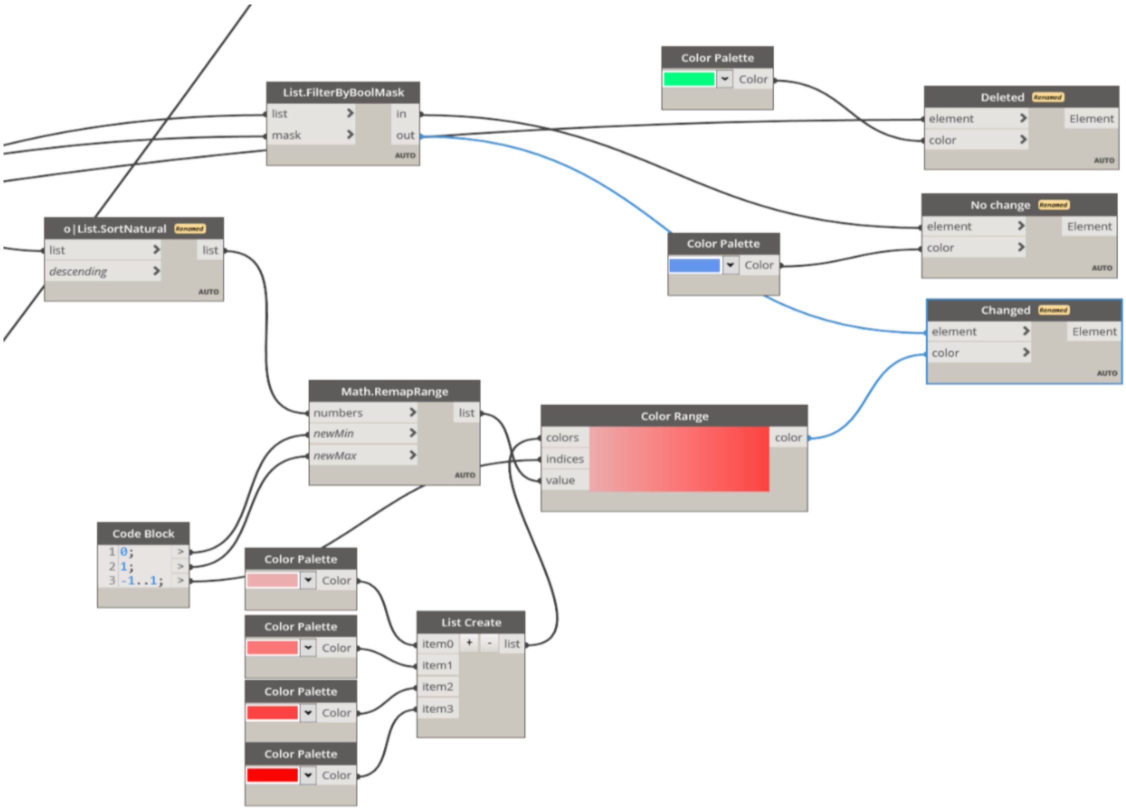


Figure 23 – Comparative Analysis script part 2.

➢ Test

Having separate scripts for each building element makes it flexible for the user and reducing the run time. Quantitative results can be exported to csv or excel file. The results obtained after the analysis are presented below. The results consist of the element name, the closest grid, and the distance to it. Along with Delta coord which indicates the intensity of movement, Delta W (width) and Delta H (height) which indicates the changes in the cross-section of building element. Lastly, similarities of 5cm,10cm, and 20 cm are added to the results clarifying how similar are the two models. For instances, Simi (5 cm) as 0 % indicates that no elements have moved less than 5 cm and Simi (10 cm) as 50 % indicates that half of the elements have shown movement less than 10 cm.

Table 11 – Model Changes Analysis Result for Doors (Dataset 1).

		As Designed		As Built			
Element Name	**Corresponding Grid**	**Distance to Grid(mm)**	**Status**	**Distance to Grid(mm)**	**Delta Coord (mm)**	**Delta W**	**Delta H**
0915 x 2134mm	A6	659.65	--	659.65	0	0	0
DL - 800 x 2000 - ML - 1400 x 2501	B5	921.07	Moved	348.91	1239	0	0
ML - 1635 x 2500 - DL - 800 x 1820 - EI 31	D6	498.33	Moved	1846.49	1649	0	0
0915 x 2032mm	E2	1653.54	--	1653.54	0	0	0
0915 x 2032mm	E1	1101.45	Moved	4424.94	3395	0	0
ML - 1635 x 2500 - DL - 800 x 1820 - EI 31	E6	502.52	--	502.52	0	0	0
0915 x 2032mm	E4	1643.98	--	1643.98	0	0	0
0915 x 2032mm	E2	769.35	--	769.35	0	0	0
DL - 1600 x 2000 - ML - 2800 x 2500 - EI 30	D2	2222.52	--	2222.52	0	0	0
DL - 1600 x 2000 - ML - 2800 x 2500 - EI 31	C7	2121.03	--	2121.03	0	0	0
DL - 1600 x 2000 - ML - 2800 x 2500 - EI 31	D3	2147.6	Moved	4321.29	2175	0	0
0915 x 2134mm _Smaller	F7	795.14	Moved	2344.4	1600	0	0
0915 x 2134mm _Smaller	E5	1343.07	Moved	3128.25	1810	0	0
0915 x 2134mm _Smaller	E3	1202.06	Moved	3710.66	2510	0	0
Accuracy (5 cm)	0						
Accuracy (10 cm)	0						
Accuracy (20 cm)	0						
Completeness	0.93						
Correctness	1						

Table 12 – Model Changes Analysis Result for Windows (Dataset1).

	As Designed			As Built			
Element Name	**Corresponding Grid**	**Distance to Grid(mm)**	**Status**	**Distance to Grid(mm)**	**Delta Coord (mm)**	**Delta W**	**Delta H**
2000 x 1400	G1	870.2	--	870.2	0	0	0
2000 x 1400	G5	840.92	Moved	5170.74	4330	0	0
2000 x 1400	G4	597.51	Moved	1797.31	1200	0	0
1500 x 2000 mm	F9	1420.05	--	1420.05	0	0	0
Accuracy (5 cm)	0						
Accuracy (10 cm)	0						
Accuracy (20 cm)	0						
Completeness	0.67						
Correctness	1						

Table 13 – Model Changes Analysis Result for Doors (Dataset 2).

	As Designed			As Built			
Element Name	**Corresponding Grid**	**Distance to Grid(mm)**	**Status**	**Distance to Grid(mm)**	**Delta Coord (mm)**	**Delta W**	**Delta H**
0915 x 2032mm	G1	633.96	Moved	1904.15	1400	0	0
Gleiche Feldbreit-en - EI 30	E1	522.46	--	522.46	0	0	0
0915 x 2032mm	G3	646.69	--	646.69	0	0	0
0915 x 2032mm	B3	8203.92	Moved	1585.6	6670	0	0
0915 x 2032mm	B2	1687.27	--	1687.27	0	0	0
ML - 1635 x 2500 - DL - 800 x 1820 - EI 31	E3	483.78	--	483.78	0	0	0
DL - 1600 x 2000 - ML - 2800 x 2500 - EI 31	B2	2009.75	--	2009.75	0	0	0
Gleiche Feldbreit-en - EI 31	G5	625.61	--	625.61	0	0	0
EN 179 - ML - 1260 x 2135 - GF - 800 - EI 30	F8	630.51	--	630.51	0	0	0
Gleiche Feldbreit-en - EI 31	G9	2032.65	Moved	3348.97	1317	0	0
Series SR40 - FD 30S - 2085x1799mm	G5	3289.49	--	3289.49	0	0	0
Series SR40 - FD 30S - 2085x1799mm	G4	5443.06	--	3769.79	1700	0	0
0915 x 2032mm	G4	1126.88	--	1126.88	0	0	0
0915 x 2032mm	F3	1390.08	Moved	2753.1	1400	0	0
0915 x 2032mm	D3	1929.54	--	1929.54	0	0	0
DL - 800 x 2000 - ML - 1400 x 2501	E4	251.17	--	251.17	0	0	0
0915 x 2134mm	E9	2360.91	--	2360.91	0	0	0
0915 x 2134mm	E7	1563.21	Moved	3131.44	1600	0	0
0915 x 2134mm	E6	1755.42	Moved	3626.5	1900	0	0
EN 179 - ML - 1260 x 2135 - GF - 800 - EI 30	F7	503.31	--	503.31	0	0	0
Accuracy (5 cm)	0						
Accuracy (10 cm)	0						
Accuracy (20 cm)	0						
Completeness	0.91						
Correctness	1						

Table 14 – Model Changes Analysis Result for Windows (Dataset 2).

	As Designed			As Built			
Element Name	Corresponding Grid	Distance to Grid(mm)	Status	Distance to Grid(mm)	Delta Coord (mm)	Delta W	Delta H
Accuracy (5 cm)	0						
Accuracy (10 cm)	0						
Accuracy (20 cm)	0						
Completeness	0						
Correctness	0						

Since no windows have shown movement, the results for windows are null.

Apart from the quantitative results, visualization of results has been carried simultaneously in Revit as shown in the Figures 25 and 26. The blue elements indicate no change in both models whereas red elements indicate some movement, the darker the red colour is the more is the movement and vice versa. The elements marked in green indicate the elements are missing in the 'as built' model. Similarly, the framework can be analysed to detect model changes for other elements like columns and walls.

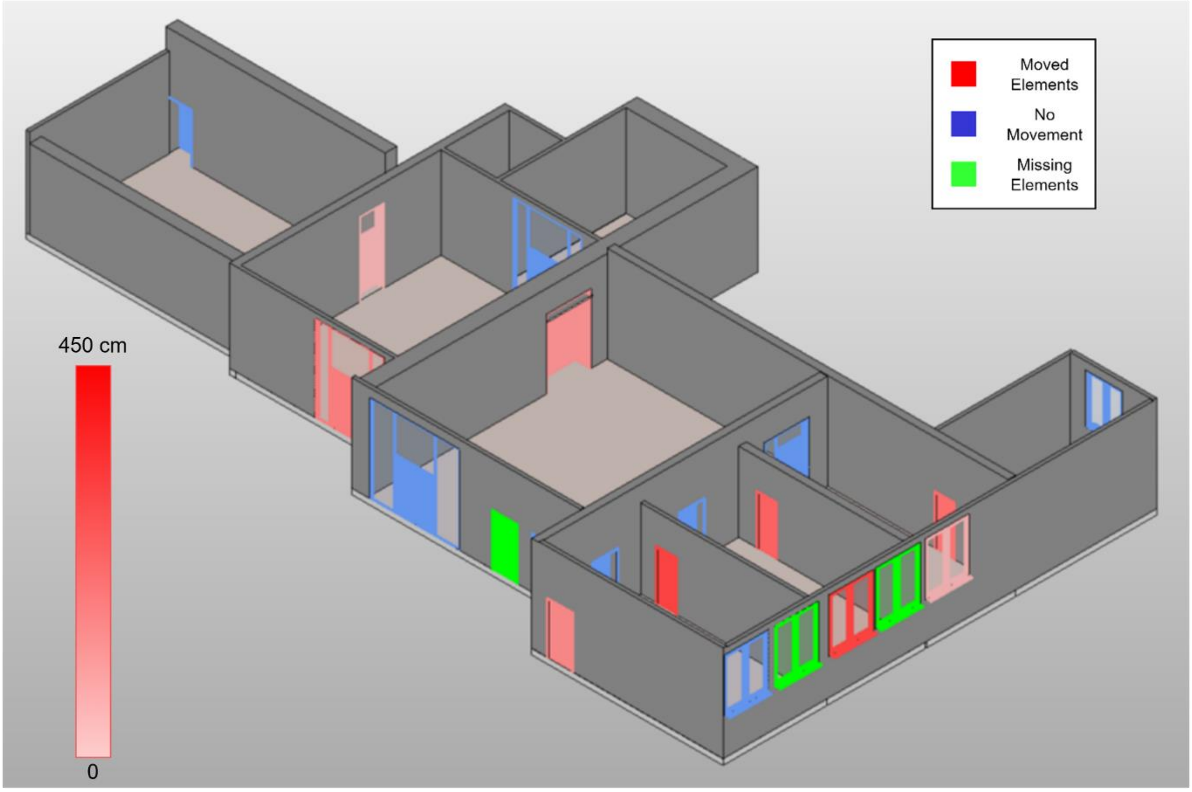


Figure 24 – Visualization of Results obtained from comparative analysis for Dataset 1.

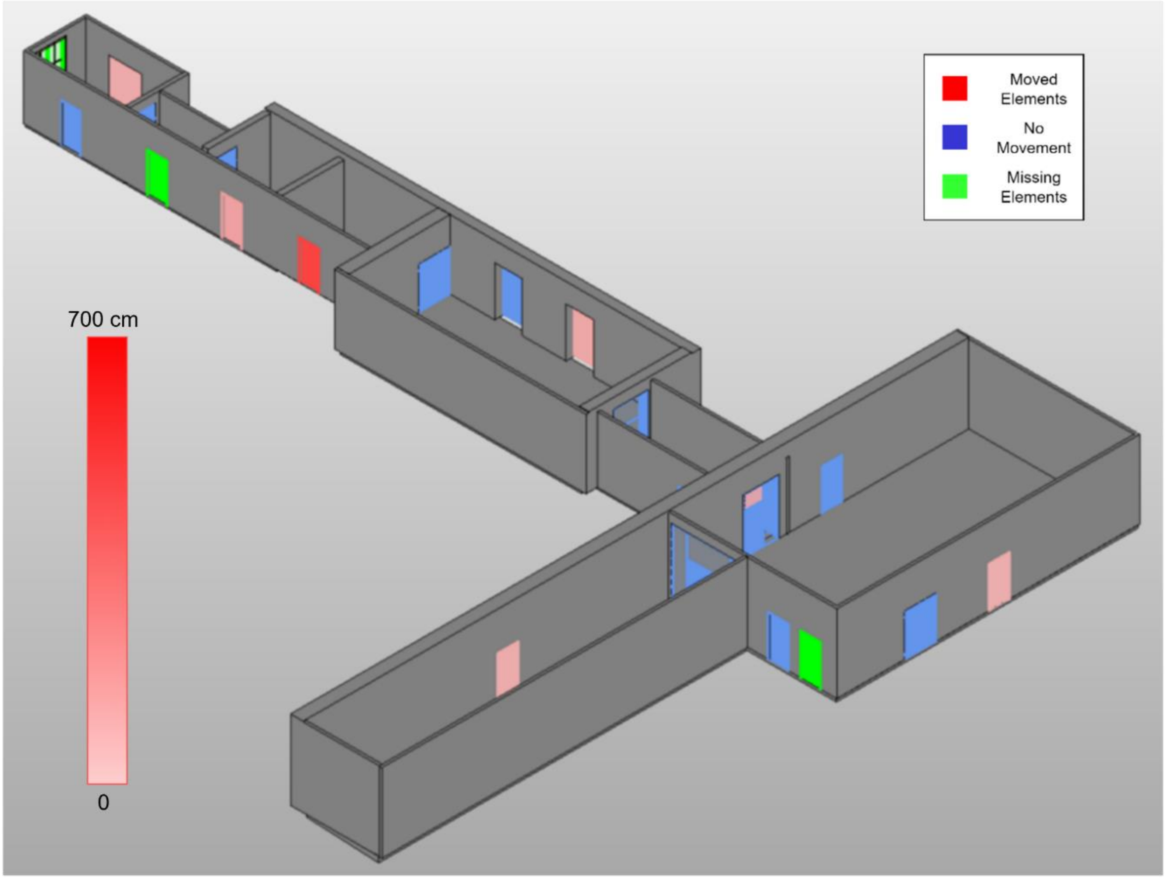


Figure 25 – Visualization of Results obtained from comparative analysis for Dataset 2.

4.6 MultiStorey Test

A two storeyed building synthetic data is tested in order to check the capability of the approach for multi-storeyed building. The building contains 8 columns, 13 doors, and 11 windows. The original model is considered to be ‘as designed‘ model and some changes are made deliberately and considered to be ‘as built‘ model. Such type of setup is necessary to run the analysis.

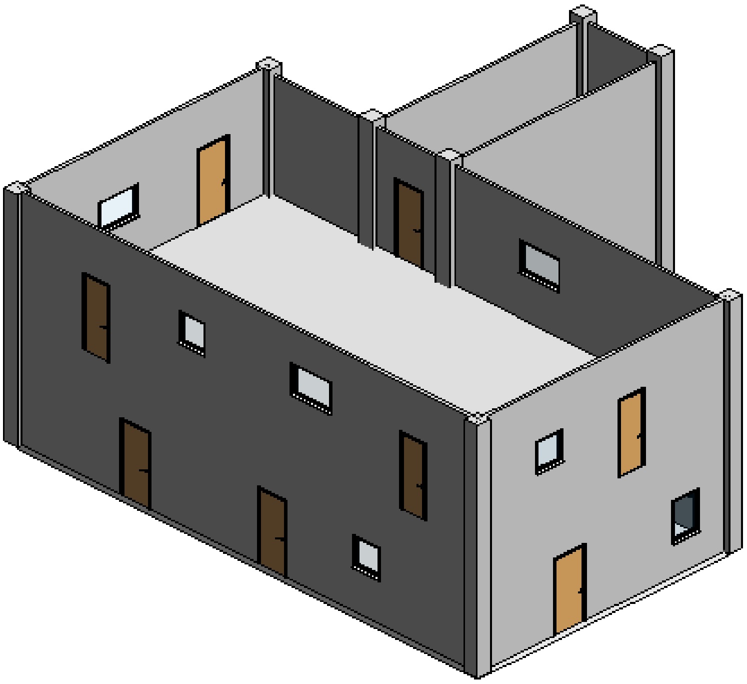


Figure 26 – MultiStorey Building (as designed model).

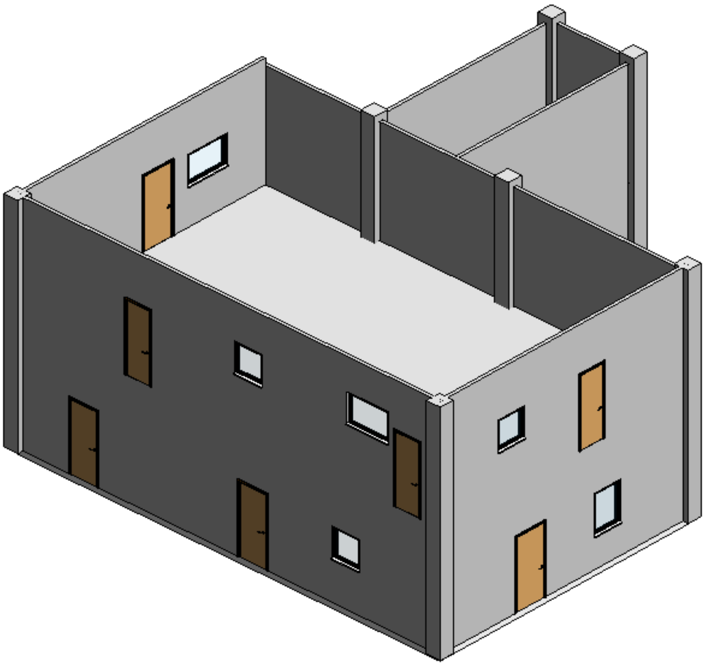


Figure 27 - MultiStorey Building (as built model).

The next step is creating grid lines and link both the models. As mentioned before, model changes are detected by measuring the distance to the closest grid line. Therefore, it is important that the grid lines are created appropriately. Figure 29 gives a glance of generated gird lines. Moreover, it is also important to link both the models such that they overlay each other.

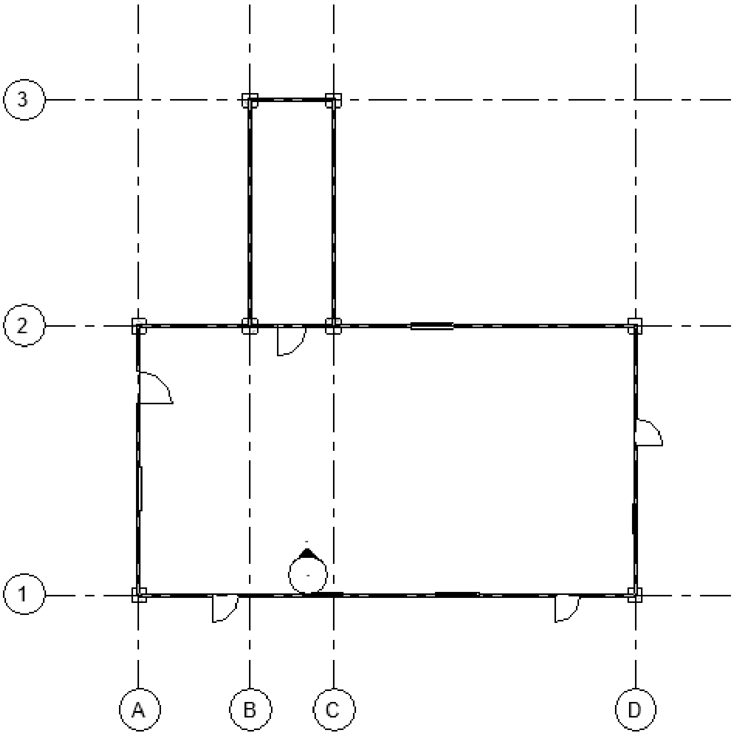


Figure 28 – Grid generation for Multistorey building.

Finally, the Models are run to detect changes, the results are displayed below.

Table 15 – Analysis Result for Columns (Multistorey).

Element Name	Corresponding Grid	Status	Distance to Grid(mm)	Delta W	Delta H
450x450mm	B3	Moved	1052.63	0	0
450x450mm	A1	--	0	0	0
450x450mm	D1	Moved	2056.12	0	0
450x450mm	D2	Moved	2056.12	0	0
450x450mm	C3	Moved	1052.63	0	0
450x450mm	C2	Moved	1690.14	0	0
450x450mm	B2	--	0	0	0
Accuracy (5 cm)	0				
Accuracy (10 cm)	0				
Accuracy (20 cm)	0				
Completeness	0.88				
Correctness	1				

Table 16 – Analysis Result for Doors (Multistorey).

	As Designed			As Built			
Element Name	**Corresponding Grid**	**Distance to Grid(mm)**	**Status**	**Distance to Grid(mm)**	**Delta Coord (mm)**	**Delta W**	**Delta H**
810x2110mm	A2	3481.06	--	3481.06	0	0	0
810x2110mm	C2	4475.24	--	4475.24	0	0	0
910x2110mm	C3	2151.52	--	2151.52	0	0	0
910x2110mm	B3	3603.89	--	3603.89	0	0	0
1010x2110mm	B3	1414.94	--	1414.94	0	0	0
810x2110mm	B1	879.39	Moved	604.88	1200	0	0
810x2110mm	D1	2196.54	Moved	3084.44	900	0	0
810x2110mm	D2	3381.83	--	3381.83	0	0	0
910x2110mm	B1	731.22	Moved	1392.77	1900	0	0
910x2110mm	C1	2628.57	Moved	1555.63	1100	0	0
1010x2110mm	A2	2016.69	Moved	3787.48	1800	0	0
1010x2110mm	D1	3094.84	--	3094.84	0	0	0
Accuracy (5 cm)	0						
Accuracy (10 cm)	0						
Accuracy (20 cm)	0						
Completeness	0.92						
Correctness	1						

Table 17 – Analysis Result for Windows (Multistorey).

	As Designed			As Built			
Element Name	**Corresponding Grid**	**Distance to Grid(mm)**	**Status**	**Distance to Grid(mm)**	**Delta Coord (mm)**	**Delta W**	**Delta H**
910x910mm	D1	3759	Moved	5058.1	1300	0	0
910x910mm	A1	2813.2	--	2813.2	0	0	0
1360x1210mm	D2	1936.83	--	1936.83	0	0	0
1360x1210mm	A2	1538.61	--	1538.61	0	0	0
1810x910mm	C2	1762.51	--	1762.51	0	0	0
910x910mm	C1	228.7	Moved	1348.99	1500	0	0
910x910mm	D1	2412.23	--	2412.23	0	0	0
1360x910mm	C1	3919.49	Moved	5218.65	1300	0	0
1360x910mm	A1	3387.4	Moved	6485.54	3100	0	0
910x1210mm	D2	1513.75	Moved	2809.7	1300	0	0
Accuracy (5 cm)	0						
Accuracy (10 cm)	0						
Accuracy (20 cm)	0						
Completeness	0.91						
Correctness	1						

Table 18 – Analysis Result for Walls (Multistorey).

Element Name	Status	Corresponding Grid	Distance to Grid(mm)	Delta W
wall-100	--	1	0	0
wall-100	--	2	0	0
wall-100	Moved	3	1052.63	0
wall-100	Moved	3	1052.63	0
wall-100	--	2	0	0
wall-100	--	1	0	0
wall-100	Moved	D	2056.12	0
wall-100	--	A	0	0
wall-100	--	B	0	0
wall-100	--	C	0	0
wall-100	--	B	0	0
wall-100	--	A	0	0
wall-100	Moved	D	2056.12	0
wall-100	--	C	0	0
Accuracy (5 cm)	0			
Accuracy (10 cm)	0			
Accuracy (20 cm)	0			
Completeness	1			
Correctness	1			

Apart from the quantitative results, visualization of results has been carried simultaneously in Revit as shown in Figure 30.

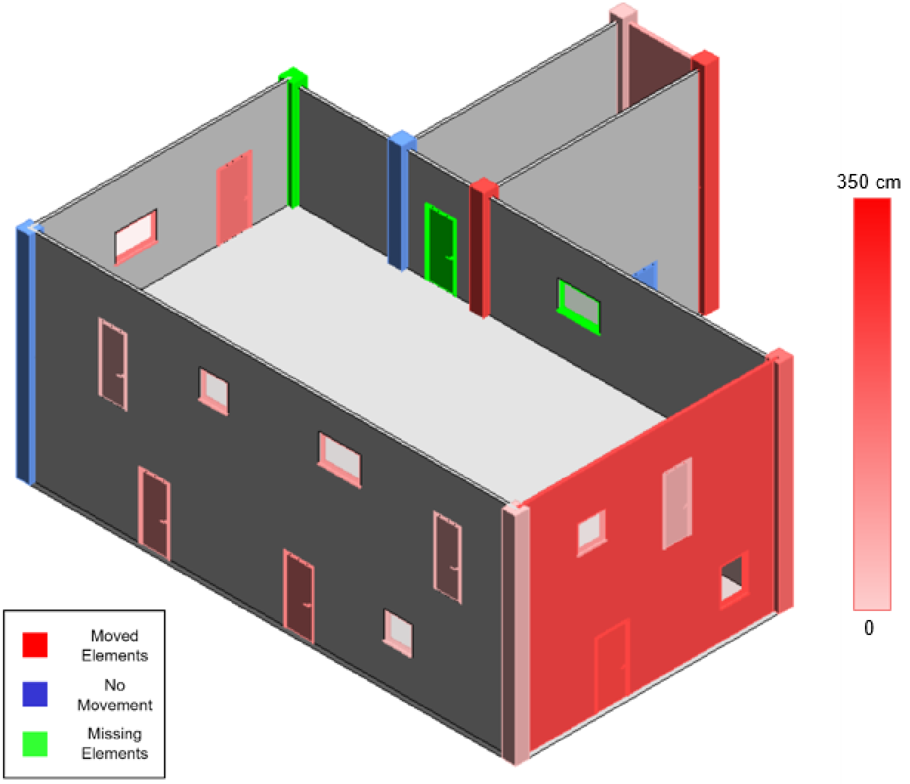


Figure 29 – Visualization of Results obtained from comparative analysis (Multistorey).

5 Conclusion and Future Work

5.1 Conclusion

The thesis primarily focused on automated model creation and detecting design changes in BIM models. As discussed in section 2.9 it is evident that there lies some gap in most of the current techniques used to detect model changes. Some lack the ability to present semantic information of the elements while others lack in detecting any new or missing elements. The thesis addresses all these gaps and solves them. Furthermore, as detailed in Chapter 4 with 2 Datasets and a multistorey structure, the comparison analysis to evaluate changes has been tested in various BIM Models and proven to be useful and convenient for the users. To run the framework, there is a requirement for carrying out the analysis. The BIM Models with building elements (doors, walls, etc.) should have well-defined parameters (length, width, etc.). There are two ways to build BIM Models, either the available models can be utilized for the analysis or points from technical drawings can be utilized to prepare an input file (spreadsheet or csv). This input file can be read by the scripts to create BIM Models, making it totally automated. Moreover, family instances can also be created automatically by making the scripts read the input file containing Host family type and the parameters (length, width, etc.) which the user wishes to duplicate. Moreover, visualization will be carried out simultaneously with different colour grading according to the status (moved, not moved, missing, etc.) of that element making it easy to identify the result visually. The proposed framework in the thesis provides a simple but quick and robust method to perform both Comparative Analysis and generate BIM Models from point inputs. The computational efficiency of the framework is fast and independent of the size of the model and number of elements. To make the run time even quicker, the user has the flexibility to choose a particular element for analysis at a time. The quantitative results are very straightforward and comprehensive and are also visualized to provide an overview of the scenario especially for the stakeholders. Altogether, it's evident that using the suggested framework can greatly accelerate the modeling process, thus enhancing the importance and adoption of BIM Models. However, there is always room for improvement; few spots for improvement have been discussed in section 5.2.

5.2 Future Work

Based on the discussed topics, some recommendations for future enhancement for this study has been prepared. For placement of the grids, Manhattan World assumptions have been used, therefore, the framework has been tested with linear walls, there might be accuracy issues when working with curved walls, it can be worked upon in future. Other parameters like material changes, etc can be added to the framework. The framework is currently available only for walls, columns, doors, and windows. There could be more elements like furniture, MEP, etc which could be added to the framework in the future.

References

[1] Takim, R., Harris, M., & Nawawi, A. H. (2013). Building information modeling (BIM): A new paradigm for quality of life within architectural, engineering and construction (AEC) industry. Procedia - Social and Behavioral Sciences.

[2] Arayici, et. al. (2012). Building information modelling (BIM) implementation and remote construction projects: issues, challenges, and critiques.

[3] Weygant, R. S. (2011). BIM content development: standards, strategies, and best practices. John Wiley & Sons.

[4] Eastman, C., Eastman, C. M., Teicholz, P. et al. (2011). BIM Handbook: A Guide to Building Information Modeling for Owner, Manager, Designer, Engineers and Contractors.

[5] Succar, B. (2009) Building Information Modelling Framework: A Research and Delivery Foundation for Industry Stakeholders. Automation in Construction.

[6] A. Borrmann, M. König, C. Koch, and J. Beetz, Eds. (2018). Building Information Modeling, Cham: Springer International Publishing, DOI: 10.1007/978-3-319-92862-3.

[7] Eastman, C., Teicholz, P., Sacks, R. & Liston, K. (2008). BIM handbook: A guide to building information modeling for owners, managers, designers, engineers, and contractors. John Wiley & Sons.

[8] NIBS (2012). National BIM Guide for Owners, https://www.nibs.org/files/pdfs/NIBS_BIMC_NationalBIMGuide.pdf, page 12

[9] Alencastro, J., Fuertes, A., & de Wilde, P. (2018). The relationship between quality defects and the thermal performance of buildings. Renew. Sustain. Energy Rev., 81, 883–894.

[10] Succar, B. (2009). Building information modelling framework: A research and delivery foundation for industry stakeholders. Automation in Construction, 18(3), 357-375.

[11] Kymmell, W. (2008). Building Information Modeling: Planning and Managing Construction Projects with 4D CAD and Simulations. McGraw-Hill Professional.

[12] Azhar, S., Hein, M., & Sketo, B. (2011). Building Information Modeling (BIM): Benefits, Risks and Challenges. Charleston, SC: Construction Research Congress.

[13] BIMForum (2017). Level of Development Specification Version 2017. BIMForum, USA. http://bimforum.org/lod/

[14] Structure Magazine - https://www.structuremag.org/wp-content/uploads/0813-is-1.jpg.

[15] Theresa Meyer, Ansgar Brunn, Uwe Stilla. (2022). Change detection for indoor construction progress monitoring based on BIM, point clouds and uncertainties, Automation in Construction, Volume 141.

[16] Park, Do-Young, Jun, Yeong-Jin, Kim, Yo-Han,Kim, Ju-Hyung, Kim, Jae-Jun. (2006). A study on the minimization of Design Change using BIM. Retrieved from https://www.irbnet.de/daten/iconda/CIB17320.pdf.

[17] ISO/TR 41013:2017 - Facility management -- Scope, key concepts, and benefits https://www.iso.org/.

[18] AUBEA (2017). Julie Jupp, Ramsey Awad, A Change Management Perspective on BIM-FM Implementation, https://easychair.org/publications/open/6lx6.

[19] Catenda - 3D BIM (Building Information Modeling), https://catenda.com/glossary/3d-modeling/

[20] 3D BIM Modeling service | Building Information Modeling, https://3dpointshot.com/bim-modeling.

[21] ALLPLAN Blog - News zu BIM & CAD, Parametric BIM Modeling - Efficiency And Flexibility In Planning Processes, https://blog.allplan.com/en/parametric-bim-modeling.

[22] Nandbox (2023), Visual Programming vs Traditional Programming: Full Guide, https://nandbox.com/visual-programming-vs-traditional-programming-full-guide/.

[23] Visual Programming - How Does It Work? Complete Guide for 2023, https://kissflow.com/low-code/visual-programming.

[24] An Intro to Visual Programming Language | OutSystems, https://www.outsystems.com/tech-hub/app-dev/understanding-visual-programming-language/.

[25] BIM Community - Parametric Design with BIM Visual Programming: BIMCommunity, https://www.bimcommunity.com/news/load/1596/parametric-design-with-bim-visual-programming.

[26] GitHub - wonderworks-software/PyFlow: Visual scripting framework for python - https://wonderworks-software.github.io/PyFlow.

[27] Dynamo BIM 2022, https://dynamobim.org/.

[28] Allplan - BIM - CAD - 3D Software, http://www.allplan.com/.

[29] Autodesk (2023). http://www.autodesk.com/.

[30] Novatr Prev OX. (2024). Dynamo For Revit: Everything You Need to Know, https://www.novatr.com/blog/dynamo-for-revit.

[31] H. Tran a, K. Khoshelham (2019). Building Change Detection through comparison of a Lidar Scan with a building information model, https://isprs-archives.copernicus.org/articles/XLII-2-W13/889/2019/isprs-archives-XLII-2-W13-889-2019.pdf.

[32] Indoor Change Detection (BIM as-build versus as-designed) (2013), https://www.ifp.uni-stuttgart.de/en/research/geoinformatics/indoor-change-detection/.

[33] Boujaoudeh Khoury K. (2019). Effective Communication Processes for Building Design, Construction, and Management. Buildings. 9(5):112. https://doi.org/10.3390/buildings9050112

[34] Viorica Pătrăucean, Iro Armeni, Mohammad Nahangi, Jamie Yeung, Ioannis Brilakis, Carl Haas. (2015). State of research in automatic as-built modelling, Advanced Engineering Informatics, Volume 29, Issue 2, ISSN 1474-0346, https://www.sciencedirect.com/science/article/pii/S1474034615000026

[35] James Ocean (2024), What is BIM Automation? The Importance of BIM Automation Tools, https://revizto.com/en/what-is-bim-automation/

[36] J. Monedero, "Parametric design: a review and some experiences," Autom. Constr., vol. 9, no. 4, Jul. 2000, doi: 10.1016/S0926-5805(99)00020-5.

[37] A. W. Stocking. (2009). "Generative design is Changing the face of architecture," Build. Des.

[38] Geng, Y. (2006) Study on Methods of Parametric Modeling of Jack-Up Platform Based on Catia. Master's Thesis, Dalian University of Technology, Dalian, China.

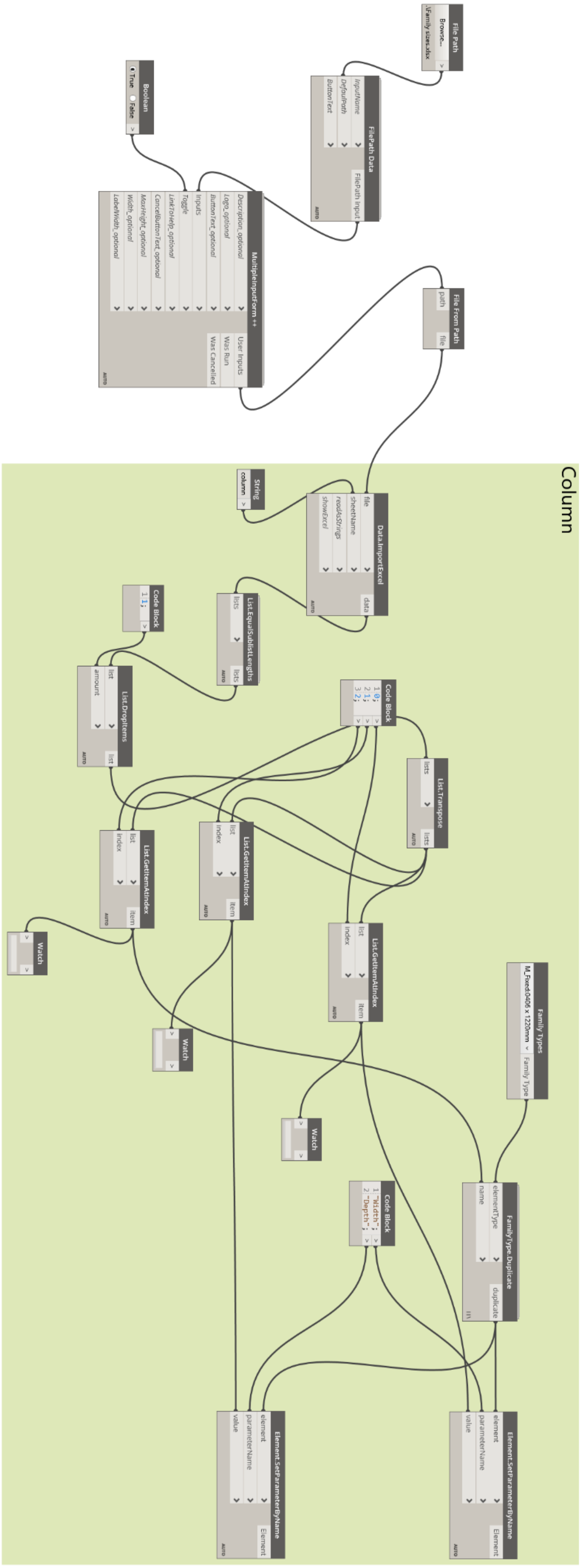
[39] Li, Y. (2018). Parametric Modeling and Mechanical Property Analysis of Jack-Up Platform. Master’s Thesis, China University of Petroleum (EastChina), Qingdao, China.

[40] Abdollahi, Ali & Arefi, Hossein & Malihi, Shirin & Maboudi, Mehdi. (2023). Progressive Model-Driven Approach for 3D Modeling of Indoor Spaces. Sensors. 23. 5934. 10.3390/s23135934.

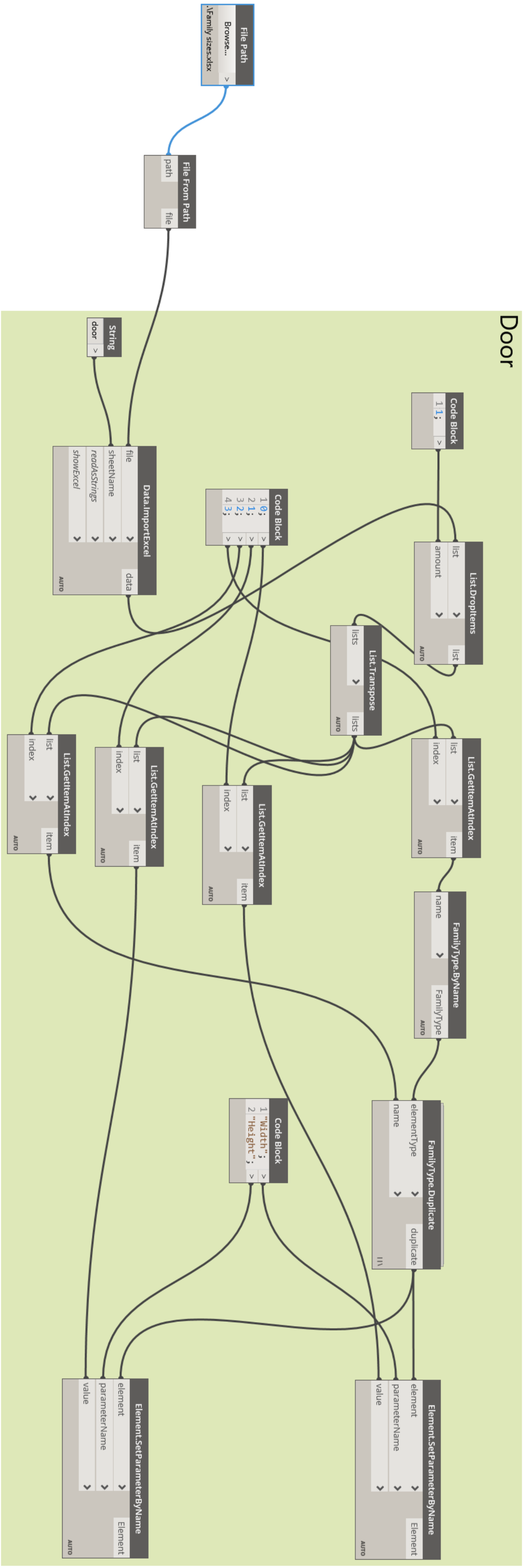
Appendix A – Dynamo Codes

The developed Dynamo codes used throughout the Master Thesis can be found in the depictions below.

Script for creating Column Insta



Script for creating Door Instances.

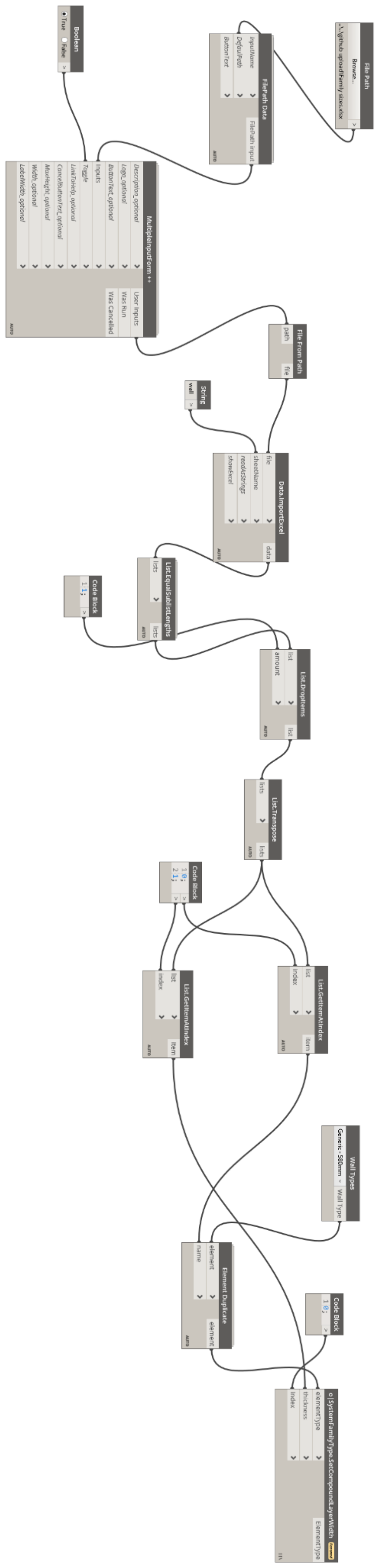


```
1;
```

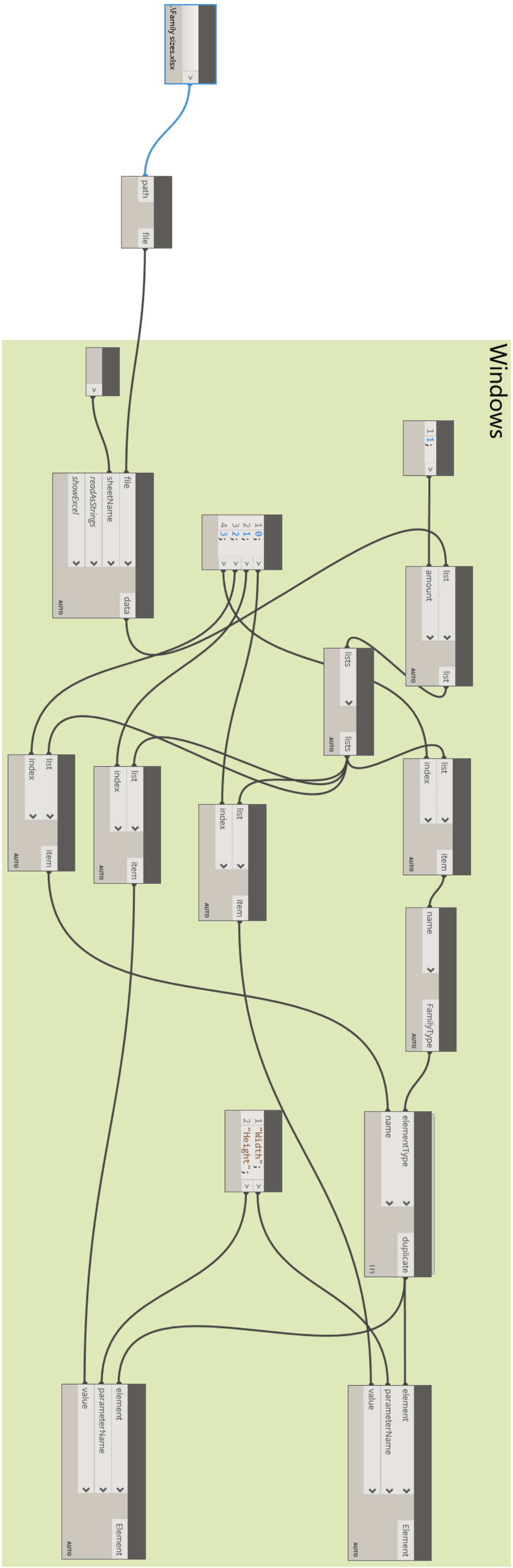
```
0;
1;
2;
3;
```

```
"Width";
"Height";
```

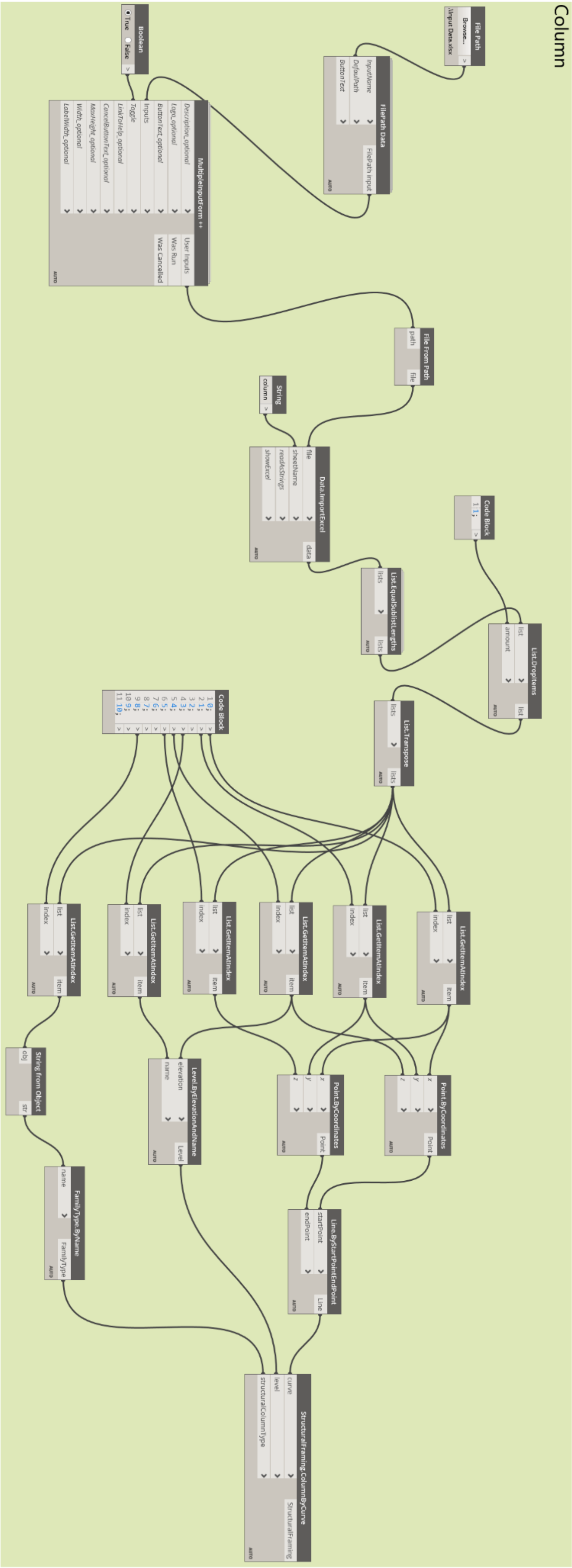
Script for creating Wall Instances.



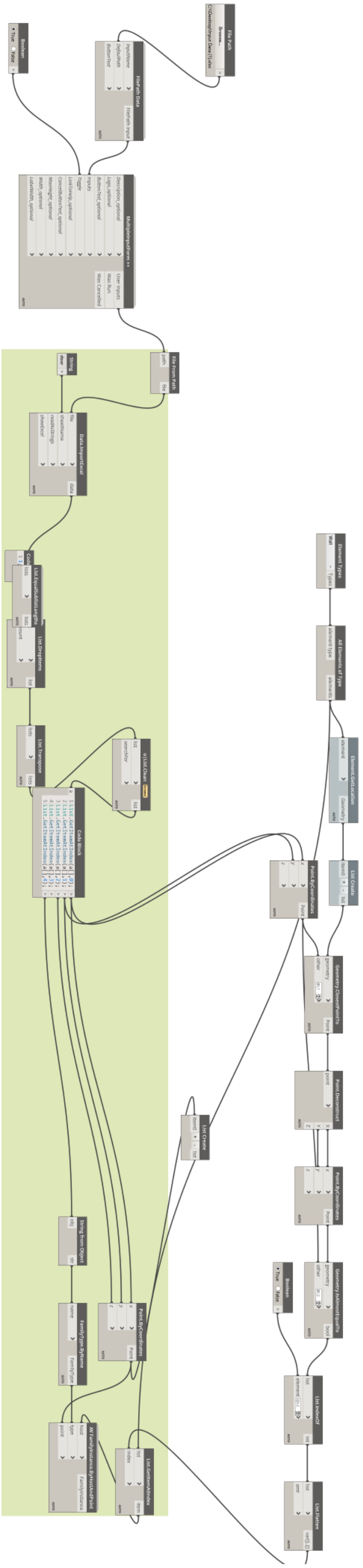
Script for creating Window Instances.



Script for generating 3D Columns.

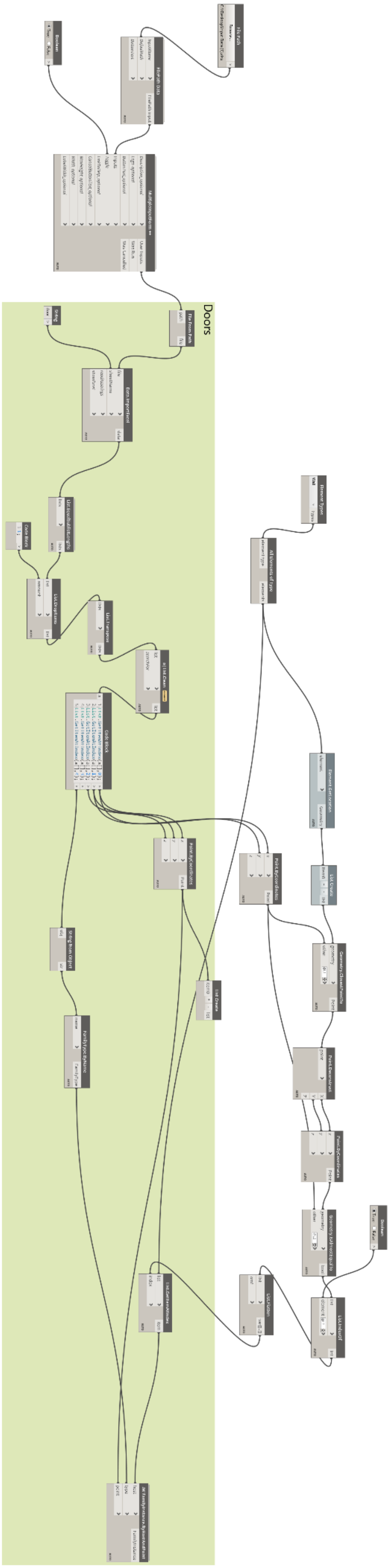


Script for generating 3D Doors.

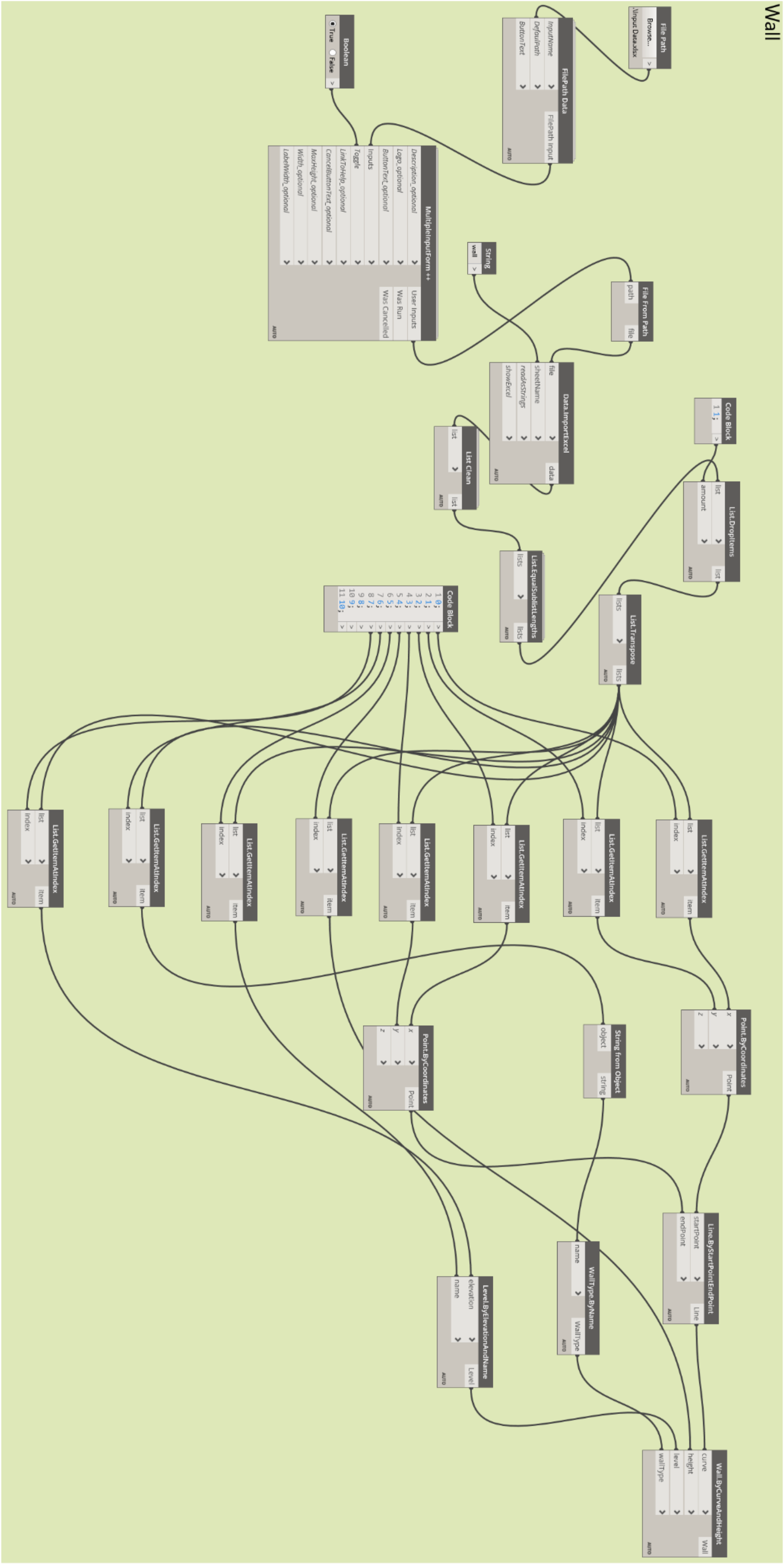


```
a
1 List.GetItemAtIndex(a[],0);
2 List.GetItemAtIndex(a[],1);
3 List.GetItemAtIndex(a[],2);
4 List.GetItemAtIndex(a[],3);
5 List.GetItemAtIndex(a[],4);
```

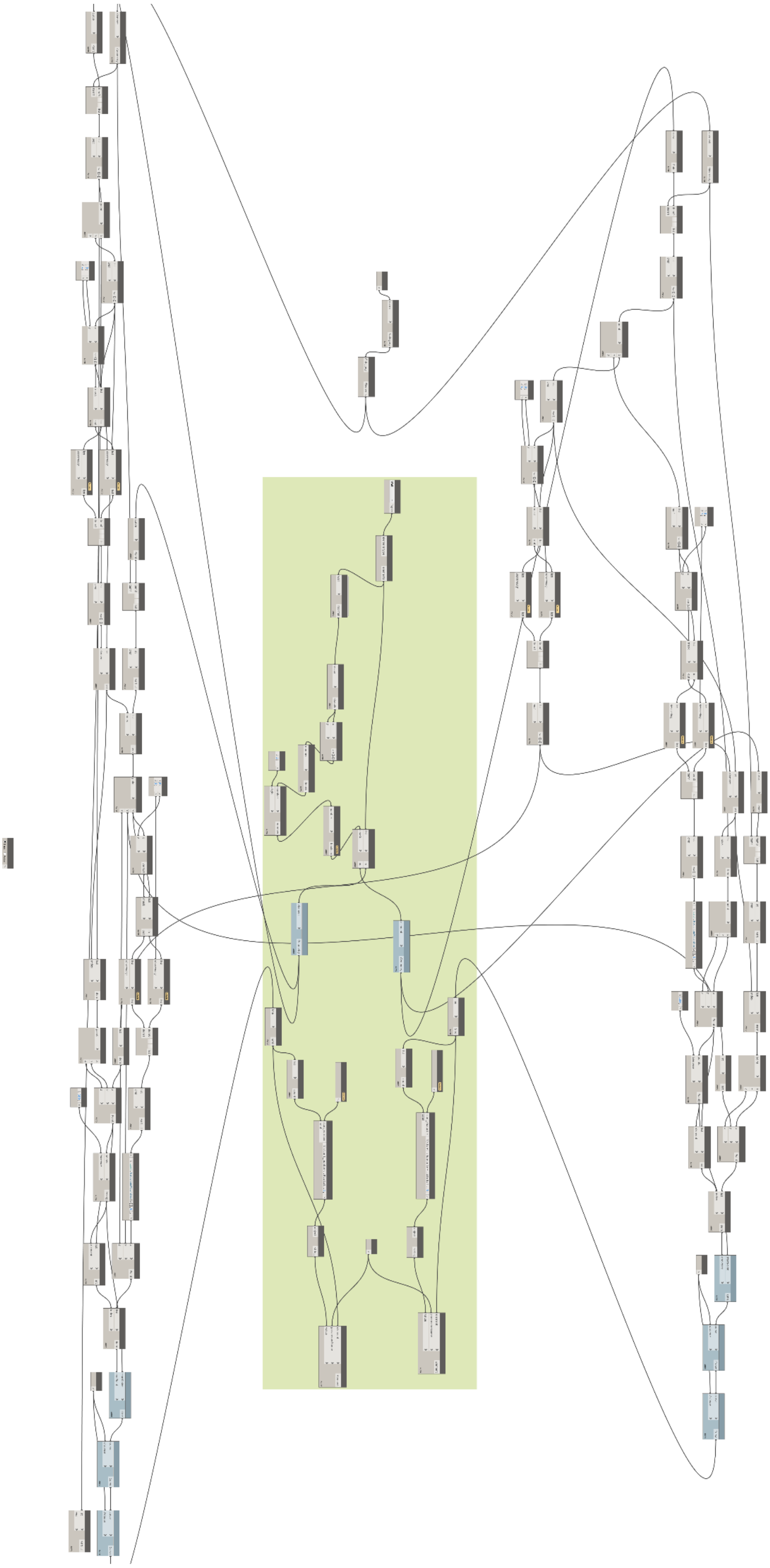
Script for generating 3D Windows.



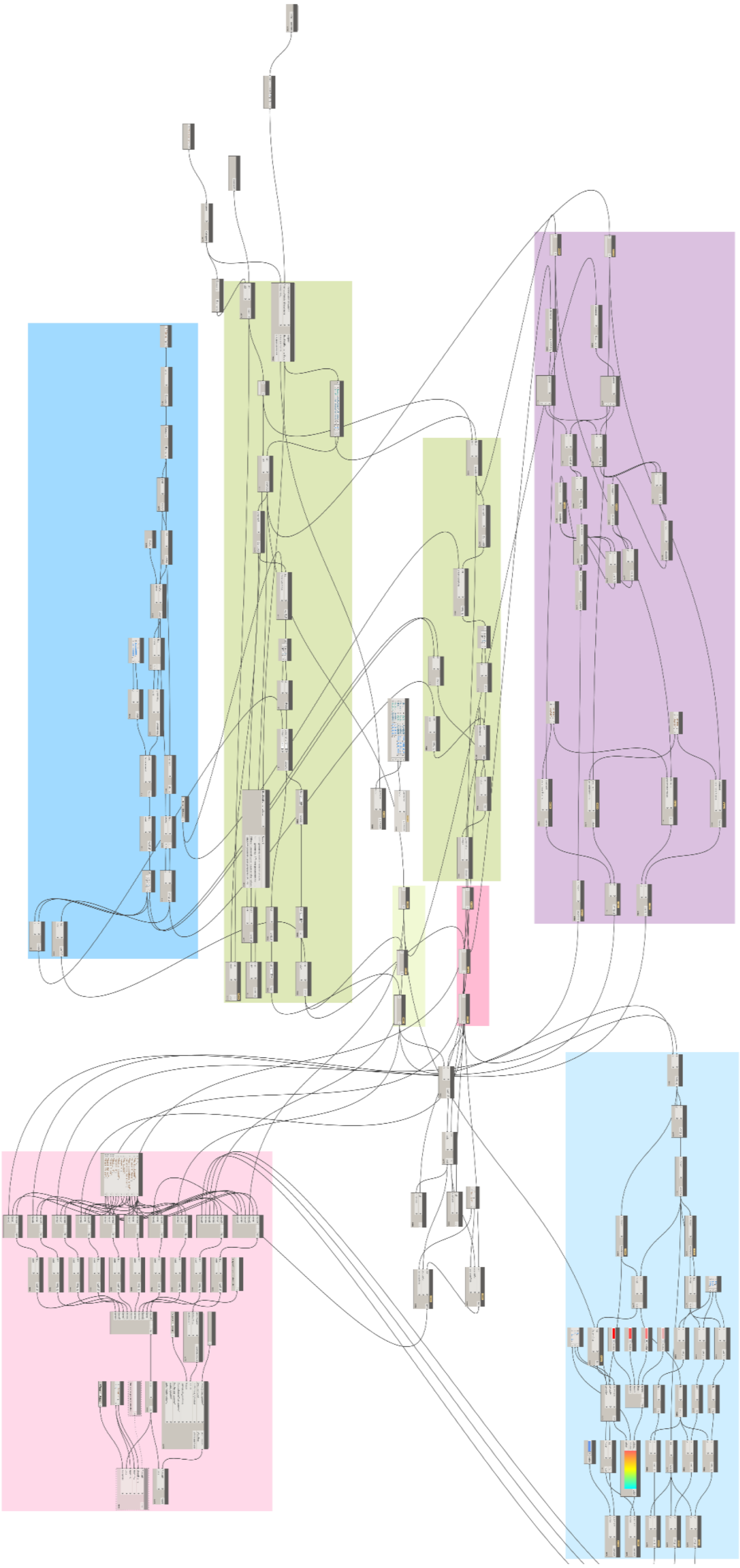
Script for generating 3D Walls.



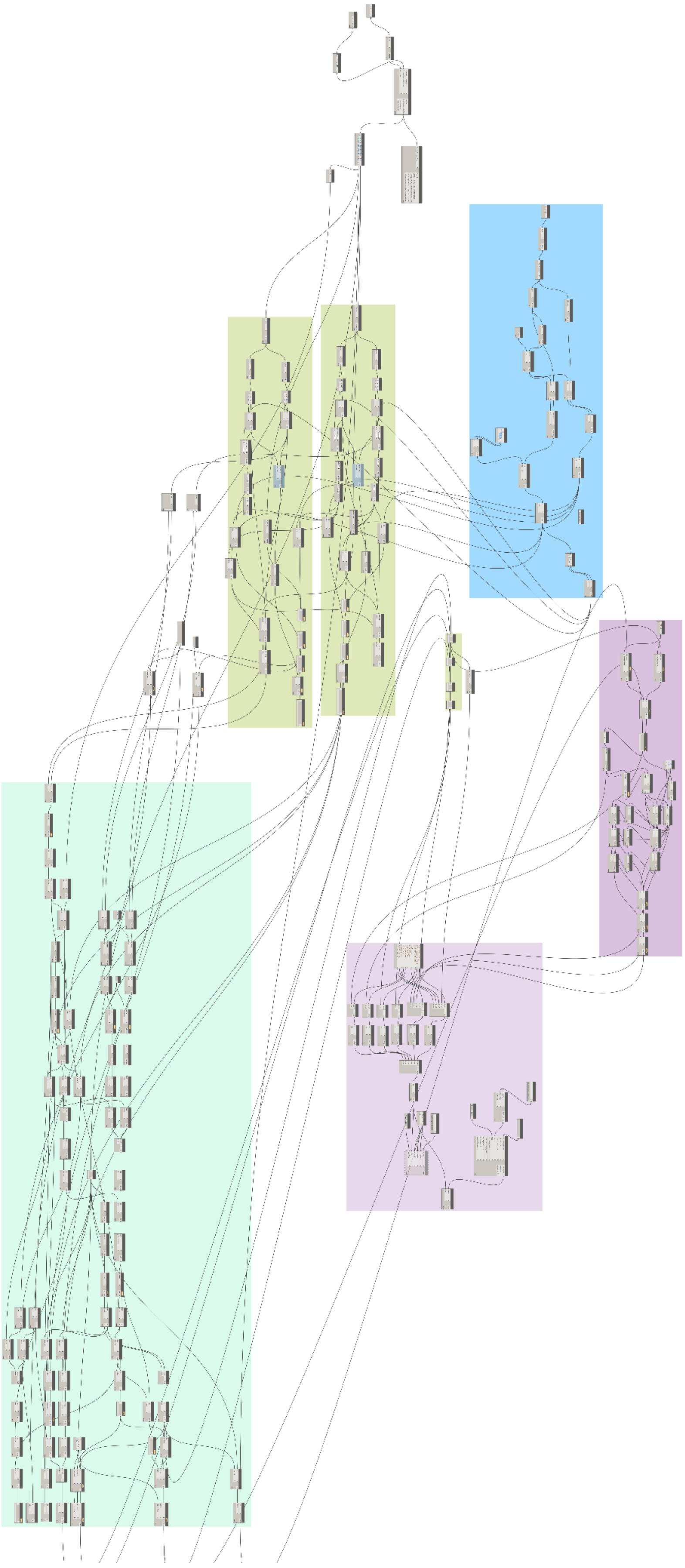
Script for generating Grids.



Script for Comparative Analysis of Columns.

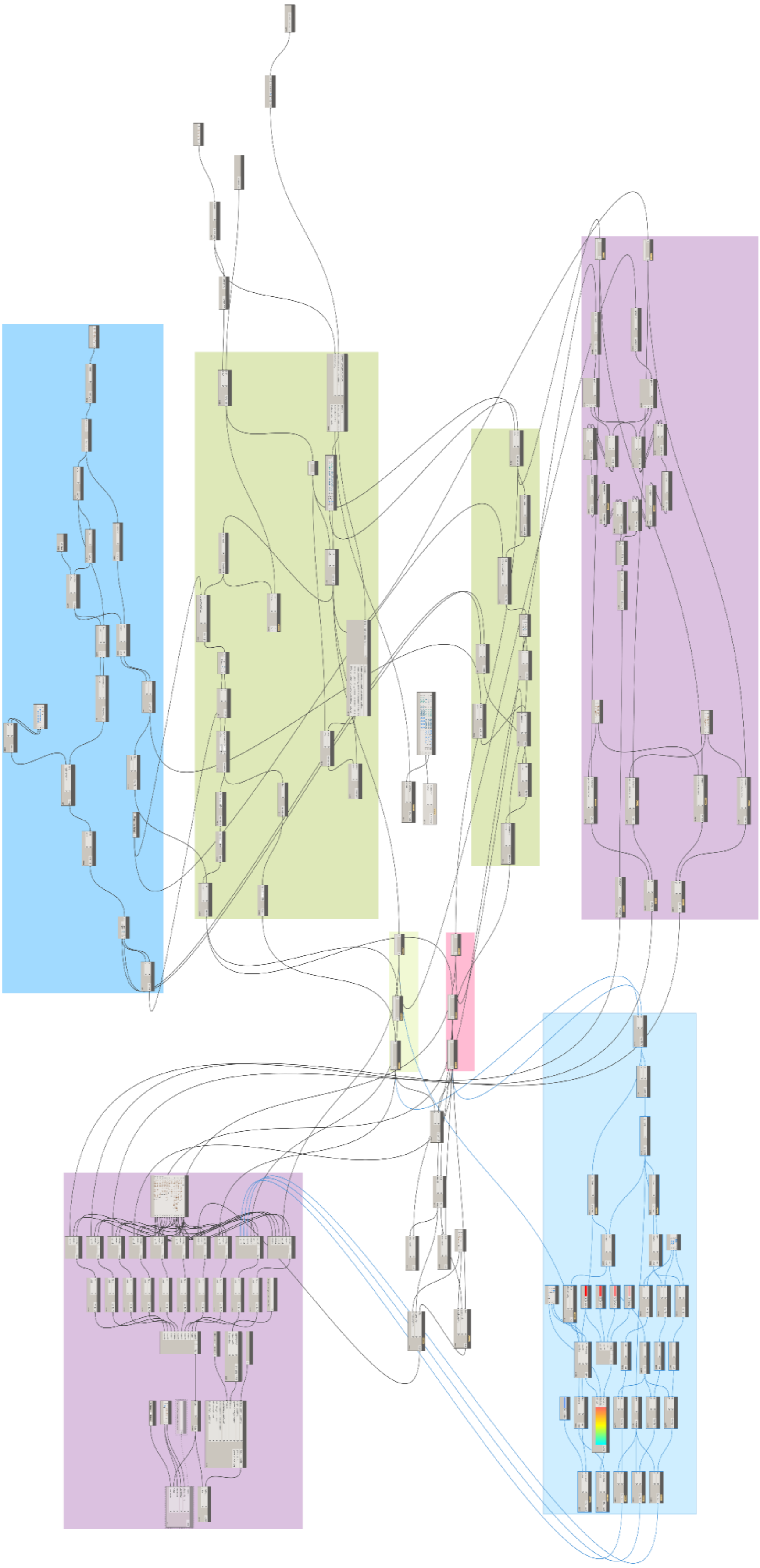


Script for Comparative Analysis of Walls.

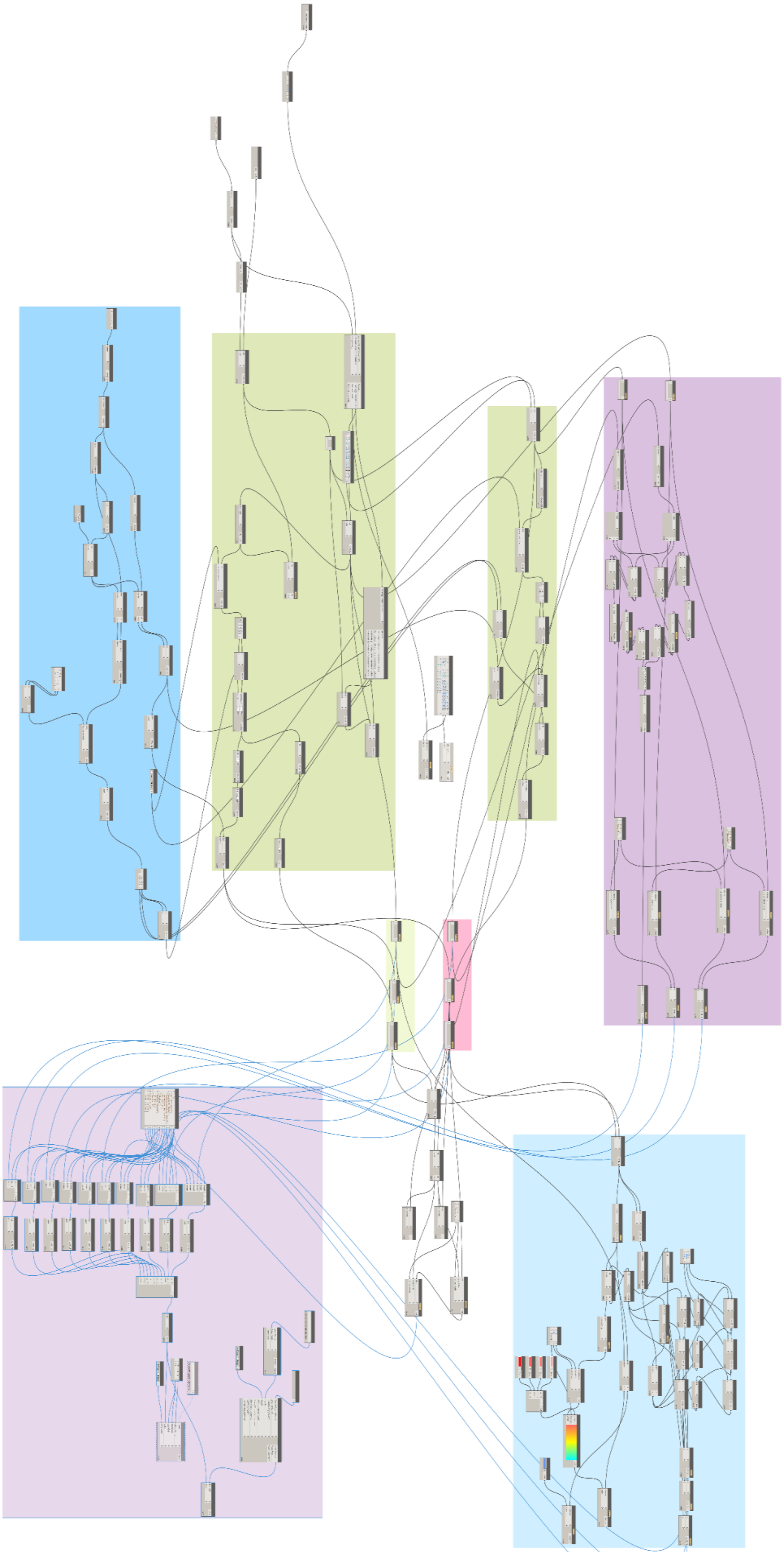


.

Script for Comparative Analysis of Doors.



Script for Comparative Analysis of Windows.



Below mentioned are the file paths to create Revit extensions: -

C:\Users\Admin\AppData\Roaming\CustomRevitExtension\3D_Elements.extension\3D_Elements.tab\Create Element Instances.panel\Create Instances.pulldown\Create Door Instances.pushbutton

C:\Users\Admin\AppData\Roaming\CustomRevitExtension\3D_Elements.extension\3D_Elements.tab\Generate 3D Elements.panel\Create Elelments.pulldown\Generate Door.pushbutton

C:\Users\Admin\AppData\Roaming\CustomRevitExtension\3D_Elements.extension\3D_Elements.tab\Generate Grids.panel\Generate Grids.pushbutton

C:\Users\Admin\AppData\Roaming\CustomRevitExtension\3D_Elements.extension\3D_Elements.tab\Visualization.panel\Visualize Elements.pulldown\Detect Door Changes.pushbutton

Declaration of Originality

With this statement, I declare that I have independently completed this master's thesis. The thoughts taken directly or indirectly from external sources are correctly marked as such. This thesis was not previously submitted to another academic institution and must be published.

Munich, 15th May 2024 Zaid Memon