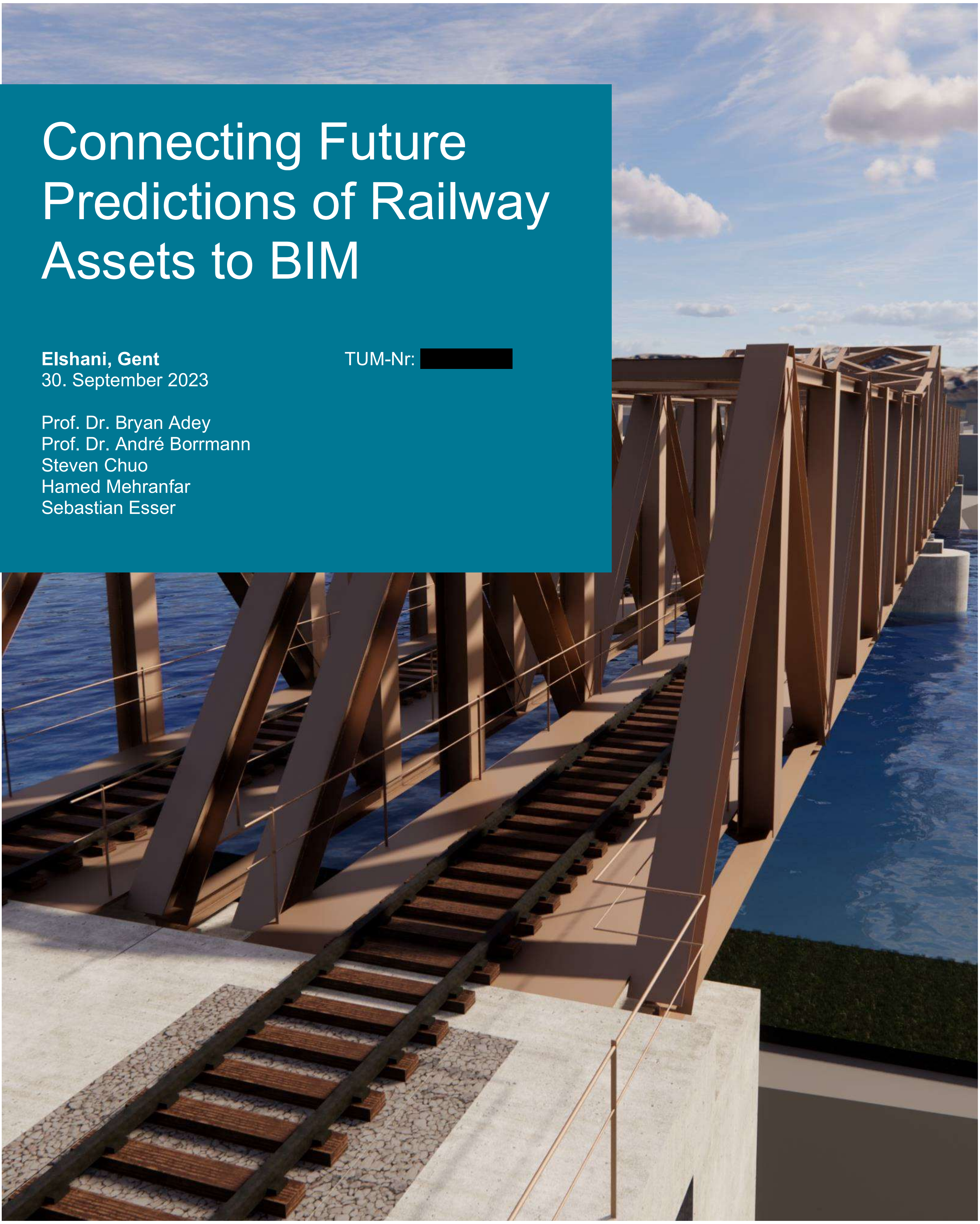
Chair of Infrastructure Management
Chair of Computational Modeling and Simulation



Connecting Future Predictions of Railway Assets to BIM

Elshani, Gent
30. September 2023

TUM-Nr: [redacted]

Prof. Dr. Bryan Adey
Prof. Dr. André Borrmann
Steven Chuo
Hamed Mehranfar
Sebastian Esser

Institut für
Bau- und Infrastruktur-
management

Abstract

The construction industry's ongoing shift towards digitalization is leading to a higher adoption of Building Information Modeling (BIM) in various construction projects, including those related to infrastructure. A notable development in this context is the growing emphasis on advanced asset management techniques, particularly the application of predictive maintenance to proactively estimate the optimal timing for maintenance activities. This holds significant relevance in the aspect of railway infrastructure due to the substantial impact of line closures. The connection between predictive maintenance data, including the condition state and planned interventions for these railway assets, and their integration into digital workflows is investigated. Furthermore, ways of how to visualize the data are proposed. Firstly, the establishment of digital data integration at the asset level is addressed. Emphasis is placed on exploring data representation through visualization to facilitate a seamless incorporation of BIM into the workflows of bridge asset managers. This approach provides them with a deeper insight into the current and future condition of their assets through visual information, thereby enhancing the decision-making process. Furthermore, the creation of a unified data source at the network level is provided. This involves the consolidation of data from bridge and track assets within a single view, with a specific focus on identifying synergies and associations. Moreover, it examines the consistent integration of BIM and Geographic Information Systems (GIS), enabling managers to collaborate, access real-time information, and share insights during the planning and execution of interventions. This collaborative approach ultimately reduces the potential for downtime during operational phases. The final part of this thesis centers on enhancing the utilization of digital models in subsequent project phases, particularly during the construction phase. It places importance on the utilization of 4D BIM models, discussing various strategies and addressing concerns related to data exchange. The advantages of employing a shared model or a common data environment to enhance the utility of BIM models are highlighted. In summary, the integration of digital workflows, with a specific focus on BIM, has the potential to significantly enhance the decision-making processes of asset managers. This can lead to more optimal decisions, reduced risks, and a decrease in unexpected costs and scheduling delays, ultimately improving the overall value of construction projects.

Keywords
BIM; Predictive Maintenance; Visualization; Asset Management; SSoT; Railway Infrastructure

Citation preference
IEEE

Zusammenfassung

Die fortlaufende Verschiebung der Bauindustrie hin zur Digitalisierung führt zu einer verstärkten Anwendung des Building Information Modeling (BIM) in verschiedenen Bauprojekten, einschließlich solcher im Zusammenhang mit der Infrastruktur. Eine eindrucksvolle Entwicklung in diesem Kontext ist die wachsende Relevanz fortschrittlicher Asset-Management-Techniken, insbesondere die Anwendung von Predictive Maintenance, um den optimalen Zeitpunkt für Wartungsarbeiten vorbeugend abzuschätzen. Das genannte Konzept gewinnt in Bezug auf die Eisenbahninfrastruktur aufgrund der erheblichen Auswirkungen von Streckensperrungen besondere Relevanz. Die Zusammenhänge zwischen Daten zur Predictive Maitenance, vor allem des Zustands und der geplanten Eingriffe für diese Eisenbahnkomponenten, und deren Integration in digitale Arbeitsabläufe wird sorgfältig untersucht. Darüber hinaus werden Möglichkeiten zur Visualisierung dieser Daten vorgeschlagen. In erster Linie wird die Etablierung der digitalen Datenintegration auf der Komponentenebene behandelt. Dabei wird großer Wert auf die Erkundung der Datenrepräsentation durch Visualisierung gelegt, um die kontinuierliche Integration von BIM in den Arbeitsprozessen von Brücken-Asset-Managern zu erleichtern. Dieser Ansatz bietet einen klareren Einblick in den aktuellen und zukünftigen Zustand der Komponente durch visuelle Informationen, wodurch der Entscheidungsprozess erheblich verbessert wird. Des Weiteren erfolgt die Erstellung einer einheitlichen Datenquelle auf Netzwerkebene. Es umfasst die Konsolidierung von Daten aus Brücken- und Streckenkomponenten in einer singulären Ansicht, wobei ein spezieller Schwerpunkt auf der Identifizierung von Synergien und Verknüpfungen liegt. Darüber hinaus wird die durchgängige Integration von BIM und Geographic Information Systems (GIS) untersucht, um Managern die Möglichkeit zu bieten, während der Planung und Ausführung von Eingriffen zusammenzuarbeiten, auf Echtzeitinformationen zuzugreifen und Erkenntnisse auszutauschen. Dieser kooperative Ansatz trägt letztendlich dazu bei, die potenzielle Betriebsausfallzeit zu minimieren. Der abschließende Aspekt dieser Arbeit konzentriert sich auf die Verbesserung der Nutzung digitaler Modelle in folgenden Projektphasen, insbesondere während der Bauphase. Hierbei wird besonderer Wert auf die Nutzung von 4D-BIM-Modellen gelegt, wobei verschiedene Strategien behandelt werden, um den Zusammenhang mit dem Datenaustausch anzusprechen. Die Vorteile der Verwendung eines gemeinsamen Modells oder einer gemeinsamen Datenumgebung zur Verbesserung der Nützlichkeit von BIM-Modellen werden betont. Zusammenfassend lässt sich aufweisen, dass die Integration digitaler Arbeitsabläufe, hauptsächlich im Bereich BIM, das Potenzial hat, die Entscheidungsprozesse von Asset-Managern erheblich zu verbessern. Dies kann zu optimierten Entscheidungen, reduzierten Risiken sowie geringeren unerwarteten Kosten und Verzögerungen im Zeitplan führen und somit den Gesamtwert von Bauvorhaben nachhaltig steigern.

Acknowledgements

I would like to express my sincere gratitude to the Chair of Infrastructure Management at ETH Zurich, and in particular, Prof. Bryan Adey, for their support and guidance throughout my research journey.

I am also deeply thankful to my dedicated supervisors, Steve and Hamed, who have been with me every step of the way, offering their wisdom and encouragement.

Furthermore, I extend my appreciation to the Chair of Computational Modeling and Simulation, led by Prof. Andre Borrmann, and my supervisor, Sebastian. Engaging in insightful discussions and receiving support has enriched my research experience.

Recognition is given to Andreas, whose prior research has served as a source of inspiration and motivation.

Special mention is made of the BIM team at the SBB, with Jens and Roberto, for their willingness to share valuable information and engage in meaningful discussions.

Lastly, heartfelt appreciation is extended to my family, partner, and friends for their unwavering support, not just during the past six months but throughout the entirety of my journey, by being a continuous source of strength and motivation for me.

Contents

List of Tables

List of Figures

List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
API	Application Programming Interface
BCF	BIM Collaboration Format
BIM	Building Information Modeling
BMDV	Bundesministerium für Digitales und Verkehr (Federal Ministry for Digital and Transport)
CAD	Computer-Aided Design
CDE	Common Data Environment
LV	Landesvermessung (State surveying)
CS	Condition State
CSG	Constructive Solid Geometry
CSV	Comma-Separated Values
DB	Database
DWG	Drawing
DXF	Drawing Exchange Format
EPSG	European Petroleum Survey Group
ESRI	Environmental Systems Research Institute
FBX	Filmbox
FDK	Fachdatenkatalog (Subject Data Catalog)
FME	Feature Manipulation Engine
GeoJSON	Geographic JavaScript Object Notation
GIS	Geographic Information System
GML	Geographic Markup Language
GPX	Global Positioning System Exchange Format
IDE	Integrated Development Environment
IFC	Industry Foundation Classes
IoT	Internet of Things
JSON	JavaScript Object Notation
km	Kilometer
KPX	Keyhole Markup Language Placemark Extension
LOD	Level of Development
LOG	Level of Geometry
LOI	Level of Information
m	Meter
NURBS	Non-Uniform Rational B-Spline
NWD	Navisworks Document
OOM	Object Oriented Modeling
OSM	OpenStreetMap
OWL	Ontology Web Language
PDF	Portable Document Format
PM	Predictive Maintenance
SAT	Standard Advanced Computerized Implementation of Standards Text
SBB	Schweizerische Bundesbahnen (Swiss Federal Railways)
SHM	Structural Health Monitoring
SQL	Structured Query Language
SSoT	Single Source of Truth
STL	Standard Tessellation Language
UI	User Interface
WGS	World Geodetic System

A dissertation submitted for the degree of Master of Science in Civil Engineering

Technical University of Munich (TUM)

Connecting Future Predictions of Railway Assets to BIM

Author Gent Elshani
[redacted]

[redacted]

Supervision Prof. Dr. Bryan T. Adey, Professor of Infrastructure Management, Swiss Federal Institute of Technology (ETH Zurich)

Prof. Dr. André Borrmann, Professor of Computational Modeling and Simulation, Technical University of Munich (TUM)

Munich, September 30, 2023

1 Introduction

Automation and digitalization have become of utmost importance in modern infrastructure management, revolutionizing how we design, build, as well as maintain existing and new systems. The importance lies in their ability to enhance efficiency, reduce operational costs, and optimize intervention planning, to name a few. In this way, the overall infrastructure stability is improving. Recent advancements involve integrating new technologies such as big data analytics, the Internet of Things (IoT), and Building Information Modeling (BIM) into infrastructure management. These innovations assist real-time monitoring and predictive maintenance, helping infrastructure managers identify and address issues before they escalate, minimizing downtime and maximizing the asset lifespan. Ultimately, these technologies help achieve more reliable and efficient decision-making processes, promoting optimized workflows for infrastructure managers.

This Master's Thesis focuses on integrating results from predictive maintenance performed on railway assets into digital models created and organized following the principles of BIM. The key emphasis revolves around the management of railway bridges and tracks. The proposed methodology, however, can also be extended analogously to other infrastructure assets.

A complete guide on how to use digital tools to improve the daily tasks of asset managers, including bridge, track, corridor, and network managers, is provided in this report. A more extensive exploration of how these methods can assist the proposed intervention planning process, designed to take place at the Swiss Federal Railways (SBB), as well as a case study on one part of their railway assets, is presented here.

1.1 Intervention planning process

This Thesis is part of a more extensive project called STABILITY. The project aims to achieve a more reliable intervention planning and execution in infrastructure, as well as to expand the existing network so that they will be carried out in precisely allocated time slots. This will be achieved through the integration of digital tools into the processes of the SBB [1].

As part of the STABILITY project, the Intervention Planning Process has been developed through extensive research and a number of interviews with the SBB. It includes numerous processes and several domains with the aim of optimizing the procedures to tackle the mentioned obstacles. The whole Intervention Planning Process is depicted in Figure 1, divided into several sectors indicated by the coloring [2]. Nevertheless, the included procedures are planned to be executed significantly in advance, spanning a period of 15 to 20 years before the actual interventions, focusing on the decision-making and planning part of the interventions.

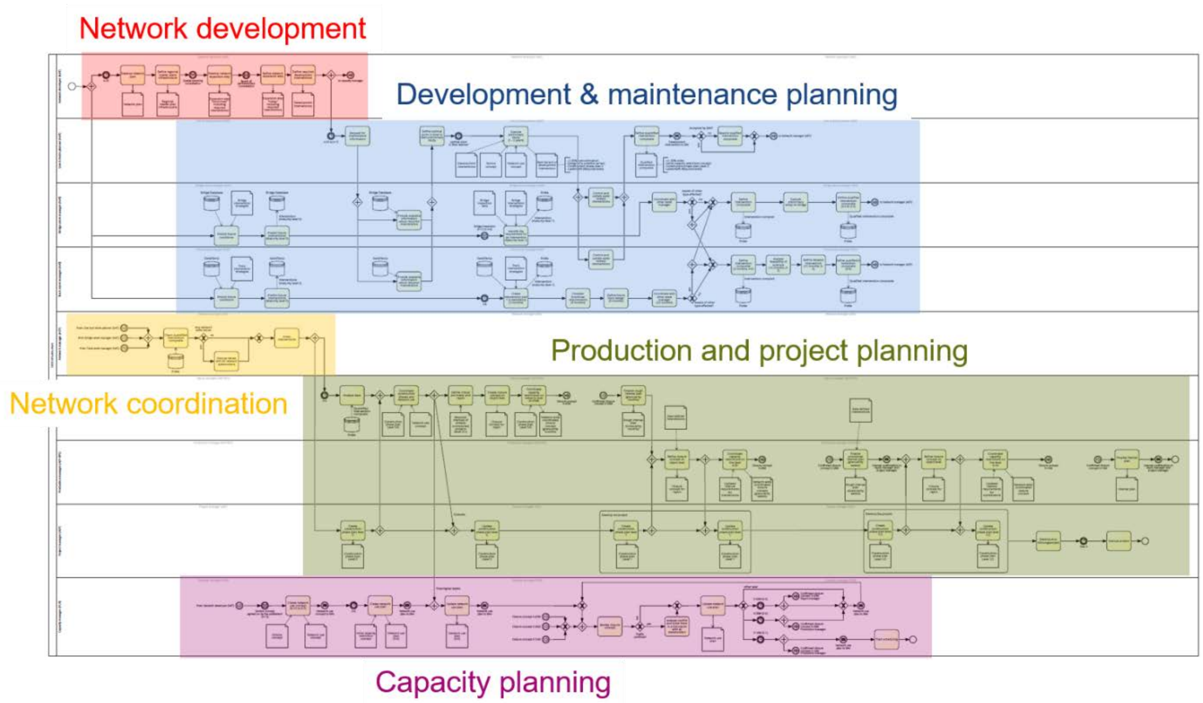

Network development
Development & maintenance planning
Production and project planning
Network coordination
Capacity planning

Figure 1: The Intervention planning process and its consisting parts [2].

In the scope of this Thesis, the emphasis mostly remains on the development and maintenance planning, which addresses existing infrastructure and includes algorithms for predicting condition states and interventions on the given assets. As part of the Network coordination, a wider-scaled view has also been provided. Furthermore, concepts on how the proposed digital tools might influence and be useful in the production and project planning, particularly within the context of construction plans, have been covered.

How all these concepts can harness the potential of BIM, especially digital 3D models, and how the connection between the given data can be made and visualized is explained in more detail in the following sections of this report. A more comprehensive approach, concerning mostly the bridge assets, is covered in Section 2, whereas Section 3 addresses a combined approach of bridges and tracks at a network level.

1.1.1 Case study

As part of the STABILITY Project, particularly in this Thesis, a case study has been conducted to investigate a railway corridor. A total of 29 bridge sites and a grand total of 39 individual bridges, as well as roughly 35 km of railway tracks have been analyzed. Four of the bridges have been further analyzed, including bridges constructed from different construction materials and methods.

The SBB online database served as a basis for the bridge assets of this case study because of the consistent distinction and identification of the assets. All the bridges have been integrated in the analysis, including 3D model generation, with the exception of two bridges for which a digital representation has not been created because of the lack of data regarding the condition state and intervention predictions on those assets. Furthermore, a separation regarding the building material of the bridges has been performed, dividing two bridge sites into two distinct models since in both cases a bridge built from concrete and one from masonry was found.

The information about the materials of the bridges was obtained from a separate database. This additional source of information from the SBB contained information about all the railway assets, including the material and other data concerning, for instance, life expectancies.

The input information available to assess the bridges was in the form of construction drawings, reports, and on-site evaluations of the assets. Since these details were available only for four bridges on the line, the models have been treated with greater attention, resulting in more precise and comprehensive digital models compared to the other assets. Table 1 summarizes the information concerning the length of the four mentioned bridges.

Table 1: Information on the detailed bridges of the line.

Name	Length (m)
Example concrete bridge	48.05
Example masonry bridge	48.05
Example steel bridge	113.16
Example composite bridge	16.901

1.2 Building Information Modeling

With recent advancements in the fields of information technology and data management, the advantages of digital workflows and Building Information Modeling in engineering processes have become more accessible and impactful. BIM is a comprehensive digital representation of a structure's physical and functional characteristics. It serves as a central source of information throughout a project's lifecycle. Regarding the definition of BIM, multiple interpretations exist based on the time, field, and institute defining it [4]. According to the American Institute of Architects (AIA), BIM is defined as a model-based technology linked with a database of project information [5]. This innovative approach allows stakeholders to collaboratively plan, visualize, and simulate every aspect of a project from design to the actual construction of the assets. With real-time access to BIM data, project teams can make informed decisions, detect clashes or discrepancies early on, and optimize construction processes. It has been proven that digital workflows provide a more efficient approach through the different phases of a built structure by conserving the project information and avoiding data loss through the different intervals (refer to Figure 2) [6].

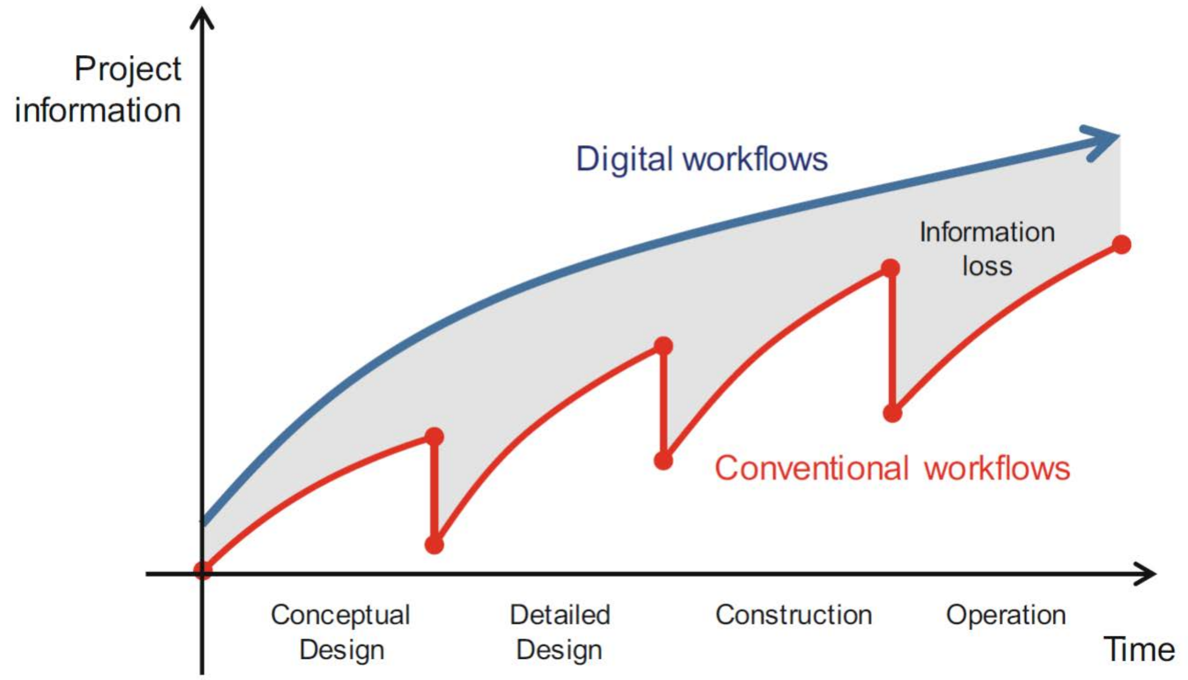


Figure 2: Project information through different phases of digital and conventional workflows [6].

As a result of these recent advancements, most industries have experienced an increase in productivity over the past few decades. A prior study revealed that, across various industries (excluding the construction sector), an average rise of approximately 85 percent in the value achieved from a process relative to the effort expended has been observed over the years (see Figure 3). Despite these developments, the construction industry has demonstrated a significant deficiency, with its resulting productivity remaining almost constant during the same time frame [7].

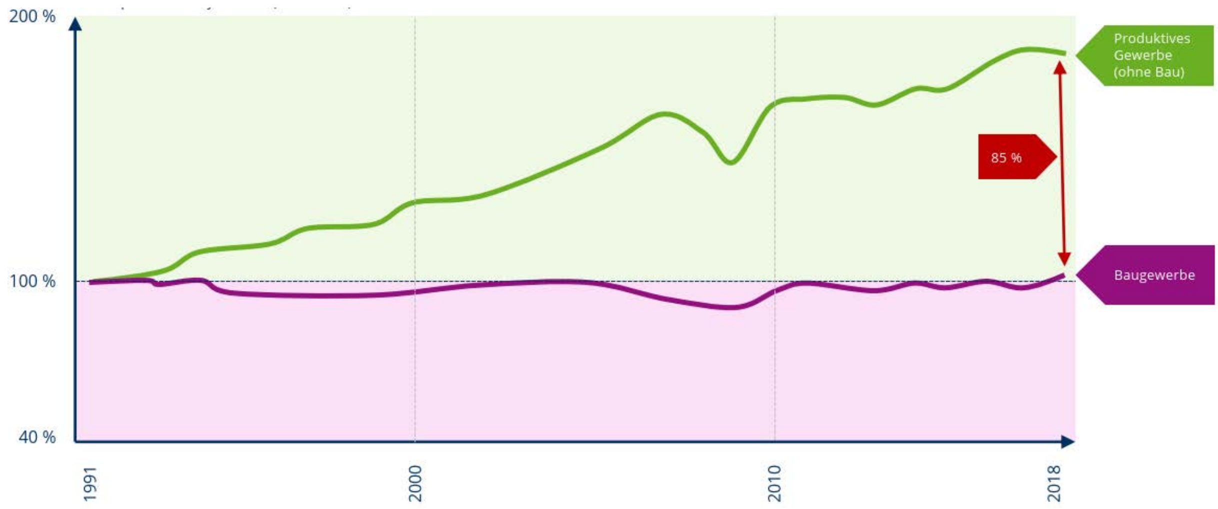


Figure 3: Productivity of other industries compared to the construction sector [7].

In order to tackle this issue, a lot of advanced technologies have emerged in the field of asset management, such as Structural Health Monitoring (SHM) and Predictive Maintenance (PM). A connection of the data generated from these processes to BIM exhibits a fundamental need to optimize the workflows.

This Thesis focuses on implementing and demonstrating a BIM-based workflow, prioritizing the different aspects of this digital method. Building Information Modeling relies on creating digital 3D models and continuous communication and enrichment of these given models with information from the stakeholders utilizing it. A key part of understanding this method depends on perceiving it not only as a 3D model with attributes but also as a network of integrated data exchange where all the different collaborators supply a common environment with relevant data that will be required for specific use cases [7]. Considering this approach, not always the creation of geometric models will be necessary since the impact and value of pure information models is also significant.

Integral to the BIM collaboration technique is its usage and acceptance on a bigger scale within the whole built environment, involving professionals and experts in the field. One way of approaching this problem is by requesting a digital model when accepting the offer for a construction project funded by public authorities. This is especially relevant in the infrastructure sector since the investor is, in most cases, a government or municipality. Some examples at the local and regional levels, especially in infrastructure projects, where this approach has been implemented include Swiss Federal Railways (SBB), with their BIM@SBB initiative [8], and the Federal Ministry of Transport and Digital Infrastructure (BMDV) in Germany, with their strategies for digital infrastructure with BIM4INFRA2020 [9] and specifically for railways with BIM4RAIL [10]. In a more global sense, an advancement in the usage of the BIM methodology has also occurred, with Figure 4 showing an increase of 31 percent in digital model creation and usage in a time frame of 6 years [11].

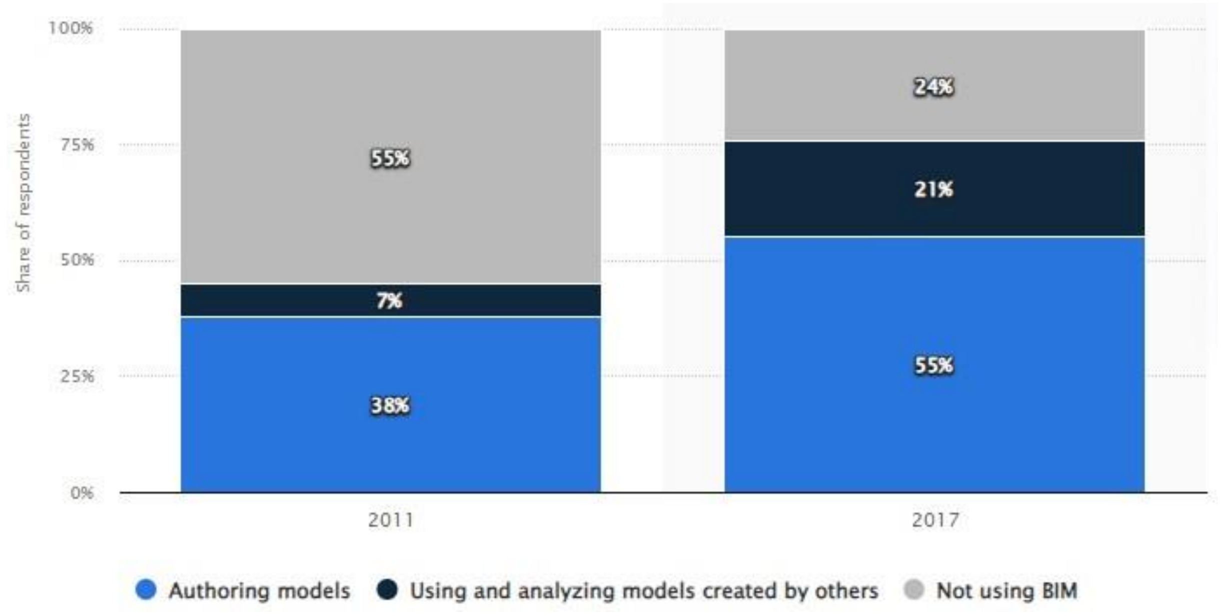


Figure 4: The percentage of professionals using digital models in their workflows [11].

Projects, such as the InterCity Railway in the eastern part of Norway, have proven that a model-based delivery ensures higher efficiency and reliability throughout different phases. With certain parts of the project already finished and an estimated overall completion target set for 2040, a full integration of BIM is planned to facilitate various operations [12], [13].

Numerous studies have demonstrated that the usage of BIM leads to an optimized workflow, resulting in improvements across several aspects of project delivery, execution, and maintenance.

Advancements in change of orders, faster project design, better decision-making, rework reduction, quality improvement, effective construction processes, and reduced cost of the construction projects are found [14], [15], [16], [17].

1.2.1 BIM and asset management

Building Information Modeling (BIM) plays an important role in advancing asset management by providing a wide-ranging and data-rich digital representation of a built environment. One of the key aspects that significantly improves through BIM integration is the ability to enhance interventions on assets and operations. BIM models store detailed information about every element. This wealth of data enables asset managers to create efficient maintenance plans, predict when assets might require restoration or replacement, and effectively allocate resources. It results in reduced downtime, extended asset lifespan, and lower operational costs.

In the field of infrastructure, BIM serves as a powerful tool for enhancing the efficiency of managing transportation networks, utilities, and public infrastructure systems. This gets achieved by facilitating data integration and visualization, allowing asset managers to integrate different data sources and gain a holistic, geospatial view of their infrastructure assets. BIM also empowers predictive maintenance, leveraging historical and real-time data to gain information about asset conditions and schedule maintenance, reducing the risk of unexpected occurrences. The interactive analyses in this field, which yield to qualitative results after implementation, should be mirrored as well when integrating BIM into the workflows.

In the realm of scientific discourse, there has been ongoing research aimed at exploring ways to establish connections between these concepts. BIM-based asset management systems are continuously being explored and developed. By taking into account over 400 academic and industry references, optimizations on current workflows, as well as limitations on the integration of BIM in the design, construction, and asset management have been investigated [18]. In order to additionally create these assets in the infrastructure domain, automation can be utilized, resulting in improved results and higher adoption of different BIM solutions [19].

While the implementation of BIM in asset management, particularly in the context of infrastructure and predictive maintenance, presents several advantages, there are also some limitations that must be overcome for it to be efficiently implemented into already existing workflows. One of the biggest challenges lies in data integration and compatibility. BIM may not seamlessly integrate with existing asset management systems, leading to data silos and interoperability issues. This can constrain the efficient utilization of data from various sources.

Data quality and accuracy are very crucial in asset management. If data integrity is endangered during the modeling phase or if updates are not consistently applied, the reliability of asset management systems can be undermined. Additionally, the accuracy of predictive maintenance models heavily relies on high-quality data, making data quality a critical concern. Particular attention should be directed towards addressing the uncertainty of the results, as these processes often yield estimations that do not align with the accurate information needed for integration into BIM systems. Finally, BIM may not provide real-time data, which are often crucial for predictive maintenance. Integrating real-time sensor data with BIM can be challenging and may necessitate additional systems.

1.3 Current state of the art

Researchers have investigated various approaches of integrating digital methods into infrastructure management operations, exploring diverse strategies for incorporating this methodology. Emphasis is placed on the integration of BIM into railway-specific systems, given the lack of widespread BIM establishment within the broader infrastructure sector.

Digital integration offers several advantages, including the ability to provide decision support by enabling simulations and visualizations, helping more accurate decision-making in the early stages of projects. Cost optimization is introduced by integrating the budget dimension and ensuring consistency of information, which provides an assistance for the management overall. In order to harness these advantages, it is essential to attain standardization in rail project management, encompassing its various phases. Furthermore, it's crucial to integrate rail-specific libraries into digital tools and software, as indicated by prior research [20].

The increasing importance of exploring effective communication methods is also under research. A previous study identified the adoption of concepts such as open BIM, especially with a focus on Industry Foundation Classes (IFC) and similar formats, as means to optimize solutions in the existing railway infrastructure, facilitating collaborative work among professionals [21]. The TUM OpenInfraPlatform (OIP), which was developed by various participants from the Chair of Computational Modeling and Simulation (CMS) of the Technical University of Munich, puts a focus on developing a platform with the same goal [22]. By integrating a lot of different file formats and providing the most efficient exchange possible, collaboration and communication are elevated to a new level.

An integration of different domains should also be considered. Especially when handling infrastructure assets, an optimal consolidation can yield to significant results. By leveraging cross-domain information specific to each railway component, an integration of geographic information systems (GIS) with BIM can be achieved [23]. Possible implementations have also been explored. Prior research has demonstrated the potential of employing a Railway Infrastructure Information Model (RIIM) to enable integration and interoperability during the entire lifecycle of railway infrastructure, spanning from planning to maintenance [24]. Studies have investigated the cross-discipline integration, leading to an improved collaboration among project stakeholders and enhanced decision making. Nevertheless, it's worth noting the absence of research dedicated to the integrated use of BIM and GIS in railway projects [25].

Monitoring systems play a crucial role in obtaining more reliable results of the assets during their operational use. Previous studies suggest methods of leveraging results derived from continuously monitored measurements for asset maintenances and application in predictive maintenance [26]. This concept is closely related to the method of structural health monitoring which has gained importance in the past years. Concepts on how SHM can be integrated effectively in the workflow of infrastructure managers is treated in several research. [27], [28]. By having this real-time information at hand, an integration into digital models can be facilitated. Overall, the integration of digital methods can yield results in multiple dimensions. As mentioned in past research, the advancements of BIM authoring tools have further enhanced this transformation [29].

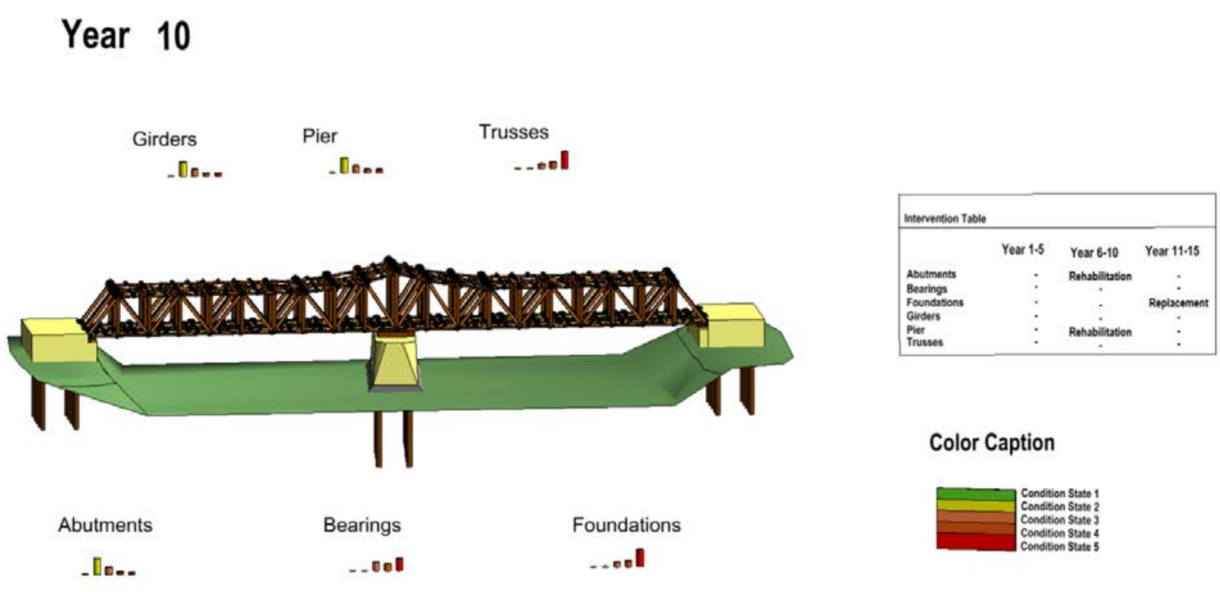
Additional research in the form of Master's Theses and Master's Projects has been conducted concerning the integration of BIM with the field of asset management. In the following section, two of them are introduced, both of which bear strong relevance to the work done in this Thesis.

Master's Thesis

The work conducted in this Thesis drew inspiration from the contributions of Mr. Andreas Stolz in his joint Master's Thesis between the Swiss Federal Institute of Technology (ETH Zürich) and the Technical University of Munich (TUM) [30]. The goal of the Thesis was to demonstrate a better way of visualizing data of predicted condition states and interventions for railway bridges, by leveraging digital models to a greater degree.

To obtain a clearer understanding of the anticipated condition of bridge assets at a certain future point, colors representing the condition state were incorporated into the 3D model. The 3D BIM models were created in the BIM authoring tool Autodesk Revit. Apart from the color representation, additional bar charts were placed around the model, providing a more detailed insight into the information of each bridge component. A table stating the type of planned interventions and their respective planning periods was also included.

Figure 5 displays the created BIM model of the a steel bridge, along with the accompanying visualization for the condition states utilizing a certain color scheme. The legend adjacent to it provides a key for decoding the color representations. This figure illustrates the probability of each component being in a particular estimated condition state ten years after the measurements have been made.



Intervention Table

	Year 1-5	Year 6-10	Year 11-15
Abutments	-	Rehabilitation	-
Bearings	-	-	-
Foundations	-	-	Replacement
Girders	-	-	-
Pier	-	Rehabilitation	-
Trusses	-	-	-

Figure 5: The BIM model of the steel bridge with visualizations of the condition states and interventions [30].

In order to achieve the presented visualization, a visual scripting code was developed in Dynamo for Revit. Figure 6 depicts the visualization process from creating the 3D models up to the final output. The functionalities of the Dynamo script and a more detailed explanation of its improvements are provided in Section 2.2.

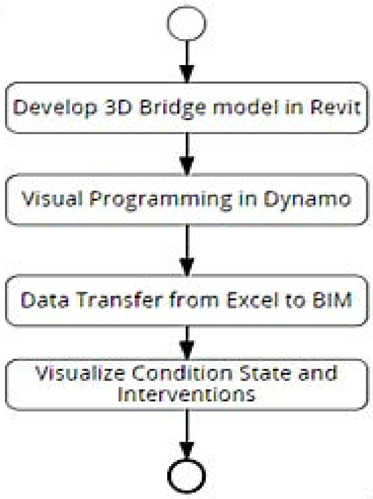
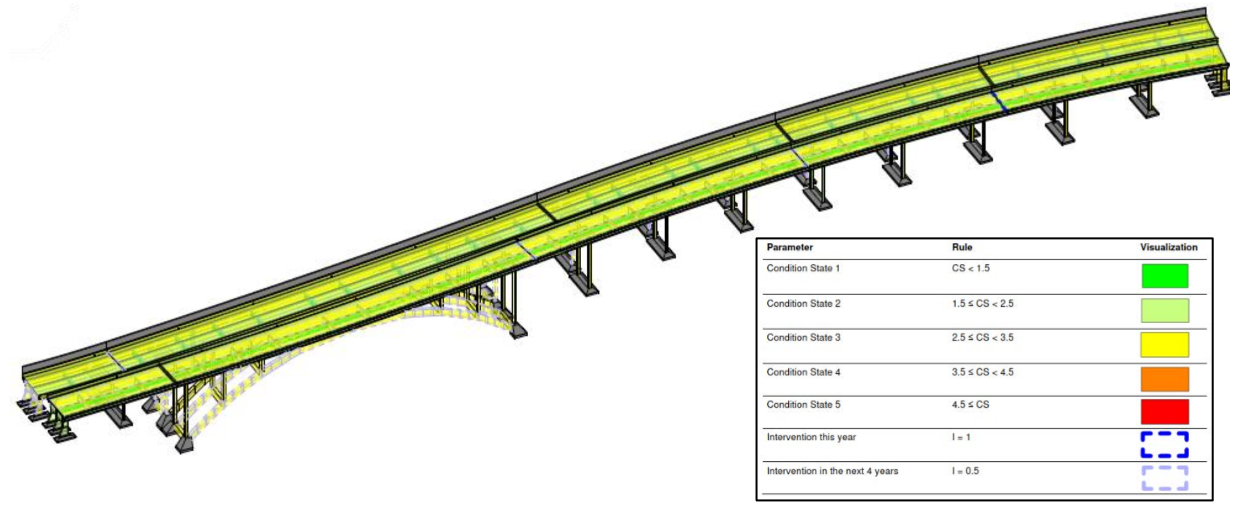

Develop 3D Bridge model in Revit
Visual Programming in Dynamo
Data Transfer from Excel to BIM
Visualize Condition State and Interventions

Figure 6: The processes followed in the previous Master's Thesis [30].

The previous Master's Thesis work had some limitations in terms of usability, representation, and automation. For instance, improvements regarding the flexibility and scalability of the workflow usage have been examined. The subsection ‘Optimization of previous work’ of Section 2 presents a deeper look into the limitations and the measures taken to provide a better overall result.

Master’s project

Another research connected to the subject matter presented in this Thesis is a study project at the ETH, part of the research project MINERVA, conducted by the Master’s students Ms. Katharina Fraberger and Mr. Fabio Tehlar [31]. They aimed to analyze and model two road bridges running next to each other and to assign and visualize the data of the condition states and interventions at an element level. Figure 7 shows a 3D visualization of the bridges where the different colors indicate the condition states of the bridge elements. The status of the interventions was shown in the model through an outline of dashed lines, with the definition explained by the provided legend.



Parameter	Rule	Visualization
Condition State 1	CS < 1.5	🟩
Condition State 2	1.5 ≤ CS < 2.5	🟩
Condition State 3	2.5 ≤ CS < 3.5	🟨
Condition State 4	3.5 ≤ CS < 4.5	🟧
Condition State 5	4.5 ≤ CS	🟥
Intervention this year	I = 1	⬚
Intervention in the next 4 years	I = 0.5	⬚

Figure 7: The BIM model of two road bridges with visualizations on the condition states and interventions [31].

For further clarity, a construction sheet was also produced out of the model. This approach allowed for a more concise capture of information for future use. Figure 8 shows an example construction sheet that was created using this workflow.

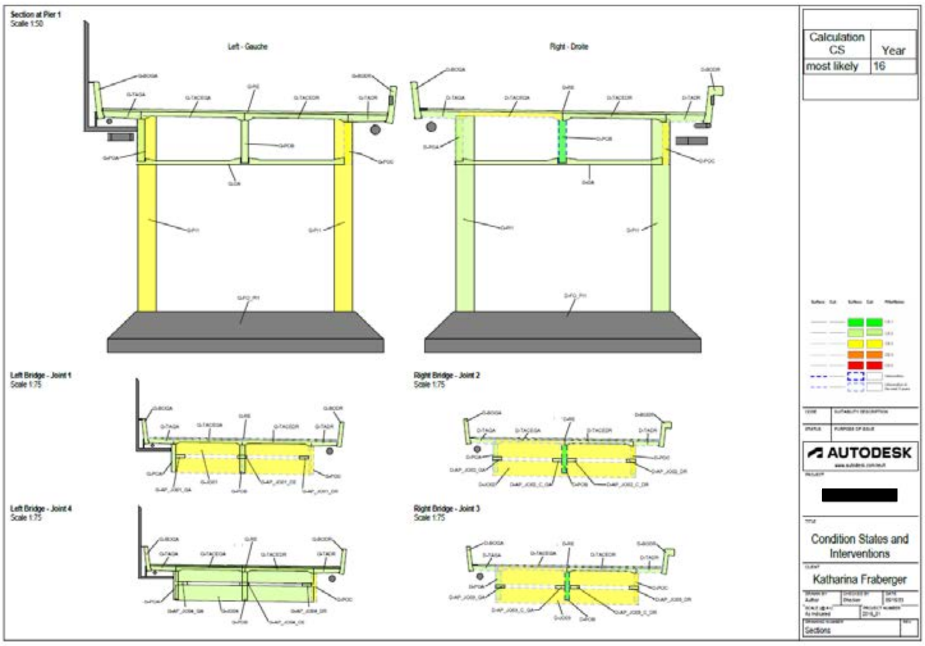


Figure 8: The construction sheet output at the element level [31].

1.4 Aims and objectives of this Thesis

Within the framework of the STABILITY project, this Thesis aims to provide a basis of visualizing results acquired from predictive maintenance analysis and facilitate the enrichment and usage of data in digital models. In this way, a framework that advances the communication between different asset managers could be created. Diverse scales, ranging from the close asset level to the broader network level, have been considered. This Thesis is built upon three main objectives:

Objective 1. Integrating digital data on an asset level

The first objective is to ensure a digital integration of data into a BIM environment. How an introduction of approaches to create digital models and exploring data representation through visualization, can assist bridge asset managers to seamlessly incorporate BIM into their workflows, is covered. The primary emphasis is on the asset level of bridges, as well as associated data regarding predicted condition states and interventions. This objective is discussed in detail in Section 2.

Objective 2. Providing a unified data source on a network level

The second objective is to investigate the subject from a more extensive perspective. It explores and discusses how combining bridge and track assets in a unified view, defining their synergies and associations, with a special emphasis on interventions within these parts, could advance the collaboration among different asset managers. The consistent integration between BIM and GIS has also been addressed. A full description of the followed approaches is provided Section 3.

Objective 3. Enhancing the usage of digital models in subsequent phases

The final objective is to consider ways of facilitating the usage of digital models in the construction phase. Propositions on how to use the same digital models in subsequent building phases and the concepts of elevating the dimensionality of BIM by leveraging 4D models have been examined. Section 4 covers the part of this objective.

Enhancing data integration and obtaining visual feedback from digital models is crucial in the aspect of asset management. Meanwhile, covering multiple disciplines and different phases is especially of importance in the context of infrastructure. The fulfillment of the stated objectives can facilitate an overall integration of BIM into the workflow of infrastructure management.

2 Modeling uncertainty

A crucial element concerning the infrastructure components analyzed within this Thesis is the condition state of these assets. The physical condition of an asset can be assessed through this metric while taking into consideration the damage on the given element [32]. The classification using condition states includes five distinct statuses determined through the precise data collected, evaluation through expert opinions or the utilization of deterministic deterioration models [33]. Table 2 shows an example of the potential condition states for the rebar of a concrete deck.

Table 2: Five condition states and the meaning of each state [33].

Item	State	Condition	Description of condition on entry
Deck	1	Good	No signs rust
	2	Acceptable	Minor spots of rust and/or spalling and thin cracks less than 2.5 mm wide
	3	Damaged	Major spots of rust and/or spalling and thin cracks
	4	Poor	Major spots of rust and/or spalling and section loss of reinforcement less than 10%
	5	Alarming	Section loss of reinforcement more than 10%

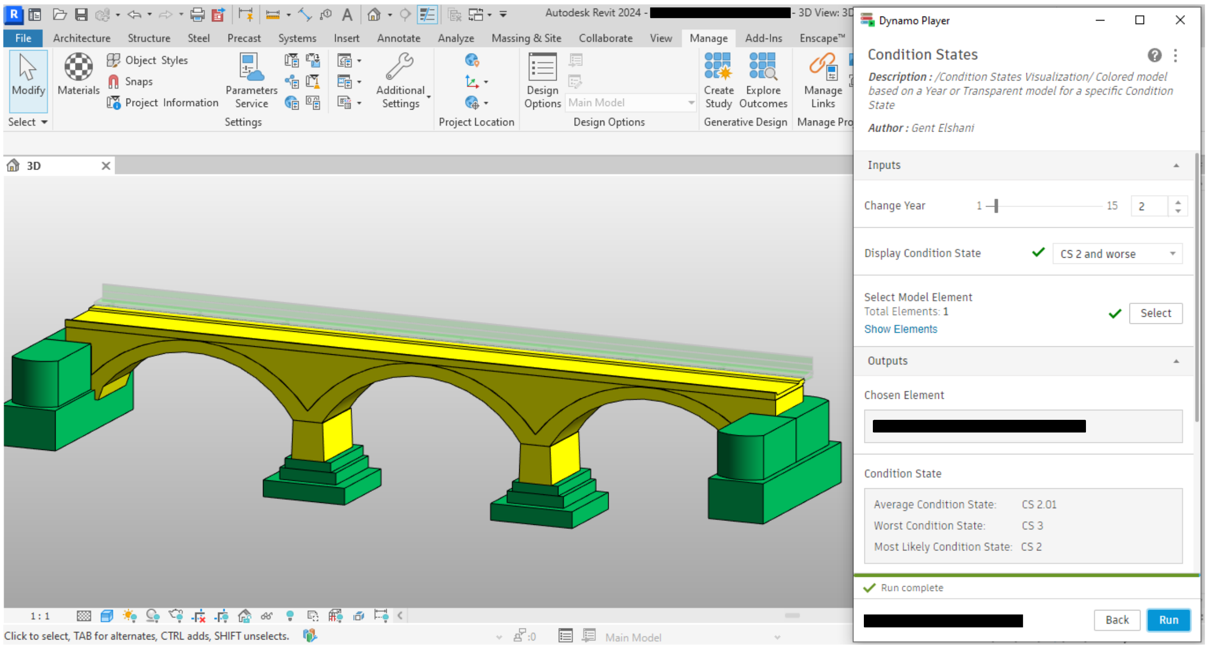
During the STABILITY project, certain algorithms have been developed to model the deterioration of the assets. The algorithms follow the principles of discrete state probabilistic modeling, which result in probabilistic condition states of the bridge components from 1 to 5. Much like the Markov Chain method, slight adaptions have been implemented to enable history information, in contrast to classic Markov Models.

As this part of the Thesis focuses more on the asset level, regarding bridges as standalone entities without considering the synergies with other components, it is conditionally linked to the so-called strategic asset management within the intervention planning process. The strategic asset management represents a major part of the development and maintenance planning, where two sections can be distinguished:

- **Bridge asset management**: involves the strategic management of a bridge portfolio to ensure safety, efficiency, and long-life expectancy. It includes data-driven decision-making, risk assessment, and budget allocation to optimize the overall performance.
- **Track asset management**: considers the strategic management of railway track infrastructure to ensure safe, efficient operations by monitoring, maintaining, and optimizing track assets.

The purpose of this section of the Master's Thesis is to utilize a digital 3D BIM model to visually represent the data concerning the condition states of bridge assets and the planned interventions generated by the aforementioned processes. This offers bridge asset managers a powerful tool for efficient decision-making and asset management. Several methods for visualizing the data include the use of a color-coding system allowing the manager to easily extract information on the health of various bridge components in a future year, making it easier to allocate resources effectively. Components in good condition can be color-coded green, while those in fair or poor states can be represented in yellow or red, respectively. This visual approach guarantees an immediate identification of critical areas that demand immediate attention, ensuring that maintenance efforts are focused where they are needed most, reducing operational risks and extending the lifespan of critical infrastructure.

Furthermore, transparency layers can be used to emphasize future condition states and planned interventions in the 3D model. This means that spatial coordination and precise resource allocation can also be planned, especially when handling complex models with a lot of components. An example of how the output could look like is presented in Figure 9. In addition to the digital model, interfaces providing further numerical information can also be incorporated into the workflow.



Condition States

Description : /Condition States Visualization/ Colored model based on a Year or Transparent model for a specific Condition State

Author : Gent Elshani

Inputs

Change Year 1 15 2

Display Condition State ✓ CS 2 and worse

Select Model Element
Total Elements: 1 ✓ Select
Show Elements

Outputs

Chosen Element

[redacted]

Condition State

Average Condition State: CS 2.01
Worst Condition State: CS 3
Most Likely Condition State: CS 2

✓ Run complete

[redacted] Back Run

Figure 9: The visualization of the condition states on a 3D model.

Ultimately, this integrated approach empowers bridge asset managers to make well-informed, data-driven decisions that enhance longevity, safety, and cost-efficiency of their bridge assets. It is worth noting that the intention here is not to substitute already existing methodologies that rely on more precise data obtained from tables and other sources. Instead, the presented work is only intended to serve as an aid to conventional workflows.

Optimization of previous work

The work conducted in the prior Master's Thesis by Mr. Andreas Stolz has been the subject of several improvements within the frame of this Thesis. The key improvements included the following categories:

- Quality of the models
- Scalability and flexibility of the visual programming script
- Ease of use of the workflow

Specific measures have been carried out in each of these sections to tackle the mentioned drawbacks. Within every section, there is a deeper analysis of the suggested solutions, accompanied by an explanation of the theoretical background and the logical thought process.

To establish a standardized approach for modeling bridge and track assets, while simultaneously speeding up the modeling time and reducing the required effort, a family catalog containing a library of standard parametric Revit elements has been created. Further details regarding this catalog can be found in Section 2.1. This report presents a detailed description of the improvements made in the visual programming script in Section 2.2. Such improvements facilitate the work with the script, automating a lot of tasks that previously had to be carried out manually. They also serve as metrics for enhancing the script's reusability across different settings and adopting a more general approach to distinct types of bridges. To ensure a smoother workflow, a more user-friendly user interface has been introduced. This aspect also potentially increases the user's acceptance to use the proposed tool in order to achieve a more efficient work process. In these terms, the Dynamo Player of Revit has been integrated into the method, which will be explained thoroughly in Section 2.3.

2.1 Family catalog

To secure an efficient workflow while minimizing potential future complications, the proposed solution involves establishing a standardized method for asset modeling. By using a set of standard elements, the modeling process can be facilitated by significant measures, making the creation of BIM models much easier and more accepted by potential users. It complies with the idea of thinking in objects, where all building or bridge elements are treated as individual objects with properties associated to their geometry, embracing the perspective of Object Oriented Modeling (OOM) [34].

While creating these objects, at least two aspects are analyzed: the geometry representing the 3D objects and the information underlying them. However, many more aspects have also been treated, such as the interoperability between different BIM authoring tools, the quality of the produced objects in terms of reusability in projects and further use in different use cases, as well as the parametrization and limitation of over-detailing.

The objects within the family catalog developed within the scope of this Thesis have been organized into multiple categories based on several conditions. Firstly, the approach in this Thesis aims to involve assets of railway bridges and tracks at the component level, which means leaving out more minor elements like connections. Also, the data of inspections available for different bridge and track sections is considered as a measure to form this catalog. Finally, a general separation of the components is examined based on the data catalog (Fachdatenkatalog) of the BIM@SBB initiative [35].

By using the mentioned data catalog as a basis, the family catalog has emerged as part of this Thesis, including the following domains:

Bridge domain

Here, the most relevant bridge assets are included, considering only components present in the construction drawings and data of the bridges are incorporated in this Thesis. However, if needed in newly created bridge models, the general methodology can be extended to include other bridge components, taking the FDK as a basis. Inside the bridge domain, the following categories of object types are involved, which further contain separate object types as the smallest instance in this hierarchy.

Superstructure:

- Deck
- Beam
- Arch

Substructure:

- Abutment
- Pier
- Bearing
- Spandrel Wall
- Wing Wall

Foundation:

- Footing
- Pile

Equipment:

- Railing
- Safety walkway

Track domain

As part of the track domain, five components are incorporated in the condition state deterioration and intervention models, which are now also included as objects in the BIM model. Based on the data catalog from the SBB, one object type has been included for each category. Additional equipment and elements to the components have not been taken into consideration. However, they can be incorporated following the proposed methodology, thereby extending the existing data structure. The object types that are included in the family catalog comprise:

- Rail
- Sleeper
- Fastening System
- Ballast/Subballast
- Subgrade

Also worth mentioning are the shared parameters that have been included in the family catalog. The integration has been done using a text file where all identity data regarding bridge and track assets are captured. This file can be included in any Revit family and provides a consistent and automatic way of including this data in newly created additions to the family catalog.

2.1.1 Parametric BIM modeling

The underlying theory behind creating the mentioned family catalog relies on the concepts of Parametric BIM Modeling [36]. This approach to geometry representation is one of three different parametric geometry modeling approaches, promoting the creation of objects from a certain library that should also contain parametric and general information. The different types of parametric modeling and some software solutions utilizing these objects can be observed in Figure 10.

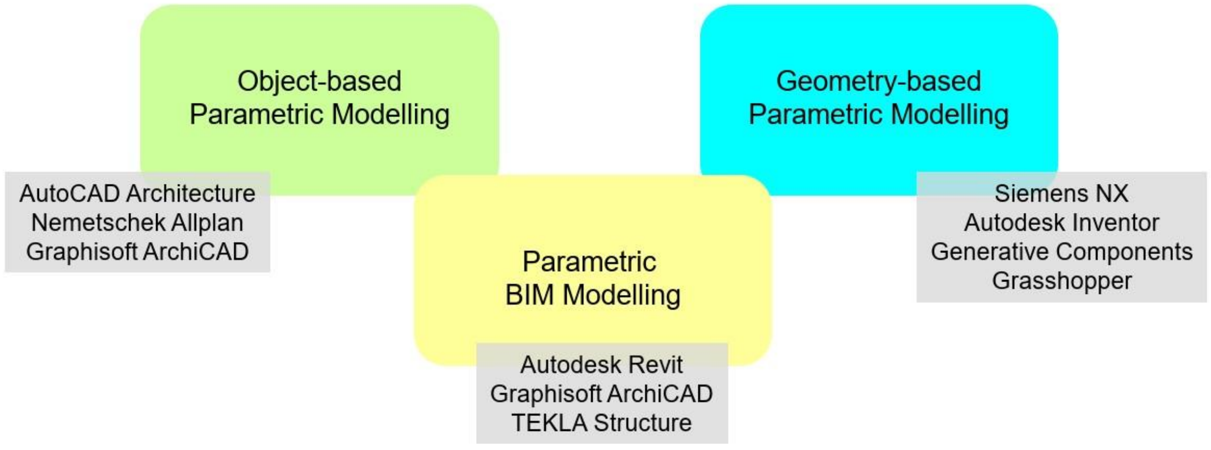


Figure 10: Different types of parametric modeling [36].

The other methods include Object-based Parametric modeling, where a standard library of objects is created and utilized, even without the necessity of having any parameters associated with them. However, this makes the usage of the same element in different scenarios more challenging since just changing a single parameter to fit another case is not possible.

Geometry-based Parametric Modeling, on the other hand, includes the parametrization of 2D or 3D geometry directly on the drawing or model by introducing constraints. This approach is then further subdivided into Sketch-based and Graph-based modeling depending on how the named constraints were created. In the case of Sketch-based parametrization, usually, the standard user interface of the authoring software is used in these senses by creating a sketch of the desired outcome on a reference plane. In the matter of Graph-based parametrization, the approach of visual programming is utilized. Further details concerning visual programming can be found in Section 2.2. Figure 11 depicts an example of a visual representation of the Sketch-based modeling, whereas Figure 12 shows a representation of Graph-based modeling.

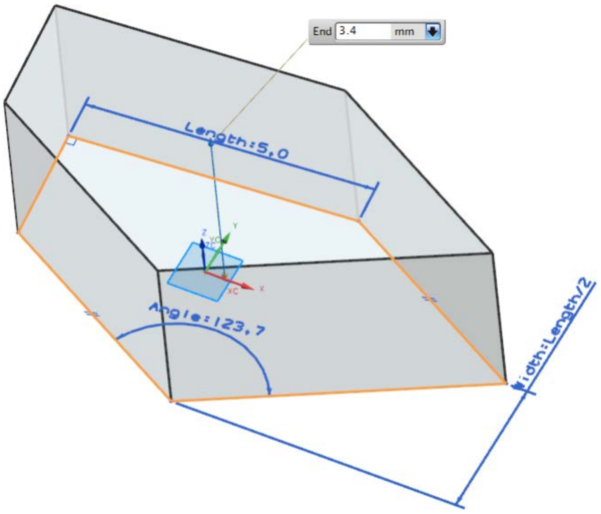


Figure 11: An example of the sketch-based parametric modeling [37].

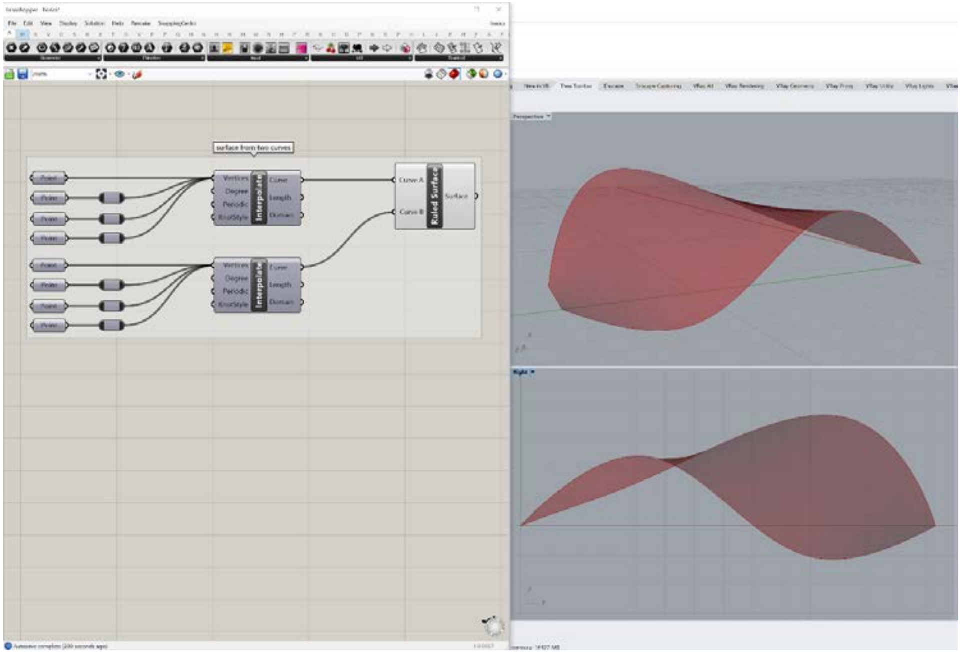


Figure 12: An example of the graph-based parametric modeling [38].

Parametric BIM modeling can be viewed as a combination of the aforementioned groups, where parametrized objects are created through either a sketch or a graph and organized within a library of objects for easier replication. The concept of Object orientation is also closely related to this modeling principle. Object orientation indicates a complex system which is broken down and characterized by its composing objects. By modeling the geometry in accordance with the introduced concept, the framework of OOM is created [34].

2.1.2 Level of development

BIM models can significantly differ from one another depending on specific metrics, which can be used to compare the maturity of the models, express the quality, and evaluate the usability of these digital models. One of the most commonly performed assessments is the leverage of the Level of Development (LOD). Through it, the establishment of a BIM model can be understood [34]. Depending on the followed regulations and specified sources, the meaning and representation of these abbreviations can differ. In the context of this Master's Thesis, the specifications from the BIMForum have been taken as a reference [39].

In Figure 13, the five different levels of the mentioned competence can be considered in terms of geometry, ranging from conceptual detail to an as-built model. In Figure 13, the as-built model, represented by LOD 500, has not been included as it presents the same model as in LOD 400, with the sole distinction of having on-site assurance performed.

Various sources are available for obtaining information about specific components, including references like VDI 2552 [40]. It has to be pointed out that these measures are specific to buildings and engineering structures, mainly based on the scale of the structure under evaluation. On this basis, the LOD of a larger-scale terrain model or city model is measured slightly differently.

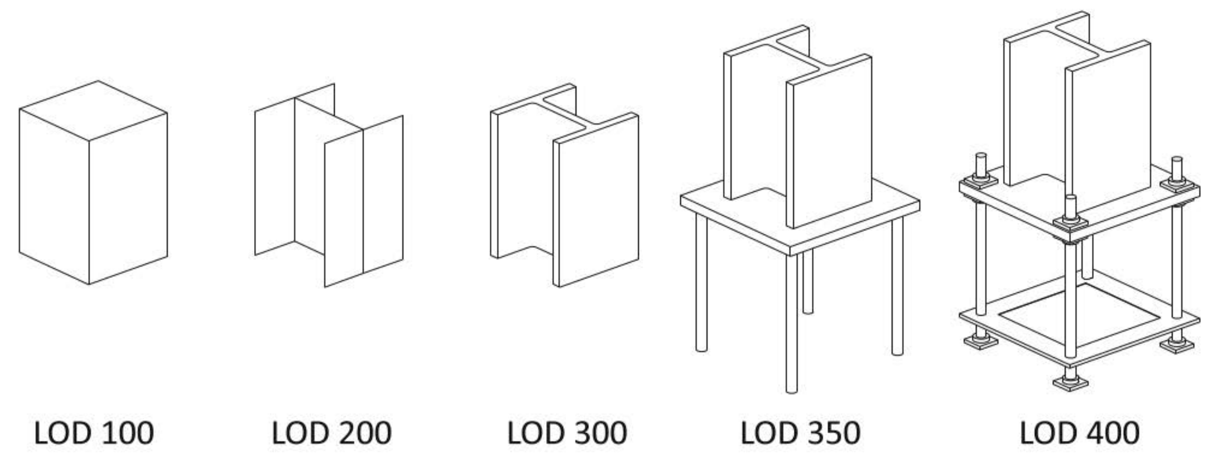


Figure 13: Different levels of development of a steel profile element [41].

The Level of Development is subdivided into two parts: the Level of Geometry (LOG) and the Level of Information (LOI). The modeled bridge assets in this Thesis aim to fulfil a LOD of 200. The specifics of this evaluation are described in the following parts.

Level of Geometry

The definition of level of geometry follows the same maturity as the level of development, meaning that a LOG from 100 to 500 is possible [34]. Figure 14 illustrates another example of the preciseness of the model elements in relation to the corresponding LOG.

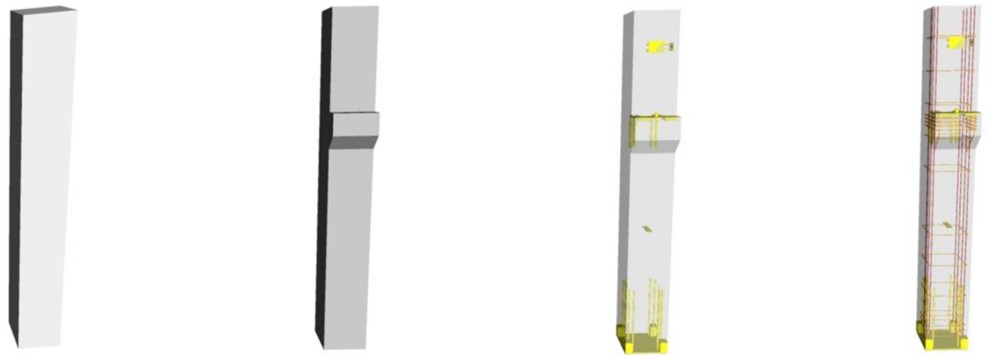


Figure 14: Different levels of geometry of a column element [39].

Especially at more complex model elements, the determination of the appropriate LOG it belongs to can be highly subjective. For this reason, usually sample tables are provided as a basis for comparison. More developments have been made regarding building elements and accompanying components to them, whereas specifications considering infrastructure, bridges, and tunnels were made only in the recent years.

In the context of this Thesis's case study, the aim is to achieve an LOG of no greater than 200 for the overall models, while specific elements may have an LOG of 300. Since the use-case being addressed does not require highly detailed models, this Level of Geometry has been selected. However, specific elements can have a more precise representation due to better visibility.

Level of Information

Apart from the geometric models, other semantic data have also been included in the standard objects in the form of parameters. Special emphasis has been placed on the parameters describing the geometry, such as the thickness, width, and length.

Additional data have been added while focusing on the data exchange and identity data. The parameters of an object in Revit have been assigned in compliance with data taken from the BIM@SBB initiative [35].

The needed data has been provided in the form of a series of Tables. For example, Tables 3 and 4 show part of the tables containing precise information needed in the objects and describing different forms of data exchange formats, with a particular focus on the IFC data format. Even the most recent IFC format, IFC 4x3, has been incorporated, making use of the classes specifically developed to address the absence of bridge and railway track components in past versions [42].

Table 3: Overview representing a subset of a larger table containing the object properties for a bridge component [35].

Object properties

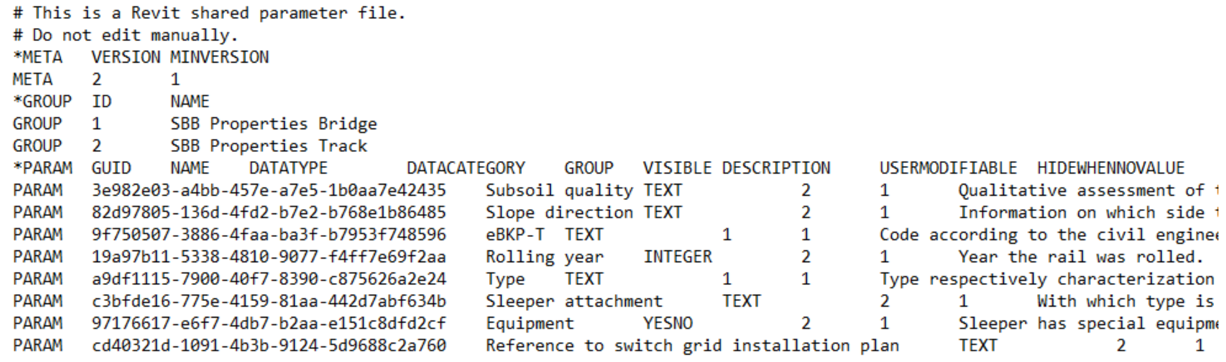
Name	Description	Format	Unit	Example	ID
Nutzungsdauer	Technische Nutzungsdauer der Brücke gemäss Nutzungsvereinbarung	Integer		80	PTY_2980
Baujahr	Baujahr der Brücke	Date	y	1999	PTY_2981
Zustandsklasse	der Brücke gemäss RTE 29900	Enumeration		true	PTY_2982
eBKP-T	Code nach Baukostenplan Tiefbau eBKP-T	String		O 5.4	PTY_2983
BKP (SBB)	gemäss Vorgaben aus BKP-Katalog SBB	String		313	PTY_2984
Typ	Typ resp. Charakterisierung des Objekts (Zusatzinformation)	String			PTY_2985
Produkt	Produkteart des Objekts	String			PTY_2986
Material	Materialisierung des Objekts	String			PTY_2987
tragendes Bauteil?	Bauteil, das eine primäre Tragfunktion der Gesamtstruktur wahrnimmt	Boolean		TRUE	PTY_2993

Table 4: A part of the IFC assignment table for a bridge component [35].

IFC Assignment

Version	Class	Type	Predefined type
IFC-4ADD2-TC1	IfcFooting	IfcFootingType	USERDEFINED
IFC-4ADD2-TC1	IfcFooting	IfcFootingType	FOOTING_BEAM
IFC-4ADD2-TC1	IfcFooting	IfcFootingType	PAD_FOOTING
IFC-4ADD2-TC1	IfcFooting	IfcFootingType	PILE_CAP
IFC-4ADD2-TC1	IfcFooting	IfcFootingType	STRIP_FOOTING
IFC-4x3	IfcFooting	IfcFootingType	CAISSON_FOUNDATION
IFC-4x3	IfcFooting	IfcFootingType	FOOTING_BEAM

Through this approach, the goal has been to achieve a LoI of 300. The specific implementation in Revit has occurred through shared parameters. A collection of the parameters has been saved into a text file called “SBB Parameters”, where all the parameters can be found and used in the objects where they are needed. A part of this text file can be seen in Figure 15. These parameters have been grouped into two types: the bridge and the track parameters. Thereby, a coherent collection of parameters has been created to facilitate the assignment of information into the objects.



```
# This is a Revit shared parameter file.
# Do not edit manually.
*META   VERSION MINVERSION
META    2       1
*GROUP  ID      NAME
GROUP   1       SBB Properties Bridge
GROUP   2       SBB Properties Track
*PARAM  GUID    NAME    DATATYPE        DATACATEGORY    GROUP   VISIBLE DESCRIPTION     USERMODIFIABLE  HIDEWHENNOVALUE
PARAM   3e982e03-a4bb-457e-a7e5-1b0aa7e42435    Subsoil quality TEXT            2       1       Qualitative assessment of
PARAM   82d97805-136d-4fd2-b7e2-b768e1b86485    Slope direction TEXT            2       1       Information on which side
PARAM   9f750507-3886-4faa-ba3f-b7953f748596    eBKP-T  TEXT            1       1       Code according to the civil engine
PARAM   19a97b11-5338-4810-9077-f4ff7e69f2aa    Rolling year    INTEGER         2       1       Year the rail was rolled.
PARAM   a9df1115-7900-40f7-8390-c875626a2e24    Type    TEXT            1       1       Type respectively characterization
PARAM   c3bfde16-775e-4159-81aa-442d7abf634b    Sleeper attachment      TEXT            2       1       With which type is
PARAM   97176617-e6f7-4db7-b2aa-e151c8dfd2cf    Equipment       YESNO           2       1       Sleeper has special equipm
PARAM   cd40321d-1091-4b3b-9124-5d9688c2a760    Reference to switch grid installation plan      TEXT            2       1
```

Figure 15: Text file containing the shared parameters in Revit.

2.1.3 Modeling workflow

With the intent of establishing the family catalog within Revit, the family level of the hierarchy of elements in Revit (see Figure 16) has been selected [43]. This turned out to be the most flexible level but robust enough to provide a stable basis for a correct grouping. The possibility of creating custom-made families from scratch, in addition to the default system families, has enforced the choice of this layer. The categorization happens through parameters, meaning that depending on what input values are expected, a new family needs to be created. For example, in Figure 16, two families for the columns are created since the input parameter in one is the radius, as opposed to the other one where width and length have to be provided as input.

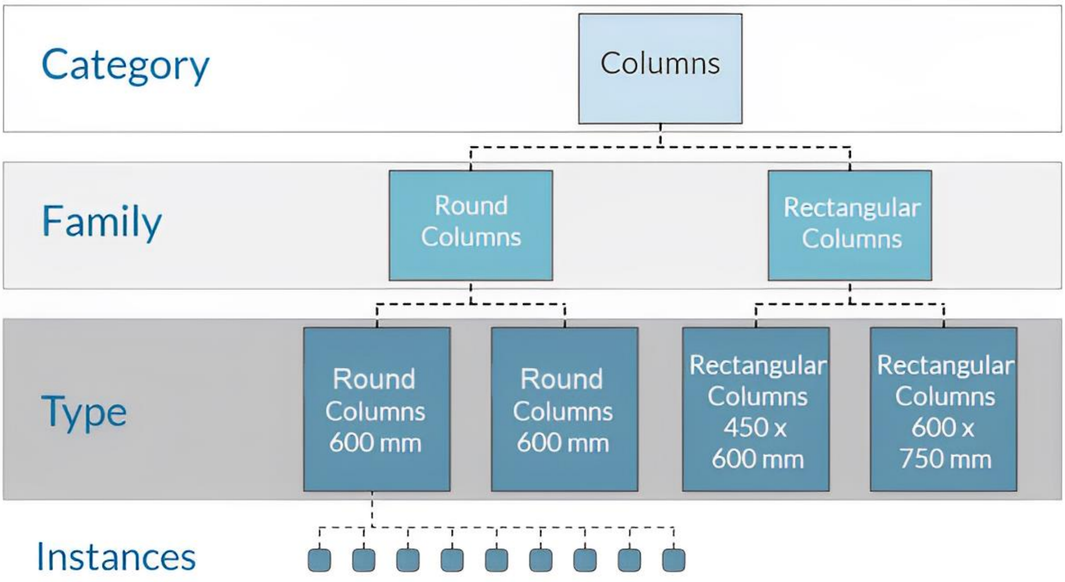


Figure 16: Element hierarchy in Revit [43].

Most of the already created geometry in the previous Master’s Thesis was produced as in-place modeled families, where their further use in other projects is very restricted. The assignment of parameters and other information is limited in these objects, significantly impairing the scalability and flexibility. Due to these reasons, the approach of loadable families has been selected, packing them all together into one catalog, as explained earlier.

A particular emphasis has been placed on correctly assigning the accurate Revit categories to those objects. This has not been dealt with in the already modeled assets, which can nonetheless considerably enhance productivity. This assignment can help the automation process by providing an aid for filtering and scheduling tasks, which is especially crucial in infrastructure projects because of the big scale of the projects.

Family creation

The family creation process can be carried out very straightforwardly through the Family Editor Tools. This can be achieved by choosing from a wide range of templates, where the primary focus should be on the selection based on fitting categories, as well as model characteristics. These features could encompass information such as whether the model element should be wall-based or constrained to two levels. After opening the interface, the model can be created by adding geometric forms into the user interface and manipulating them. The geometric forms follow the principles of implicit geometric modeling, also known as procedural modeling methods, meaning that the history of creating these geometric bodies is stored and can be recreated by following the stored steps [34]. Some approaches of implicit geometric modeling include Constructive Solid Geometry (CSG) and Extrusion and Rotation Methods.

The geometric elements created in the family editor of Revit focus on the geometric bodies generated through the Extrusion and Rotation Methods. Meanwhile, in the CSG methods, simple geometric objects, so-called primitives, are combined using Boolean operators like union, difference, and intersection forming in this way more complex geometric elements. The aim of the extrusion and rotation methods is moving a 2D shape along a 3D path. Depending on the way this relation is executed, methods like extrusion, sweep, and loft can be distinguished (see Figure 17) [44].

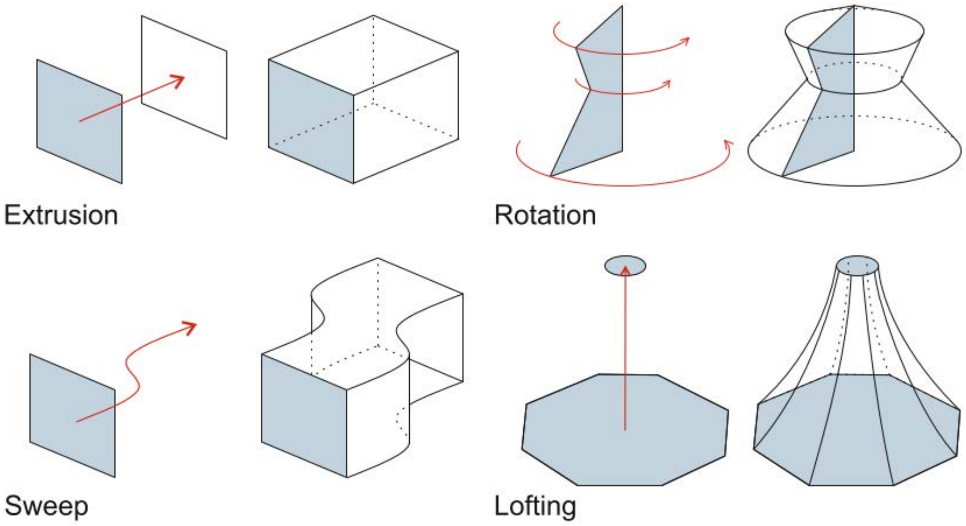
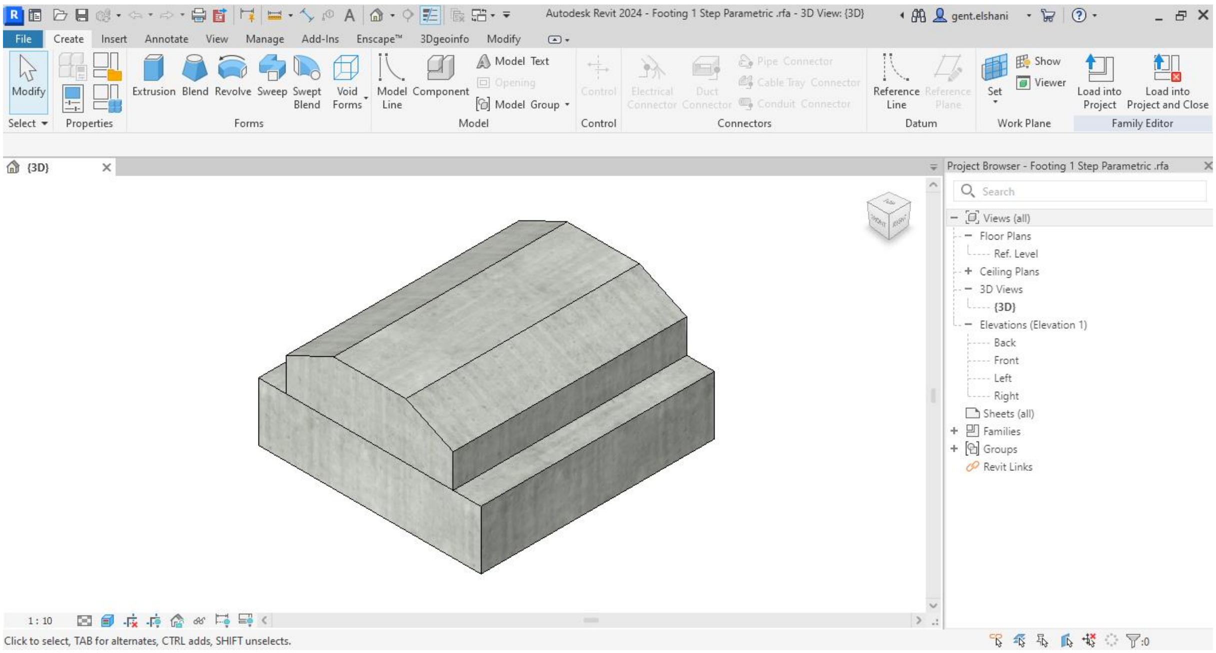


Figure 17: Different types of extrusion and rotation methods [44].

In order to achieve the creation of more complex geometry inside the family editor, void forms can also be created. These bodies adhere to the same principle as the creation of geometric bodies, but they produce compound bodies when combined with filled geometries, following the principles of geometry trimming [44]. This trimming can be an issue in relation with using NURBS curves of a higher polynomial degree, which is usually the case for representing curves in CAD authoring tools. The creation of dirty geometries can result in this way, making it not possible for the geometric bodies to be created [45].

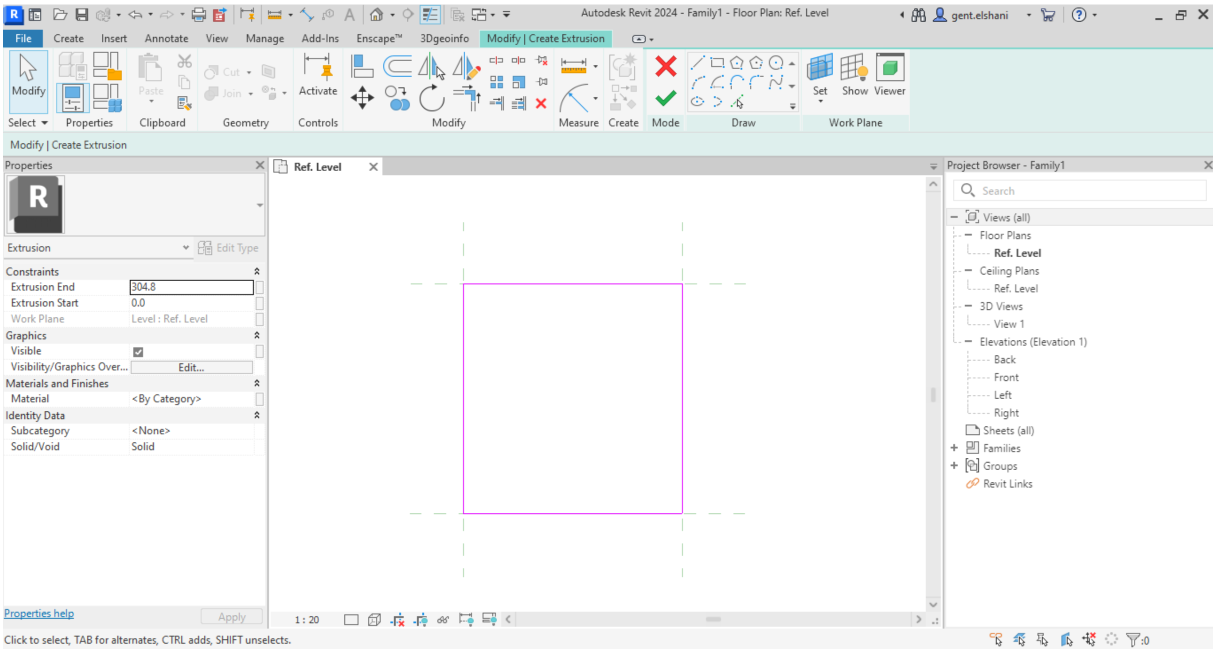
As an example, the creation of the family of a custom-made structural foundation will be demonstrated using the family editor in Revit. Special emphasis will be placed on assigning parameters and automating the attribution of repeating information using shared parameters. As seen in Figure 18, the final result yields a custom parametric object that can be used in various projects by only modifying the parameters.



Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 18: A custom-made parametric family object in Revit.

Firstly, the family template of the structural foundation has been used as a basis. By doing this, the category has been assigned to the structural foundation category, as well as a few parameters have been added to the family, such as foundation thickness, length, and width. These parameters are still not functional, since they must be assigned to the geometry, which yet remains to be created. Following the general setup of the family file, the geometry can be created using the extrusion method, as explained earlier. It is of best practice to constrain the geometry on reference planes and then manipulate the parameters based on them. After creating and modifying the reference planes in the view, an extrusion can be created while using the reference planes as a guide for the extrusion profile (see Figure 19).



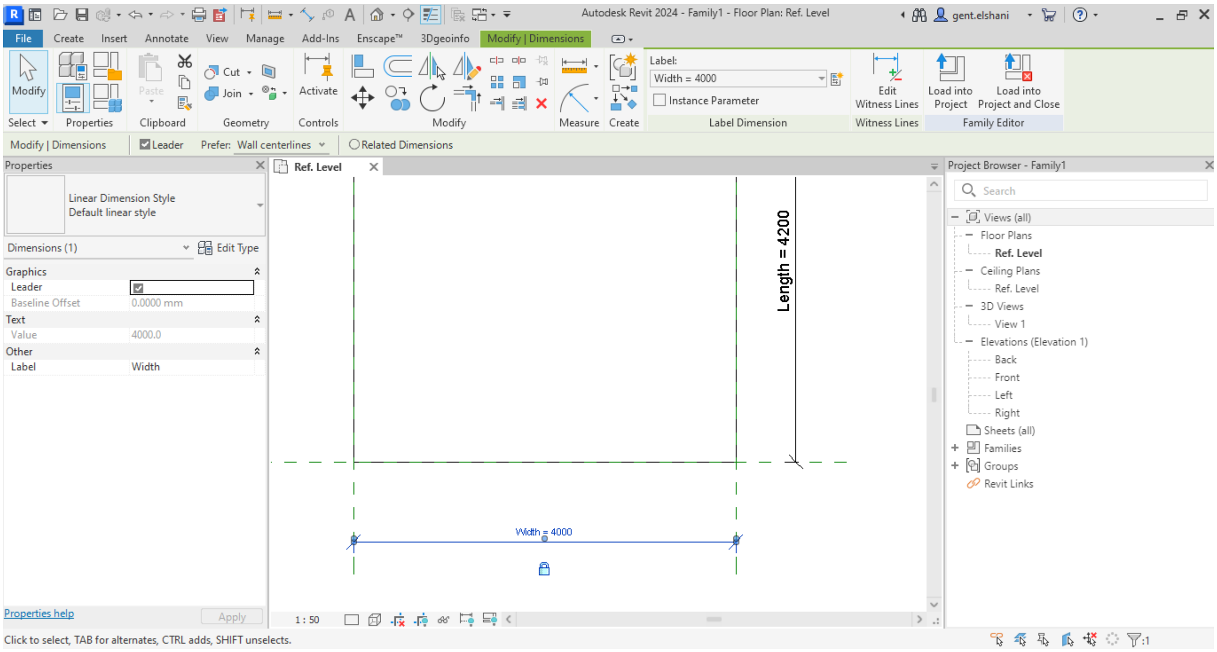
Extrusion

Constraints	
Extrusion End	304.8
Extrusion Start	0.0
Work Plane	Level : Ref. Level
Graphics	
Visible	☑
Visibility/Graphics Over...	Edit...
Materials and Finishes	
Material	<By Category>
Identity Data	
Subcategory	<None>
Solid/Void	Solid

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 19: The extrusion profile with the reference planes.

Once the extrusion is generated, the parameters for the length and width can be assigned to the dimensions measuring the reference planes, thus constraining the distance to a numerical input value (see Figure 20). Similarly, the foundation thickness can also be constrained through the distance between the reference level and a newly created reference plane.



Label: Width = 4000

□ Instance Parameter

Modify | Dimensions ☑ Leader Prefer: Wall centerlines ○ Related Dimensions

Linear Dimension Style
Default linear style

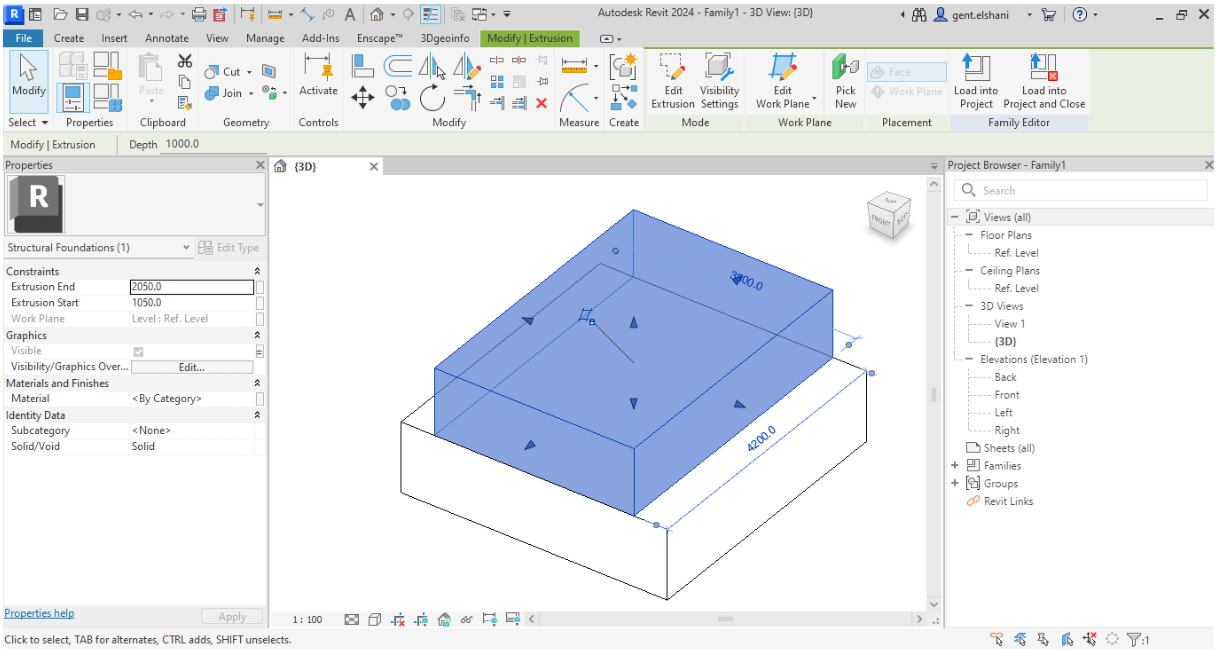
Dimensions (1)

Graphics	
Leader	☑
Baseline Offset	0.0000 mm
Text	
Value	4000.0
Other	
Label	Width

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 20: Parametric information assigned to dimensions.

Within one family instance, it is possible to generate multiple geometric bodies, offering the possibility to create another extrusion, similar to the procedure explained earlier, on top of the first element that represents the step of the foundation. By assigning the ‘Visible’ property of this extrusion to a newly created parameter, which contains the options to set it as ‘True’ or ‘False’, the step of the foundation can be enabled to be visible by manipulating this parameter (see Figure 21).


Autodesk Revit 2024 - Family1 - 3D View: {3D}

Modify | Extrusion Depth 1000.0

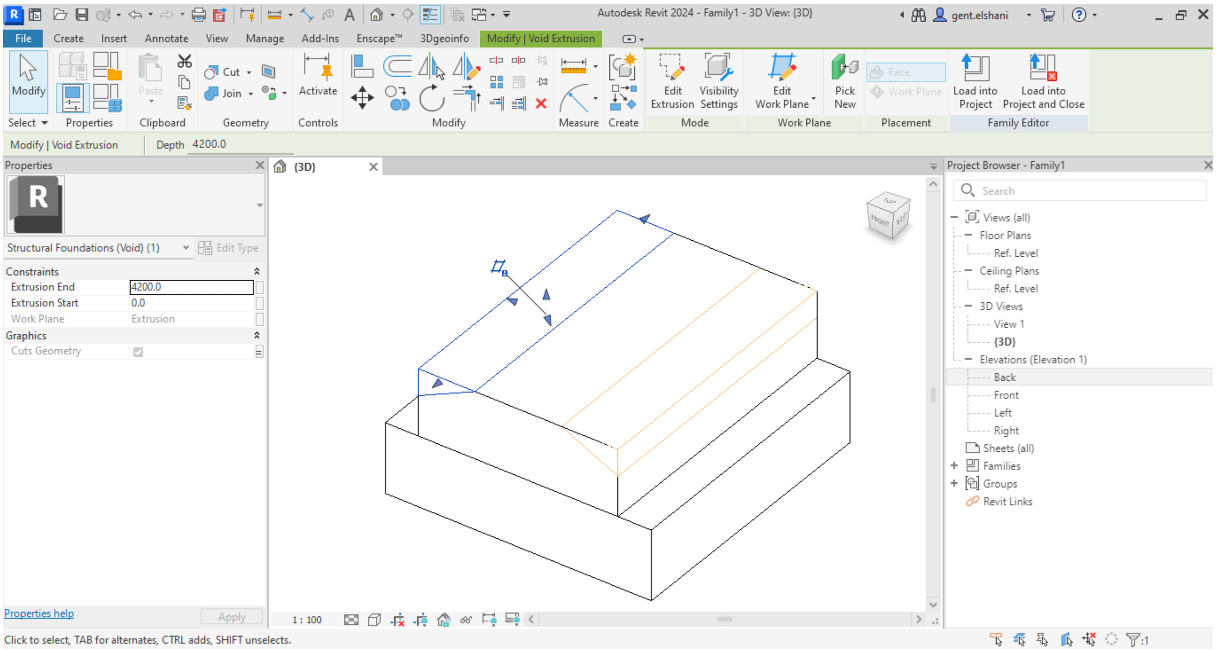
Structural Foundations (1)

Constraints	
Extrusion End	2050.0
Extrusion Start	1050.0
Work Plane	Level : Ref. Level
Graphics	
Visible	☑
Visibility/Graphics Over...	Edit...
Materials and Finishes	
Material	<By Category>
Identity Data	
Subcategory	<None>
Solid/Void	Solid

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 21: Assignment of the ‘Visible’ parameter.

Afterwards, the option to include chamfered edges to this foundation can be enabled by creating void extrusions on two sides of the structural foundation, which will only cut the geometry if the property called ‘Cuts Geometry’ is set in an implemented state (see Figure 22). This property is also linked to a newly created parameter, making it possible to change the value from outside the family editor.


Autodesk Revit 2024 - Family1 - 3D View: {3D}

Modify | Void Extrusion Depth 4200.0

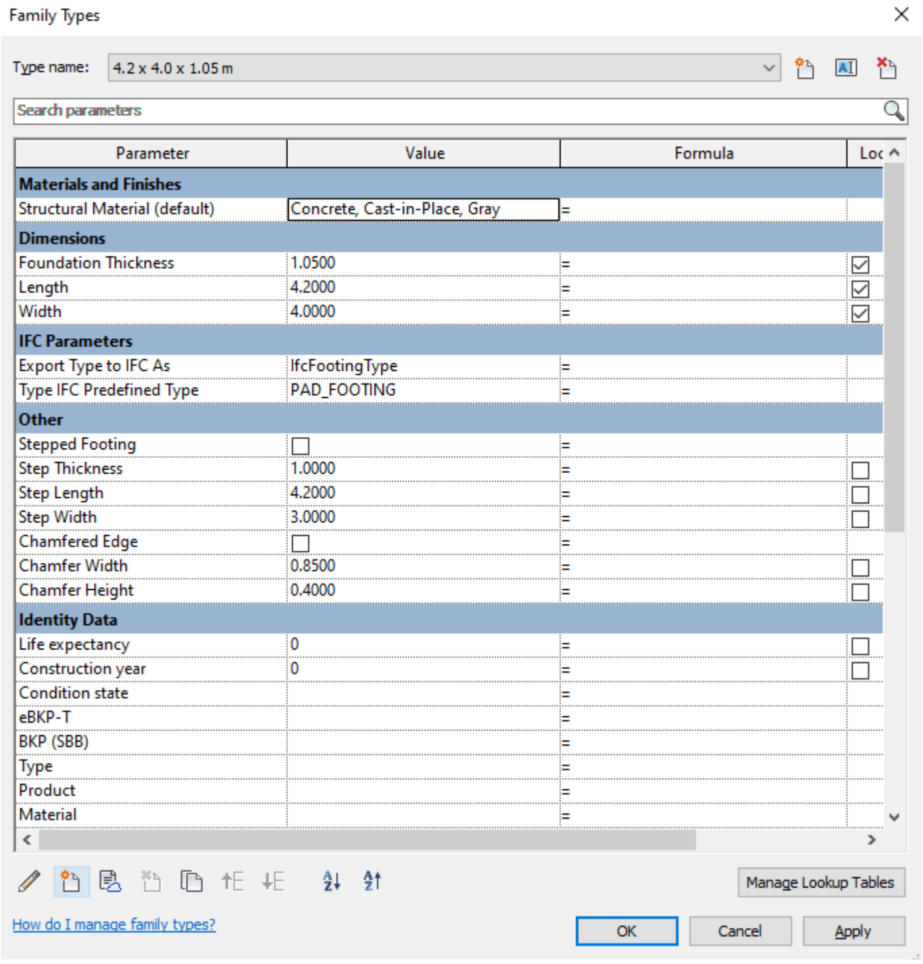
Structural Foundations (Void) (1)

Constraints	
Extrusion End	4200.0
Extrusion Start	0.0
Work Plane	Extrusion
Graphics	
Cuts Geometry	☑

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 22: Creation of void forms in Revit.

Finally, further parameters, such as the material, can be assigned. An overview of how the parameters of a newly created family for the family catalog can look like is provided in Figure 23. Here, the IFC exchange parameters and the identity data have been obtained from the mentioned tables provided by the BIM@SBB initiative.



Family Types

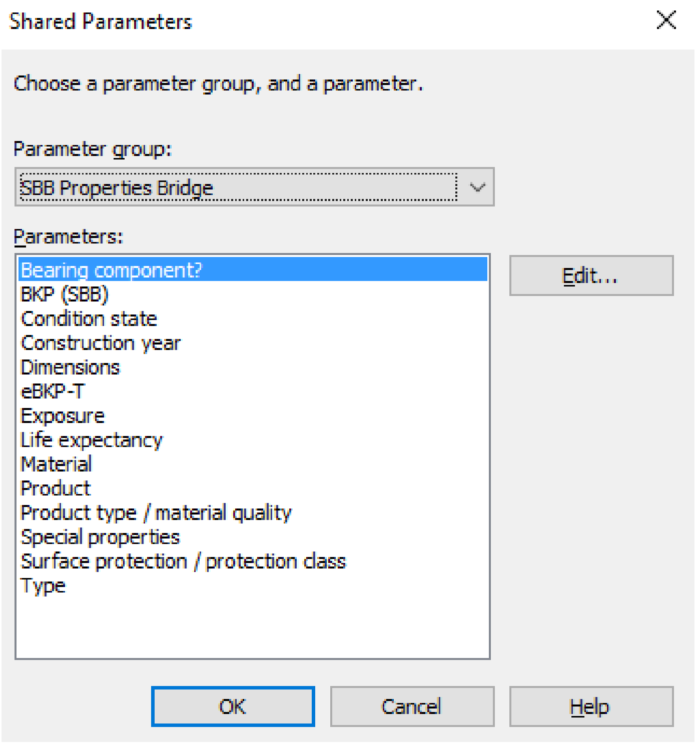
Type name: 4.2 x 4.0 x 1.05 m

Search parameters

Parameter	Value	Formula	Loc
Materials and Finishes			
Structural Material (default)	Concrete, Cast-in-Place, Gray	=	
Dimensions			
Foundation Thickness	1.0500	=	☑
Length	4.2000	=	☑
Width	4.0000	=	☑
IFC Parameters			
Export Type to IFC As	IfcFootingType	=	
Type IFC Predefined Type	PAD_FOOTING	=	
Other			
Stepped Footing	□	=	
Step Thickness	1.0000	=	□
Step Length	4.2000	=	□
Step Width	3.0000	=	□
Chamfered Edge	□	=	
Chamfer Width	0.8500	=	□
Chamfer Height	0.4000	=	□
Identity Data			
Life expectancy	0	=	□
Construction year	0	=	□
Condition state		=	
eBKP-T		=	
BKP (SBB)		=	
Type		=	
Product		=	
Material		=	

Figure 23: Some parameters of a family instance.

Since the identity data of these parametric elements include a lot of information in the form of parameters, an assignment through shared parameters in Revit has been utilized. The needed parameters can be selected and included in the produced Revit family by locating the shared parameters, which are written into a text file, and opening them in the dialogue generated when creating a new parameter (see Figure 24). The text file containing all the relevant information is separated into SBB properties for bridges, as well as for railway tracks, since the requirements change slightly.


Shared Parameters

Choose a parameter group, and a parameter.

Parameter group: SBB Properties Bridge

Parameters:

- Bearing component?
- BKP (SBB)
- Condition state
- Construction year
- Dimensions
- eBKP-T
- Exposure
- Life expectancy
- Material
- Product
- Product type / material quality
- Special properties
- Surface protection / protection class
- Type

Figure 24: Assignment of parameters through shared parameters.

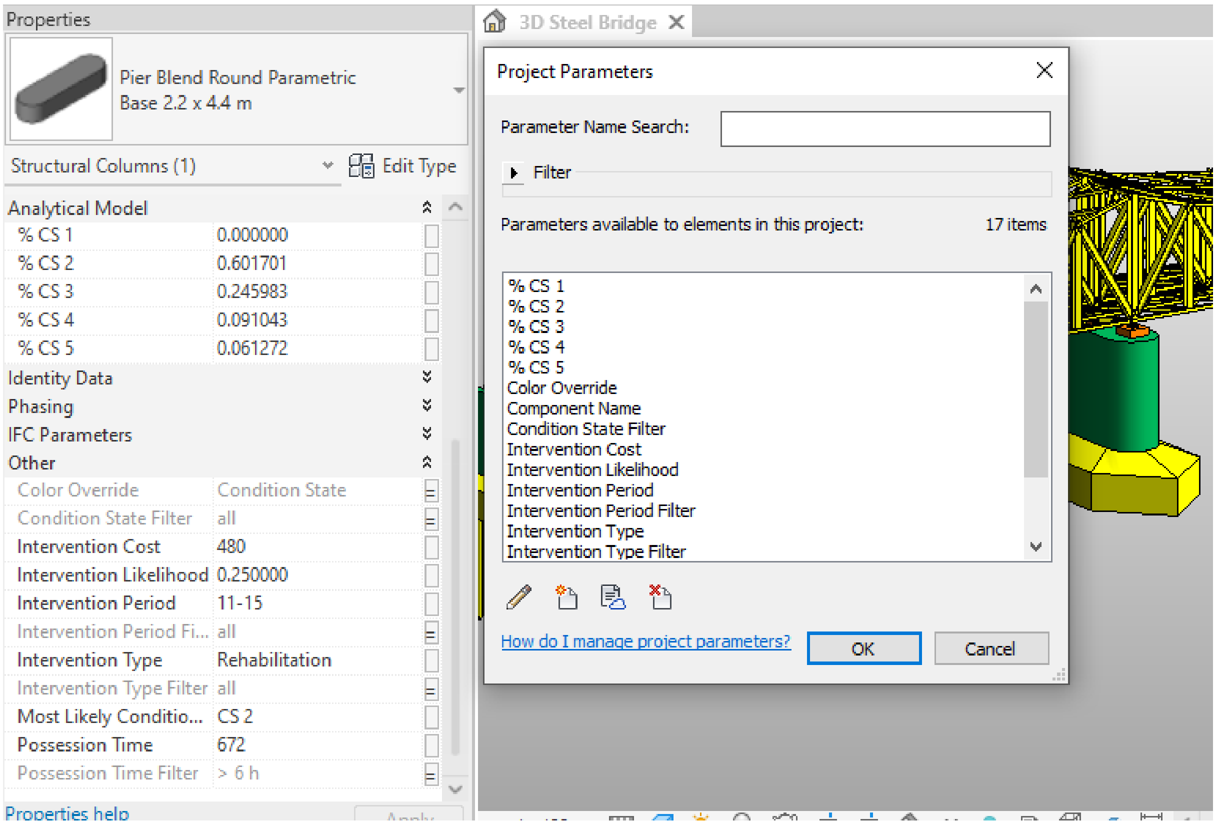
By utilizing the demonstrated workflow, a custom-created parametric family can be created. Any number of types can be generated through this family, depending on the geometry requirements of the project. Usually, in the case when what is expected as an input into the geometry varies a lot between different components, new families need to be created, fitting the needs of the required parameters.

Project setup

In order to efficiently use the family catalog and to acquire the proposed visualization, a specific setup of the Revit project containing the bridge assets needs to be done. This setup includes the arrangement of the following entities:

- Project Parameters
- Global Parameters
- Filter Rules
- Filter Overrides
- View Settings

The project parameters are needed, to store information about the condition state deterioration and different aspects of the interventions planned to be executed, on a particular asset. These parameters have been assigned using the Revit category the components belong to, following the selection of only the categories treated in this Thesis's scope. Afterwards, all components will automatically have the named parameters available in the instance information. Figure 25 shows an example of the project parameters assigned to a bridge pier. This embedded information can then be used for the proposed visualization, as well as other use cases of the model.



Properties

Pier Blend Round Parametric
Base 2.2 x 4.4 m

Structural Columns (1)

Analytical Model	
% CS 1	0.000000
% CS 2	0.601701
% CS 3	0.245983
% CS 4	0.091043
% CS 5	0.061272
Identity Data	
Phasing	
IFC Parameters	
Other	
Color Override	Condition State
Condition State Filter	all
Intervention Cost	480
Intervention Likelihood	0.250000
Intervention Period	11-15
Intervention Period Fi...	all
Intervention Type	Rehabilitation
Intervention Type Filter	all
Most Likely Conditio...	CS 2
Possession Time	672
Possession Time Filter	> 6 h

Project Parameters

Parameter Name Search:

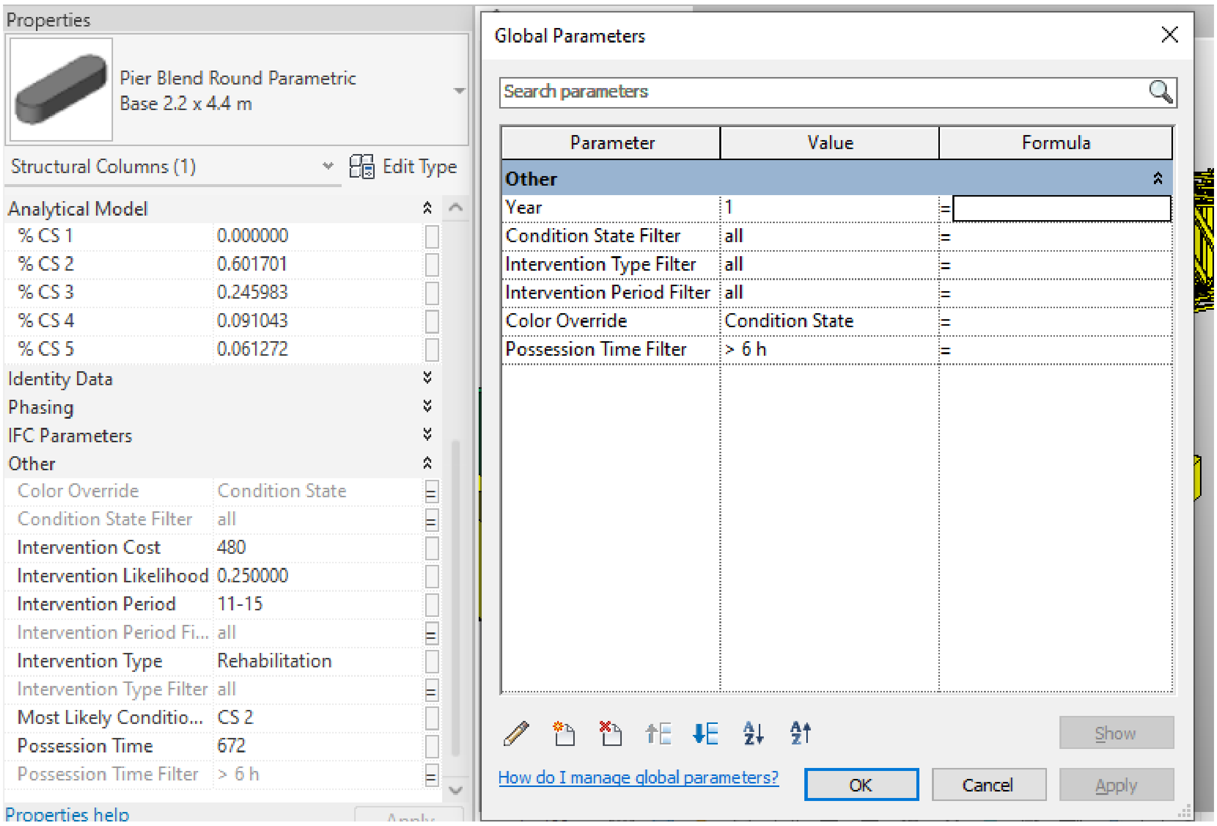
Filter

Parameters available to elements in this project: 17 items

% CS 1
% CS 2
% CS 3
% CS 4
% CS 5
Color Override
Component Name
Condition State Filter
Intervention Cost
Intervention Likelihood
Intervention Period
Intervention Period Filter
Intervention Type
Intervention Type Filter

Figure 25: Project parameters and their assignment on a component.

In order to provide values as a basis to compare the user input with the calculated values for the condition states and interventions on the components which will be assigned using the visualization script, global parameters have been additionally included in the project setup. In this Thesis, the values have been added in the form of filters, which change based on user input. The assignment of these parameters works similarly to the approach mentioned in the project parameters' part, using component categories as a basis for the assignment. Figure 26 shows several global parameters included in the project, which provide filters that are fed into the workflow.



Properties

Pier Blend Round Parametric
Base 2.2 x 4.4 m

Structural Columns (1) — Edit Type

Analytical Model	
% CS 1	0.000000
% CS 2	0.601701
% CS 3	0.245983
% CS 4	0.091043
% CS 5	0.061272
Identity Data	
Phasing	
IFC Parameters	
Other	
Color Override	Condition State
Condition State Filter	all
Intervention Cost	480
Intervention Likelihood	0.250000
Intervention Period	11-15
Intervention Period Fi...	all
Intervention Type	Rehabilitation
Intervention Type Filter	all
Most Likely Conditio...	CS 2
Possession Time	672
Possession Time Filter	> 6 h

Properties help

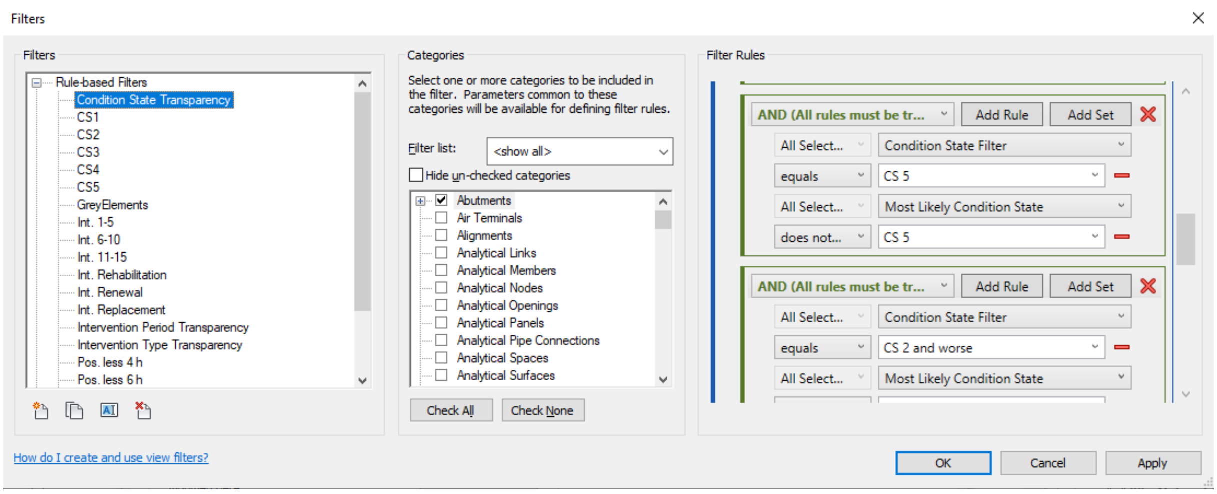
Global Parameters

Search parameters

Parameter	Value	Formula
Other		
Year	1	=
Condition State Filter	all	=
Intervention Type Filter	all	=
Intervention Period Filter	all	=
Color Override	Condition State	=
Possession Time Filter	> 6 h	=

Figure 26: Global parameters and their assignment on a component.

For the purpose of visualizing the information that the separate components possess, view specific filter overrides have been deployed. These filters follow specific rules, as seen in Figure 27, where two parameters are being compared. In the case of the values fulfilling the conditions, the filters will become active in the view, providing either one or multiple line, pattern, and transparency overrides of the components (see Figure 28). Using these filters provides a quick and straightforward adaptation of the visualization of the given data. Of importance is the fact that these filter overrides will apply only specific to a certain view in the Revit user interface and do leave the global visibility of the components unaffected.



Filters

Filters

Rule-based Filters
- Condition State Transparency
- CS1
- CS2
- CS3
- CS4
- CS5
- GreyElements
- Int. 1-5
- Int. 6-10
- Int. 11-15
- Int. Rehabilitation
- Int. Renewal
- Int. Replacement
- Intervention Period Transparency
- Intervention Type Transparency
- Pos. less 4 h
- Pos. less 6 h

Categories

Select one or more categories to be included in the filter. Parameters common to these categories will be available for defining filter rules.

Filter list: <show all>

□ Hide un-checked categories

☑ Abutments
□ Air Terminals
□ Alignments
□ Analytical Links
□ Analytical Members
□ Analytical Nodes
□ Analytical Openings
□ Analytical Panels
□ Analytical Pipe Connections
□ Analytical Spaces
□ Analytical Surfaces

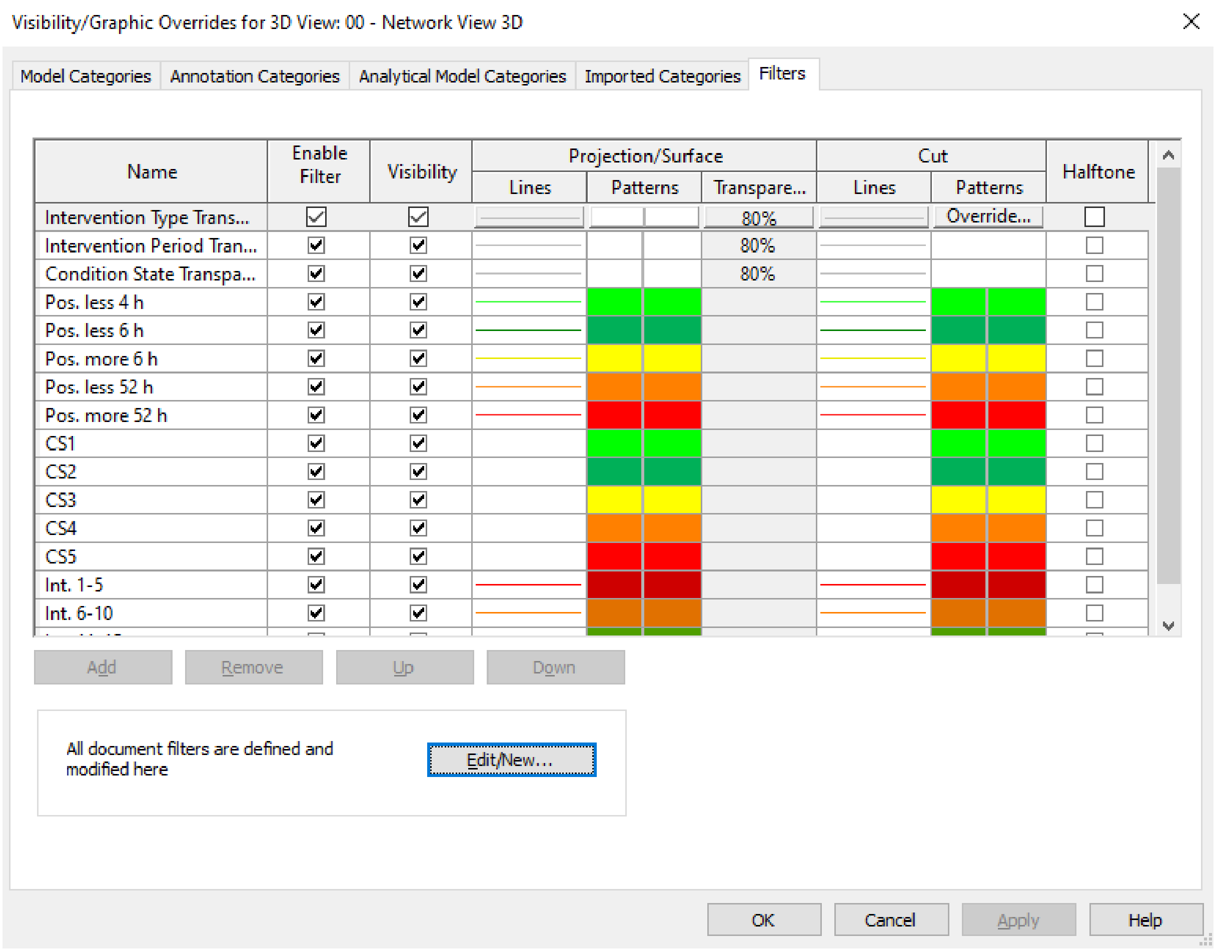
Check All — Check None

Filter Rules

AND (All rules must be tr... — Add Rule — Add Set
All Select... — Condition State Filter
equals — CS 5
All Select... — Most Likely Condition State
does not... — CS 5

AND (All rules must be tr... — Add Rule — Add Set
All Select... — Condition State Filter
equals — CS 2 and worse
All Select... — Most Likely Condition State

Figure 27: View filters rules of a view in Revit.


Visibility/Graphic Overrides for 3D View: 00 - Network View 3D
Model Categories
Annotation Categories
Analytical Model Categories
Imported Categories
Filters

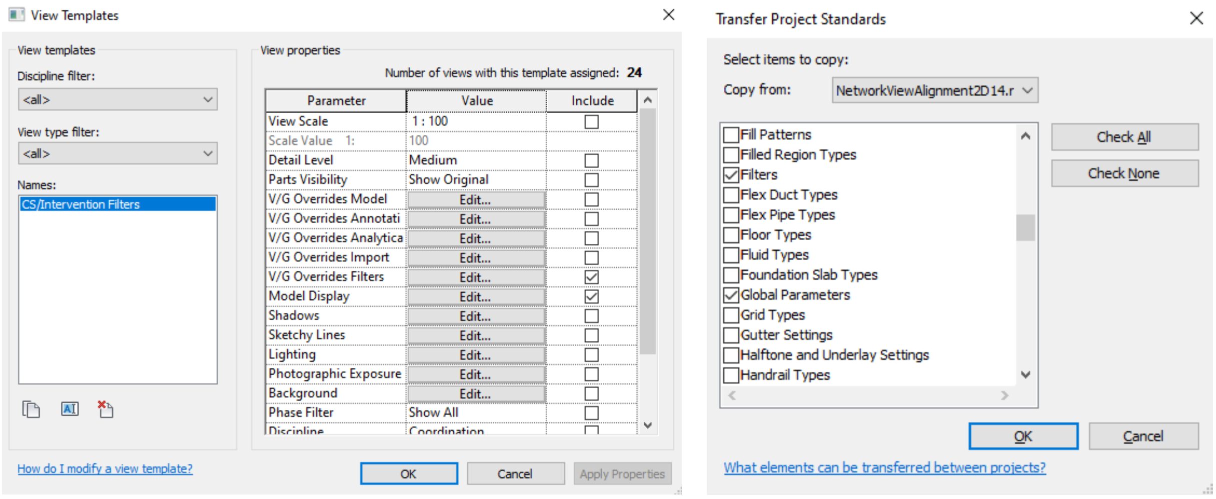
Name	Enable Filter	Visibility	Projection/Surface			Cut		Halftone
			Lines	Patterns	Transpare...	Lines	Patterns	
Intervention Type Trans...	☑	☑			80%		Override...	□
Intervention Period Tran...	☑	☑			80%			□
Condition State Transpa...	☑	☑			80%			□
Pos. less 4 h	☑	☑	—	🟩🟩		—	🟩🟩	□
Pos. less 6 h	☑	☑	—	🟩🟩		—	🟩🟩	□
Pos. more 6 h	☑	☑	—	🟨🟨		—	🟨🟨	□
Pos. less 52 h	☑	☑	—	🟧🟧		—	🟧🟧	□
Pos. more 52 h	☑	☑	—	🟥🟥		—	🟥🟥	□
CS1	☑	☑		🟩🟩			🟩🟩	□
CS2	☑	☑		🟩🟩			🟩🟩	□
CS3	☑	☑		🟨🟨			🟨🟨	□
CS4	☑	☑		🟧🟧			🟧🟧	□
CS5	☑	☑		🟥🟥			🟥🟥	□
Int. 1-5	☑	☑	—	🟥🟥		—	🟥🟥	□
Int. 6-10	☑	☑	—	🟧🟧		—	🟧🟧	□

All document filters are defined and modified here

Figure 28: View filters overrides of a view in Revit.

For the color overrides provided by the view filters to become visible, it is important that the visual style of the active view is set on either ‘Shaded’ or ‘Consistent Colors’. To facilitate this further setup of the Revit projects, it is suggested to use view templates, which can then be applied to the views as needed (see left-hand side of Figure 29). Using a view template also generalizes the view-specific filter overrides, making the manual setup for every single view, unnecessary.

All other parts of the project setup, including project parameters, global parameters, and filter rules can be transferred between projects using the ‘Transfer Project Standards’ tool, depicted in the right-hand side of Figure 29. By utilizing this transfer, the proposed methodology can be also applied to already existing BIM models, boosting its usability.



View Templates

View templates

Discipline filter: <all>

View type filter: <all>

Names:

CS/Intervention Filters

View properties

Number of views with this template assigned: **24**

Parameter	Value	Include
View Scale	1 : 100	□
Scale Value 1:	100	
Detail Level	Medium	□
Parts Visibility	Show Original	□
V/G Overrides Model	Edit...	□
V/G Overrides Annotati	Edit...	□
V/G Overrides Analytica	Edit...	□
V/G Overrides Import	Edit...	□
V/G Overrides Filters	Edit...	☑
Model Display	Edit...	☑
Shadows	Edit...	□
Sketchy Lines	Edit...	□
Lighting	Edit...	□
Photographic Exposure	Edit...	□
Background	Edit...	□
Phase Filter	Show All	□
Discipline	Coordination	□

How do I modify a view template?

OK | Cancel | Apply Properties

Transfer Project Standards

Select items to copy:

Copy from: NetworkViewAlignment2D14.r

- □ Fill Patterns
- □ Filled Region Types
- ☑ Filters
- □ Flex Duct Types
- □ Flex Pipe Types
- □ Floor Types
- □ Fluid Types
- □ Foundation Slab Types
- ☑ Global Parameters
- □ Grid Types
- □ Gutter Settings
- □ Halftone and Underlay Settings
- □ Handrail Types

Check All | Check None

OK | Cancel

What elements can be transferred between projects?

Figure 29: A created view template (left) and the transfer project standards dialog box (right).

Model creation

In the following, the procedure of how the workflow of creating a 3D model containing the assets of a particular bridge might be applied in a concrete environment is introduced in several steps.

Selection of the proper families

The first step in creating a 3D parametric BIM model using the family catalog is to select the best-fitting families to use for the components. Those can be obtained from the family catalog, which can be expanded if needed. Figure 30 presents some of the available bridge object types from which the families can be selected. The superstructure components are included in the upper part, whereas the substructure components can be found in the lower section of the figure.

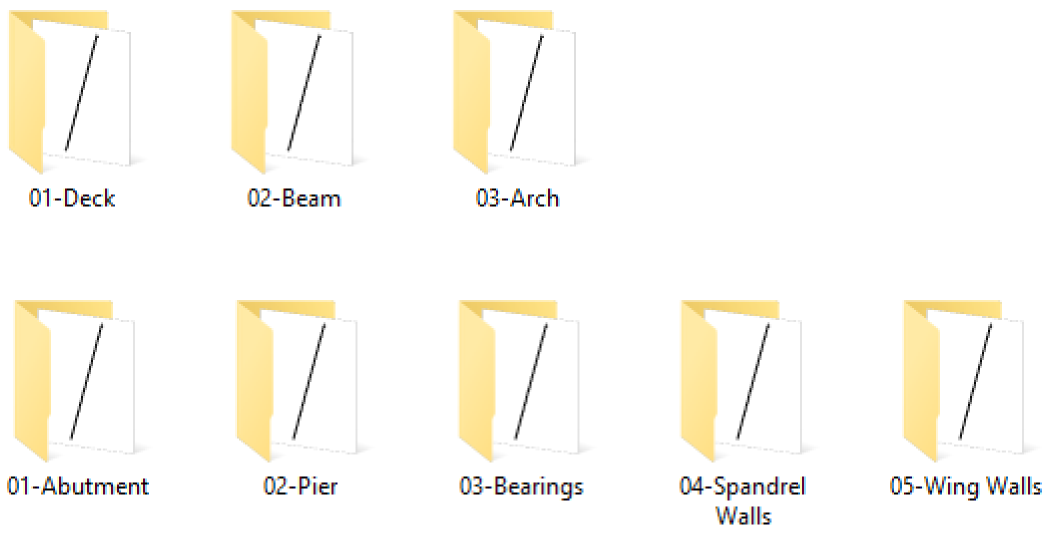


Figure 30: Bridge object types in the family catalog (superstructure and substructure).

Various families can be selected from by choosing the foundations object type inside the bridge domain. These families are presented in Figure 31, showing different types of foundations that have been used throughout the course of this Thesis.

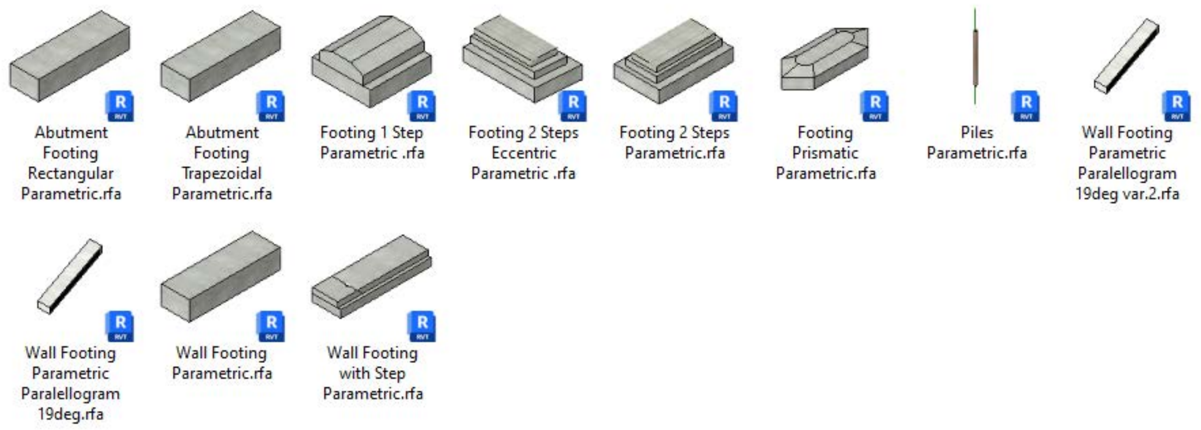
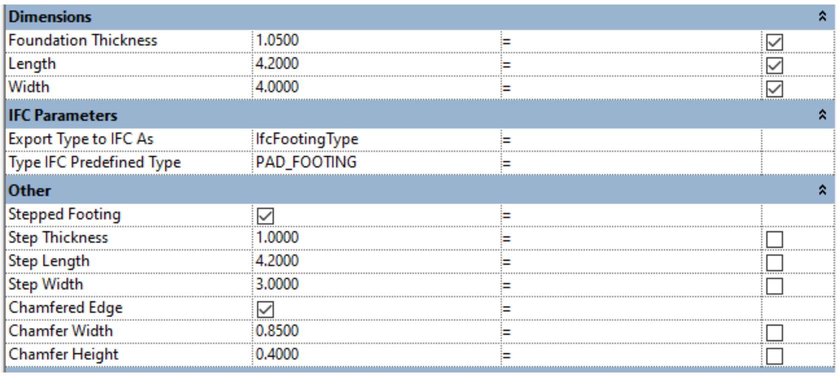

Abutment Footing Rectangular Parametric.rfa
Abutment Footing Trapezoidal Parametric.rfa
Footing 1 Step Parametric .rfa
Footing 2 Steps Eccentric Parametric .rfa
Footing 2 Steps Parametric.rfa
Footing Prismatic Parametric.rfa
Piles Parametric.rfa
Wall Footing Parametric Paralellogram 19deg var.2.rfa
Wall Footing Parametric Paralellogram 19deg.rfa
Wall Footing Parametric.rfa
Wall Footing with Step Parametric.rfa

Figure 31: The available foundation families as part of the family catalog.

For selecting which objects need to be created, the specific families are available for selection. This decision will be made based on the needed parameters for the component. Figure 32 shows the structural foundation presented in the construction drawings and the parameters of the most fitting family. In this case, because of the step and chamfered edges of the structural foundation, the chosen family 'Footing 1 Step Parametric' would be the optimal fit.



Dimensions			
Foundation Thickness	1.0500	=	☑
Length	4.2000	=	☑
Width	4.0000	=	☑
IFC Parameters			
Export Type to IFC As	IfcFootingType	=	
Type IFC Predefined Type	PAD_FOOTING	=	
Other			
Stepped Footing	☑	=	
Step Thickness	1.0000	=	□
Step Length	4.2000	=	□
Step Width	3.0000	=	□
Chamfered Edge	☑	=	
Chamfer Width	0.8500	=	□
Chamfer Height	0.4000	=	□

Figure 32: A foundation and its most fitting parameters.

Creation of the required types

After importing the most fitting families for all components into the project, adjustments are required to be made, in order for the objects to fit the specific bridge that needs to be represented. This is done in a lower level of the Revit element hierarchy, the type level, where objects have the same parameters they need input for, but have different input values from each-other. In Figure 33, two elements that are part of the same family, 'Footing 1 Step Parametric', have different types 'Chamfered Edges 4.2 x 4.0 x 1.05 m' and '4.2 x 4.0 x 1.05 m' because of the distinct input values that are provided.

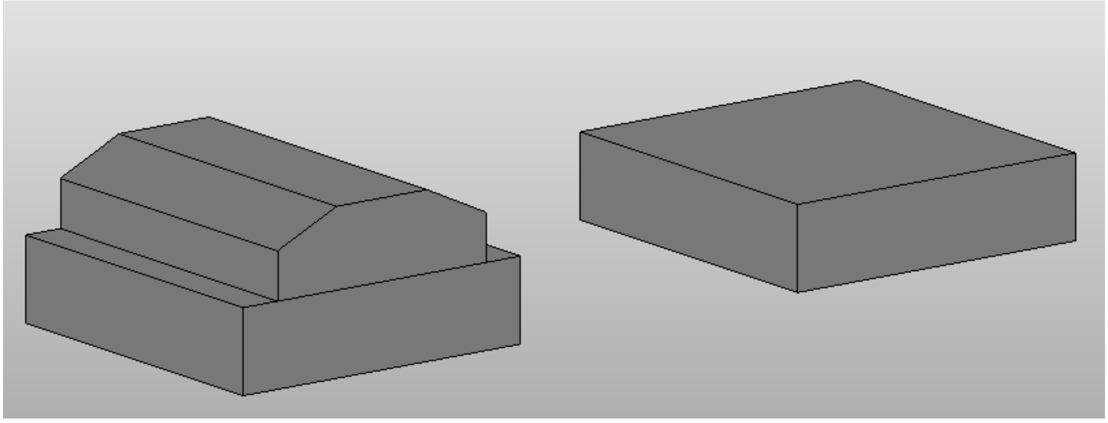
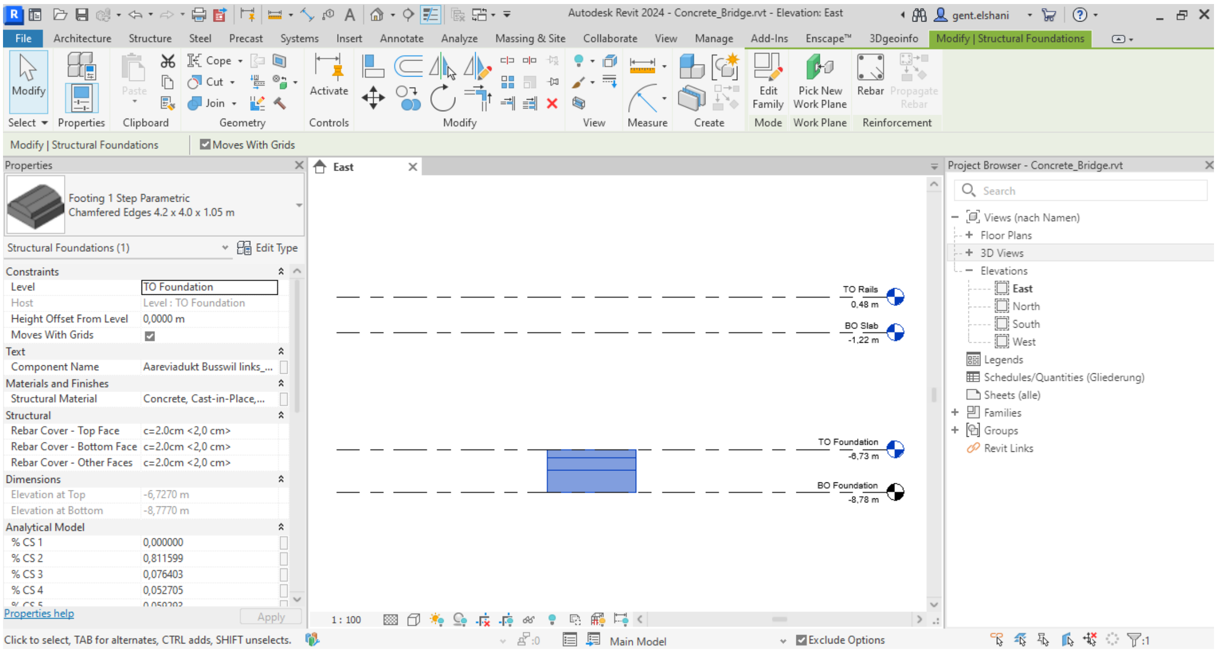


Figure 33: Two types of the same family with different parameter values.

The specific types can be created using the information about the geometry from the construction drawings or any other source of geometric information. Succeeding the creation of all the needed types for a specific bridge, the placement in the modeling space will follow.

Adjusting the components

After placing the loadable families into the workspace, the creation and adaptation of the levels in the project should be made in order to represent characteristic levels of the project, like the bottom of the foundations or the top of the bridge deck. The components should then be assigned to these levels, as shown for example in Figure 34, selecting the appropriate elevations based on the destination.



Modify | Structural Foundations ☑ Moves With Grids

Properties

Footing 1 Step Parametric
Chamfered Edges 4.2 x 4.0 x 1.05 m

Structural Foundations (1) Edit Type

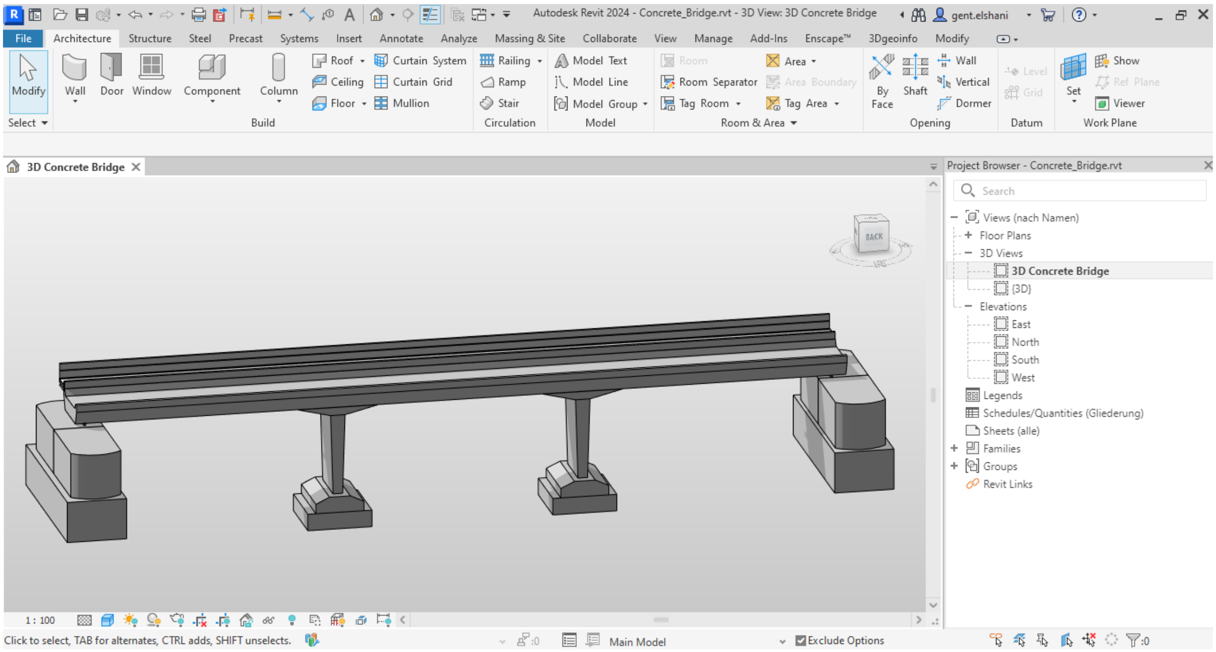
Constraints	
Level	TO Foundation
Host	Level : TO Foundation
Height Offset From Level	0,0000 m
Moves With Grids	☑
Text	
Component Name	Aareviadukt Busswil links_...
Materials and Finishes	
Structural Material	Concrete, Cast-in-Place,...
Structural	
Rebar Cover - Top Face	c=2.0cm <2,0 cm>
Rebar Cover - Bottom Face	c=2.0cm <2,0 cm>
Rebar Cover - Other Faces	c=2.0cm <2,0 cm>
Dimensions	
Elevation at Top	-6,7270 m
Elevation at Bottom	-8,7770 m
Analytical Model	
% CS 1	0,000000
% CS 2	0,811599
% CS 3	0,076403
% CS 4	0,052705
% CS 5	0,050202

Properties help Apply

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 34: Assigning a component to a level in Revit.

Finally, a planar alignment has to be carried out in order for the components to be positioned in a correct position in accordance with the other attached components. This is done using the align function in Revit, significantly facilitating the translation and rotation of the components. An example screenshot is provided in Figure 35, showing the final 3D model of the desired bridge.



Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 35: The created bridge model using the family catalog.

With this digital model at hand, further analysis can be carried out, always considering the approximations and correctness in creating the geometry in accordance with the needed use case. In the scope of this Thesis, the BIM models will be utilized for visualization and decision-making purposes, which will is described in more detail in Section 2.2.4 and Section 2.3.

In order to efficiently execute the proposed visualization and overall methodology, certain prerequisites have to be included in the BIM model projects in the form of parameters and filters.

2.2 Visual programming

As a visual programming interface, the representation of syntactic structure infused with semantic information is expressed using visual means as a support. This forms a formal language with visual syntax and semantics [46].

The use cases of visual programming range over a wide variety of processes in the built environment industry. The primary cause is to automate specific tasks, providing a more intuitive way, as opposed to conventional code written in a text format. The specific field where it can be applied ranges from general aid in architectural design to data integration and exchange [47], as well as infrastructure [48].

Visual programming tools generally work by having an interface where blocks of pre-defined code, called Nodes, are connected to each other through connections called Wires. In this way, the output of one node is used as an input for the next node connected through a wire with it. Usually, a node block can have multiple wires coming out of it, but only one wire serves as an input for every entry. Figure 36 shows the user interface of a commonly used visual programming tool, Dynamo for Revit. The main separation of the user window is between the node library on the left, where all the pre-defined and imported nodes are shown, and the bigger right-hand side, which represents the canvas or the surface where all the nodes are being placed.

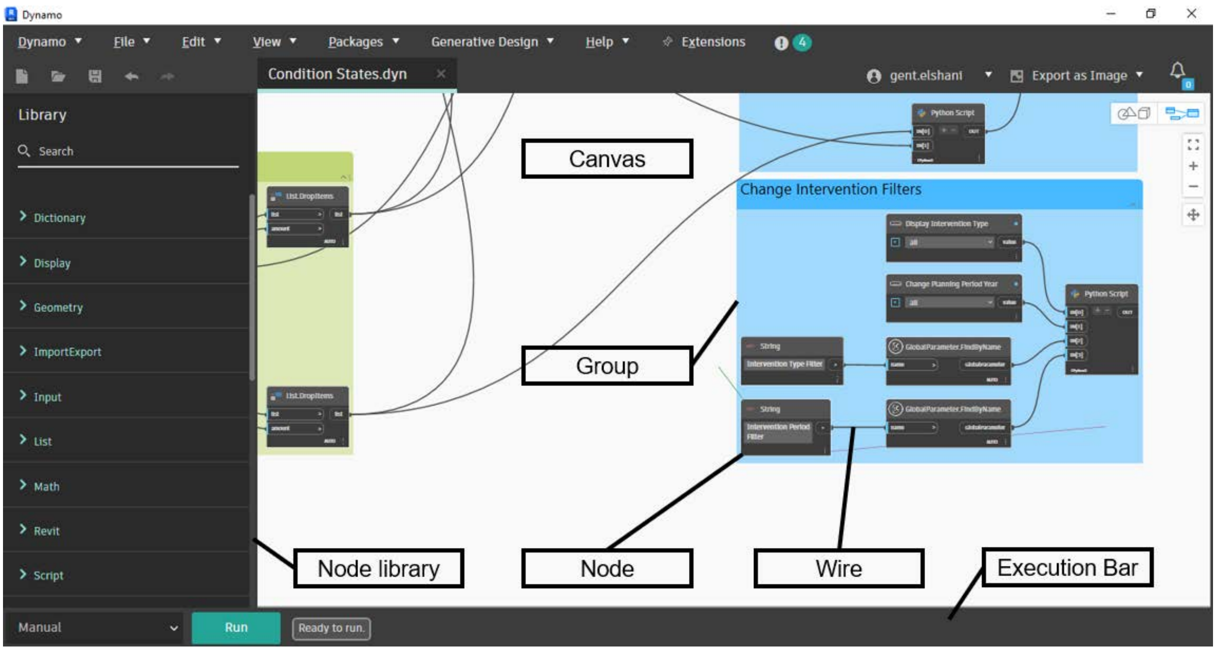


Figure 36: User-interface of Dynamo for Revit.

Many different visual programming languages and interfaces exist, where the selection depends on the BIM authoring tool being used and the user-friendliness of the tool. Table 5 presents some of the most utilized visual programming applications, along with the manufacturer and programming language employed in the interface.

Table 5: Several visual programming interfaces with their additional data [34].

Name	Manufacturer	Programming language	Complying software
Dynamo	Autodesk	C#, Iron Python	Standalone, Revit
Grasshopper 3D	Open Source	C++, C#, Python	Rhinoceros3D
Allplan Visual Scripting	Allplan	Python	Allplan
Marionette	Vectorworks	Python	Vectorworks
PyFlow	Open Source	Python	Standalone

2.2.1 Dynamo for Revit

For this Master’s Thesis, the tool Dynamo for Revit has been used for several automation steps. In the BIM authoring tool Revit, it is a preinstalled interface which can be executed whenever needed. One of its greatest advantages is the direct connection with the Revit interface, making it possible to easily access and export elements to and from it.

The nodes provided in Dynamo can be categorized into the following different types:

- Built-in nodes – These nodes are supplied together with the software and developed by Autodesk to provide the most efficient workflow in Dynamo.
- Custom nodes – Here, additionally created nodes are included to fulfil the need of specific tasks that are not included in the standard library. There are a lot of open-source custom node libraries accessible through the extensions option.

- Python nodes – A special categorization for these nodes is introduced since these nodes include the ability to perform basic Python 3 functionalities, such as conditional statements and loops, as well as a direct connection to the Revit API. This link allows elements to be modified on a much lower level than the software usually provides.

Moreover, Dynamo uses a concept called parallelism [49], where multiple operations are carried out simultaneously, making the runtime shorter and the workflow more efficient. In this way, using an interpreted programming language, such as Python, is made possible inside the plugin Dynamo of the software Revit.

2.2.2 Code review

In the scope of this Thesis, a Dynamo script has been developed to automate the dynamic assignment of condition state and intervention values to the corresponding components, as well as to provide a visualization using a color-coded 3D model. In the previous Master's Thesis, a script was set up with some functionalities, which have been optimized following the steps presented in Section 2.2.3 [30]. In Figure 37 the pre-optimized script is shown, which for clarity is reduced to a schematic view with the basic actions resulting from a group of nodes.

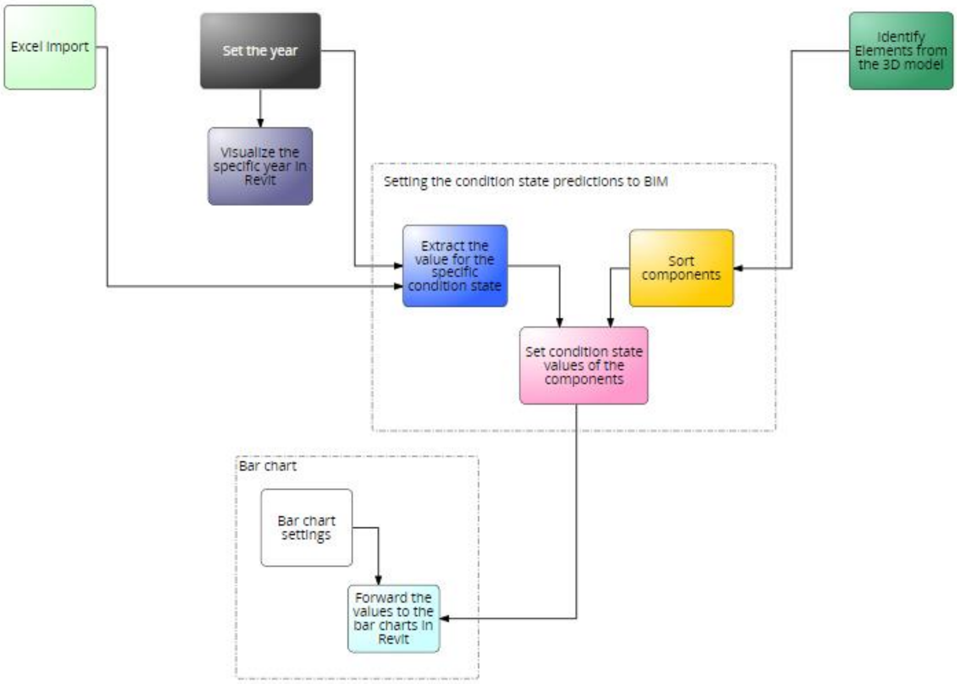

Excel import
Set the year
Identify Elements from the 3D model
Visualize the specific year in Revit
Setting the condition state predictions to BIM
Extract the value for the specific condition state
Sort components
Set condition state values of the components
Bar chart
Bar chart settings
Forward the values to the bar charts in Revit

Figure 37: The schematic view of the pre-optimized Dynamo script [30].

Following the functions in Figure 37, the script can be divided into three parts based on the data that is being handled.

Data import

The first step of the script execution includes the import of different data into Dynamo. This takes place for the Excel data including the information about the condition states of the components, which has been generated from the prediction algorithm developed by the Chair of Infrastructure Management, and the model data, which gets imported directly from the project currently open in Revit. As a user input, the year value must be supplied in order to view the year of interest in which the condition states of the specific components will be shown. This year input also changes the text next to the 3D model, showing the updated value after every script run.

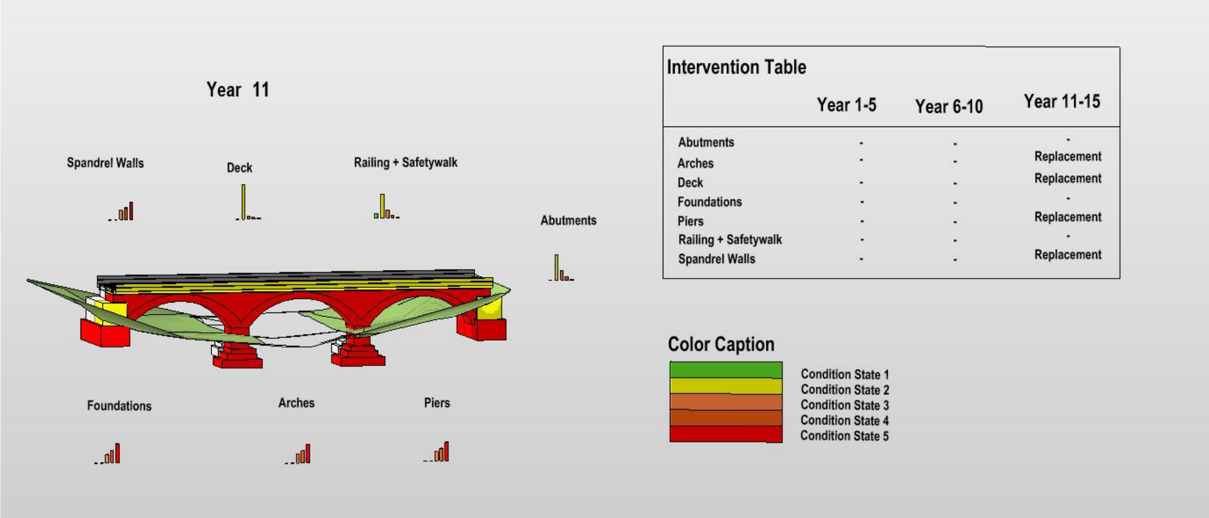
Value assignment

After obtaining the input values, the script extracts the exact value for the desired year and sorts the model elements based on their ‘Family Type’ parameter into the treated bridge components. These components are then assigned the corresponding condition state values into their parameters for the condition states 1 to 5.

Output visualization

By utilizing the set filters in the Revit 3D view, the components change color based on a threshold value of the individual condition states. In this way, an easy overview of the condition states can be obtained. Additionally, bar charts have been placed around the model to represent the likelihood of each condition state found in the provided year. Finally, a table has been placed next to the model to represent the interventions that are estimated to be carried out in the specific planning periods.

An example result can be seen in Figure 38, where the bridge colors conform to the condition state of the components. The bar charts around the model provide further insight about the percentage of the other condition states, which cannot be seen in the 3D model. The intervention table presents information on the planned actions and was manually added for each bridge.



Intervention Table

	Year 1-5	Year 6-10	Year 11-15
Abutments	-	-	-
Arches	-	-	Replacement
Deck	-	-	Replacement
Foundations	-	-	-
Piers	-	-	Replacement
Railing + Safetywalk	-	-	-
Spandrel Walls	-	-	Replacement

Figure 38: Visualization of the condition states and interventions from the previous Thesis [30].

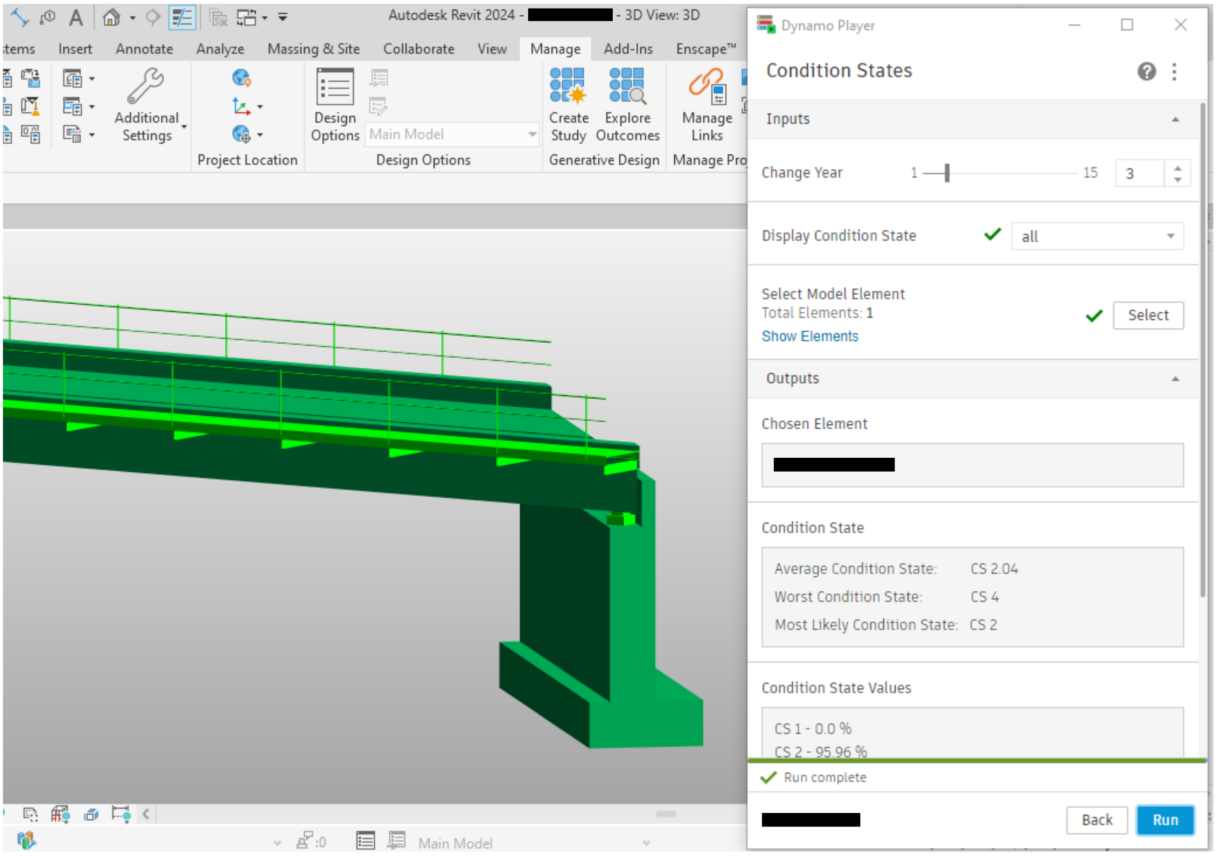
2.2.3 Code improvements

The script previously developed and described in the preceding section has been optimized in several different aspects. The key enhancements have been accomplished by tackling some of the presented script’s issues.

Firstly, as observed, the Dynamo script appears rather large and difficult to navigate through its composing nodes. The reason behind this is that for each separate component the BIM model contains, a set of nodes has been inserted, specific to only that component. In this way, depending on the number of components treated, the script range grows exponentially. Since in the examined line several bridge types exist, new scripts for different types of components have been previously created.

The aim in the optimization process is to create one Dynamo script that would fit any bridge type and would make the manual procedure of creating scripts based on the specific components in that bridge, obsolete. Additionally, the approach of placing any kind of annotation, such as the year text above the bridge, the bar charts around it, and the intervention table, was discontinued. The reason behind it is that they make the overall usefulness of a 3D visualization less valuable since the view has to be locked in a certain position for the annotation to show correctly, making any navigation, such as rotating and cutting the 3D model, impossible.

The generalization of the script is done by using the 'Component Name' parameter as an identifier to fulfill the mapping between entities. This parameter consists of the bridge name and the component, connected by an underscore. Apart from inducing a generalization into the Dynamo script, the additional user interface of the Dynamo Player, included next to the 3D model, has been integrated (see Figure 39). Several inputs and outputs to this interface are discussed in this section. Further information on how this UI works and what features it includes can be found in Section 2.2.4.



Condition States

Change Year 1 — 15 3

Display Condition State ✓ all

Select Model Element
Total Elements: 1 ✓

Chosen Element
[redacted]

Condition State
Average Condition State: CS 2.04
Worst Condition State: CS 4
Most Likely Condition State: CS 2

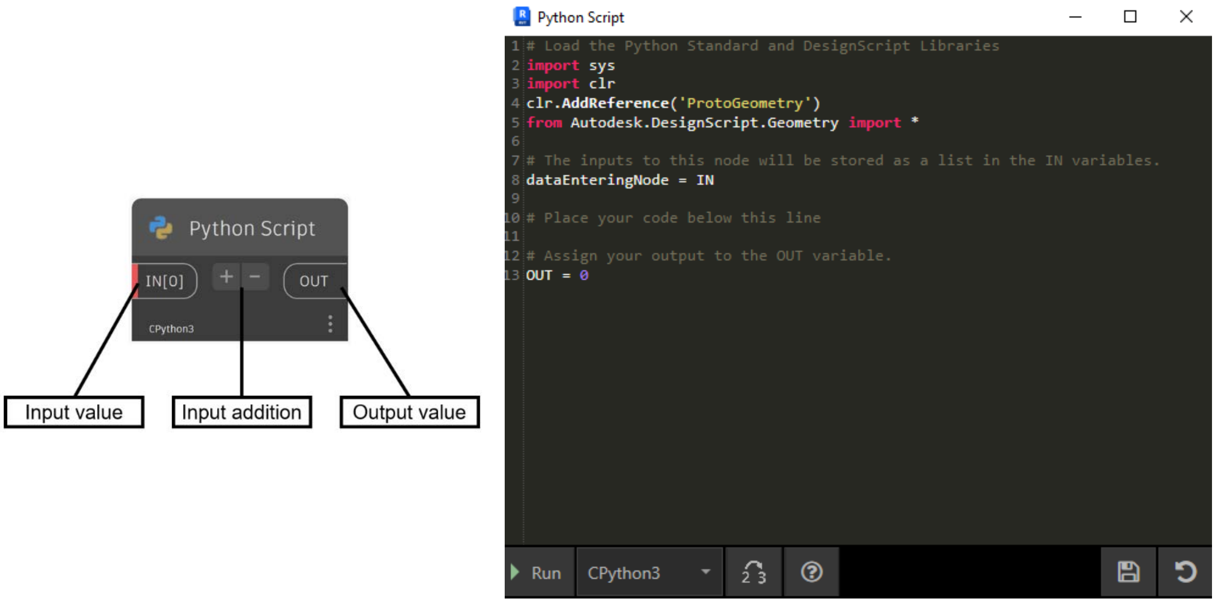
Condition State Values
CS 1 - 0.0 %
CS 2 - 95.96 %

✓ Run complete

[redacted]

Figure 39: Dynamo Player along with the 3D model in Revit.

As a key part of the newly developed script, Python nodes have been used extensively in Dynamo. As presented in the left-hand side of Figure 40, they consist of at least one input value, which is used inside the Python script to perform certain actions, and one output value from the Python script, which can be further utilized in connection to Dynamo nodes. The main advantage is the accessibility of libraries in Python, where conventional Python libraries such as Numpy and Pandas can be used for a significant part without any major limitations. More importantly, specific libraries, including Autodesk Designscript, for creating Dynamo geometry inside the Python node, and Autodesk Revit DB, for accessing and using functionalities of the Revit API, can be applicable. The default interface, where the Python script can then be inserted, is shown in the right-hand side of Figure 40.



```python
# Load the Python Standard and DesignScript Libraries
import sys
import clr
clr.AddReference('ProtoGeometry')
from Autodesk.DesignScript.Geometry import *

# The inputs to this node will be stored as a list in the IN variables.
dataEnteringNode = IN

# Place your code below this line

# Assign your output to the OUT variable.
OUT = 0
```

Figure 40: Python node with its consisting parts (left) and its IDE (right).

As an output of optimizing the Dynamo script, the following code has been created (see Figure 41). Each block of nodes performing one action was grouped together, presenting a more simplified overview of the script. These parts are explained separately in the following segment.

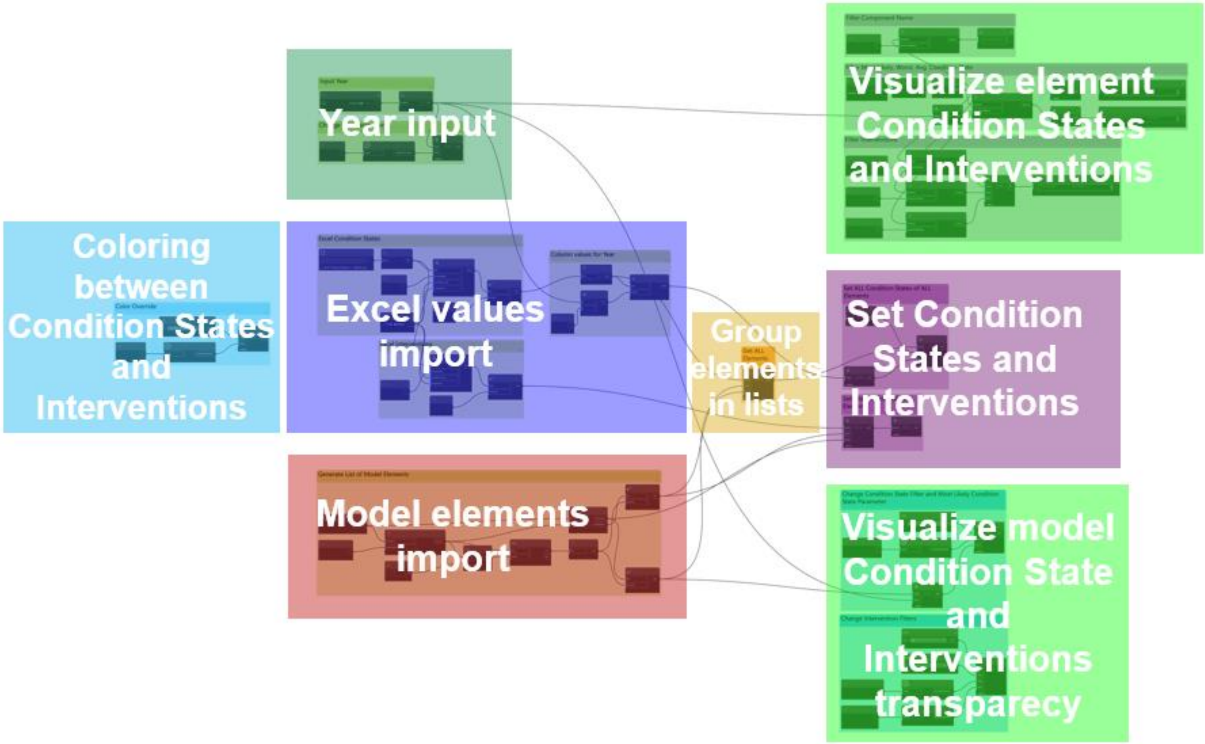

Year input
Visualize element Condition States and Interventions
Coloring between Condition States and Interventions
Excel values import
Group elements in lists
Set Condition States and Interventions
Model elements import
Visualize model Condition State and Interventions transparecy

Figure 41: A simplified view of the optimized Dynamo script.

Year input

This group of nodes (see Figure 42) allows selecting and inputting the specific year in which the condition states and interventions want to be seen. By supplying the value through a slider, the selection from a user perspective gets much easier since this value is also expected as input through the Dynamo Player.

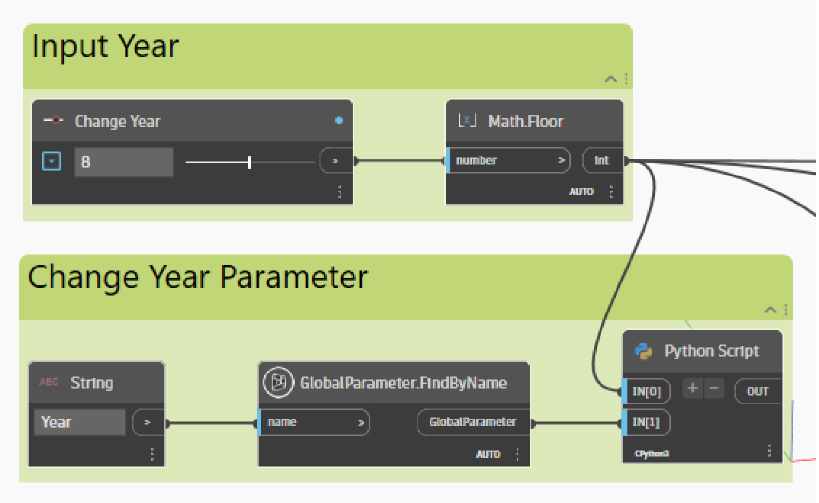
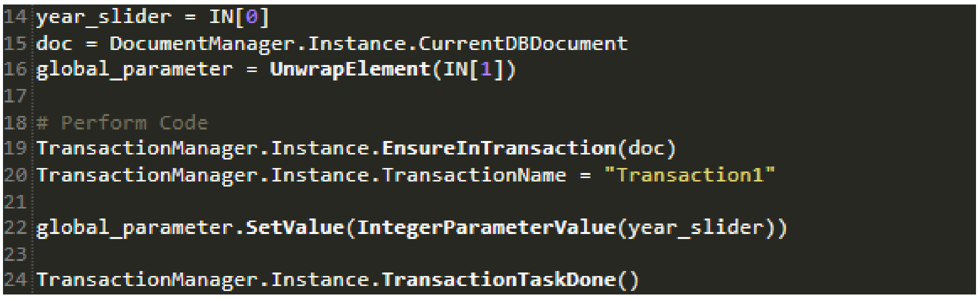


Figure 42: Nodes used for the year input and changing the year parameter.

A crucial part of this group of nodes is the Python script, which assigns the desired value of the year to a global parameter. This happens by inputting the value and parameter into the Python node and then running the script. A snippet of it can be seen in Figure 43, where one of the key parts of effectively executing this code is the unwrapping of the Revit element, which is the global parameter in this case. By executing this unwrapping, the element can be fully used inside Dynamo by accessing the Revit API and acquiring all the methods and values assigned to the element. A lower-level editing can be carried out effectively in such a manner. Another essential factor in the execution of the Python script is the inclusion of transactions when the object values of an element need to be accessed and modified. In this Dynamo script, the TransactionManager has been utilized. This method can be imported from the RevitServices.Transactions library and handles the transactions in an efficient way, offering the user the opportunity to choose the transaction name, called Transaction1 in this case.

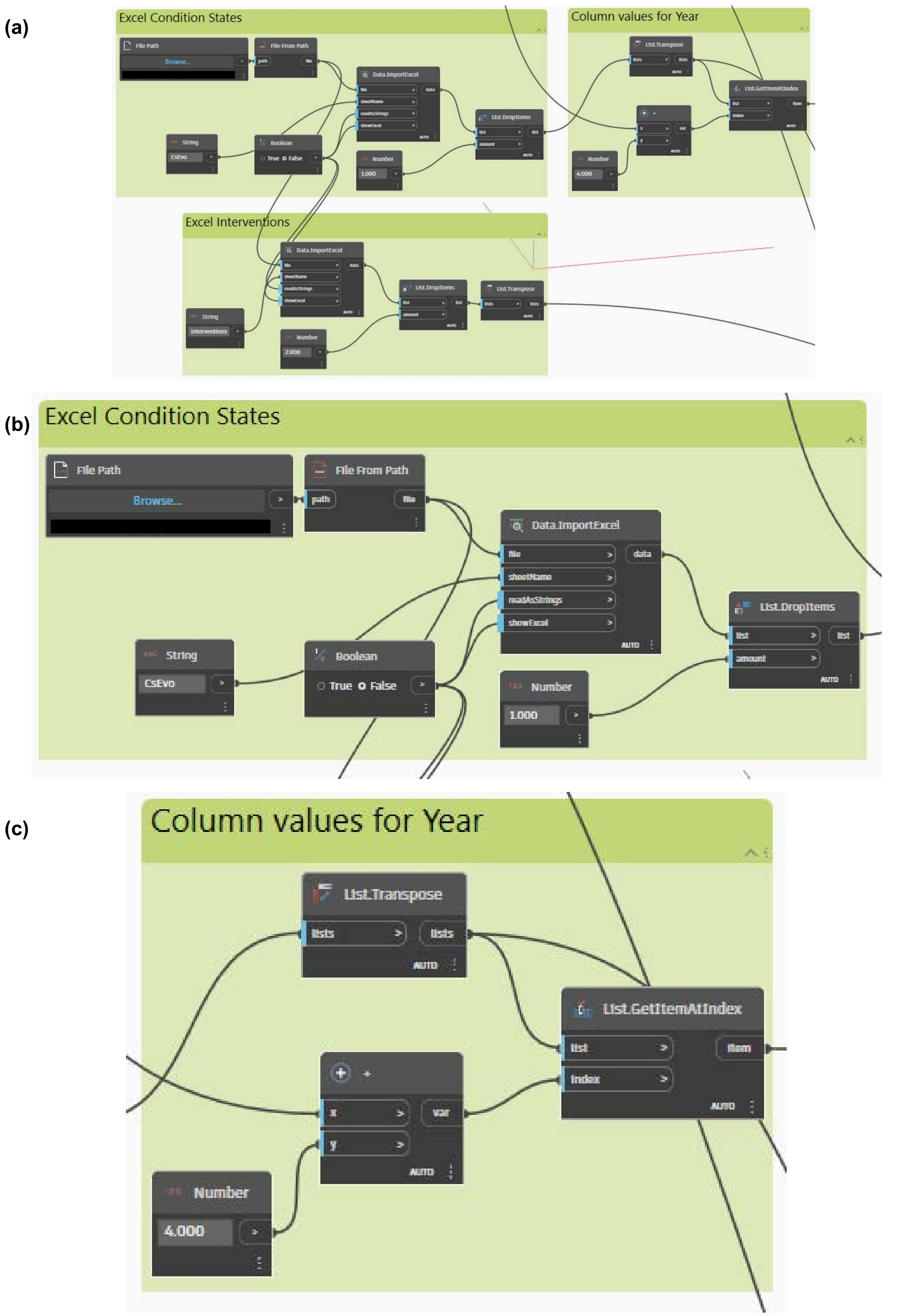


```
year_slider = IN[0]
doc = DocumentManager.Instance.CurrentDBDocument
global_parameter = UnwrapElement(IN[1])

# Perform Code
TransactionManager.Instance.EnsureInTransaction(doc)
TransactionManager.Instance.TransactionName = "Transaction1"

global_parameter.SetValue(IntegerParameterValue(year_slider))

TransactionManager.Instance.TransactionTaskDone()
```

Figure 43: A snippet of the Python code showing the elements unwrapping and transactions in Dynamo.

Excel values import

By executing this group of nodes (see Figure 44), a successful input of all the values from the Excel sheet containing the data on the deterioration of the condition states over time and the estimated interventions with their additional information, such as the intervention type, period, and cost can be inserted into Dynamo for further use.



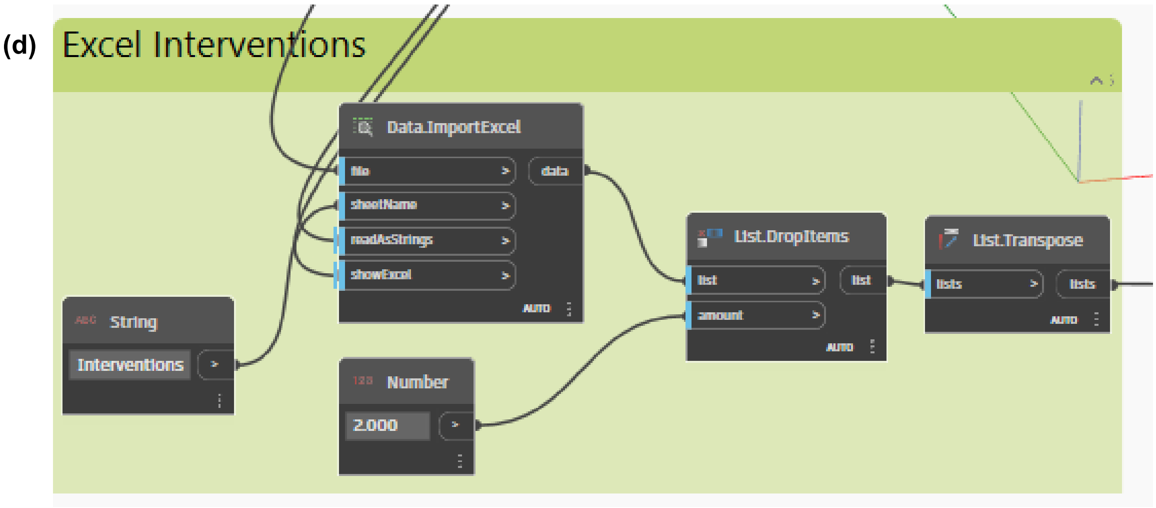


Figure 44: Nodes used for importing values on condition states and interventions from Excel and creating a list with only the numerical values of the condition states. To provide a clearer visual representation, the nodes (a) have been presented separately, labeled as ‘Excel Condition States’ (b), ‘Column values for Year’ (c), and ‘Excel Interventions’ (d).

The basic functionalities of these nodes are to import all the values from the Excel sheet based on the path where it is located and the sheet names. These values are then stored in lists in Dynamo. Values that do not contain any significant information, for example the row containing the names of all columns, are dropped and not further included in the script. Then, these items are further used in other parts of the Dynamo script. It is essential to point out that the Excel sheets must be in the same order as presented and used in this Thesis for the script to work. If the layout of the Excel sheet changes, the script needs be modified accordingly.

Model elements import

In this part of the Dynamo script (see Figure 45), all elements in the active view of the Revit model are imported into the script for further use. The filtering based on the parameter called ‘Component Name’ is carried out in order to ensure the elements fall under the categories which are covered within the scope of this Thesis. This sorting occurs by checking if the named parameter exists in the elements of the view, and then outputs only elements where this condition is true.

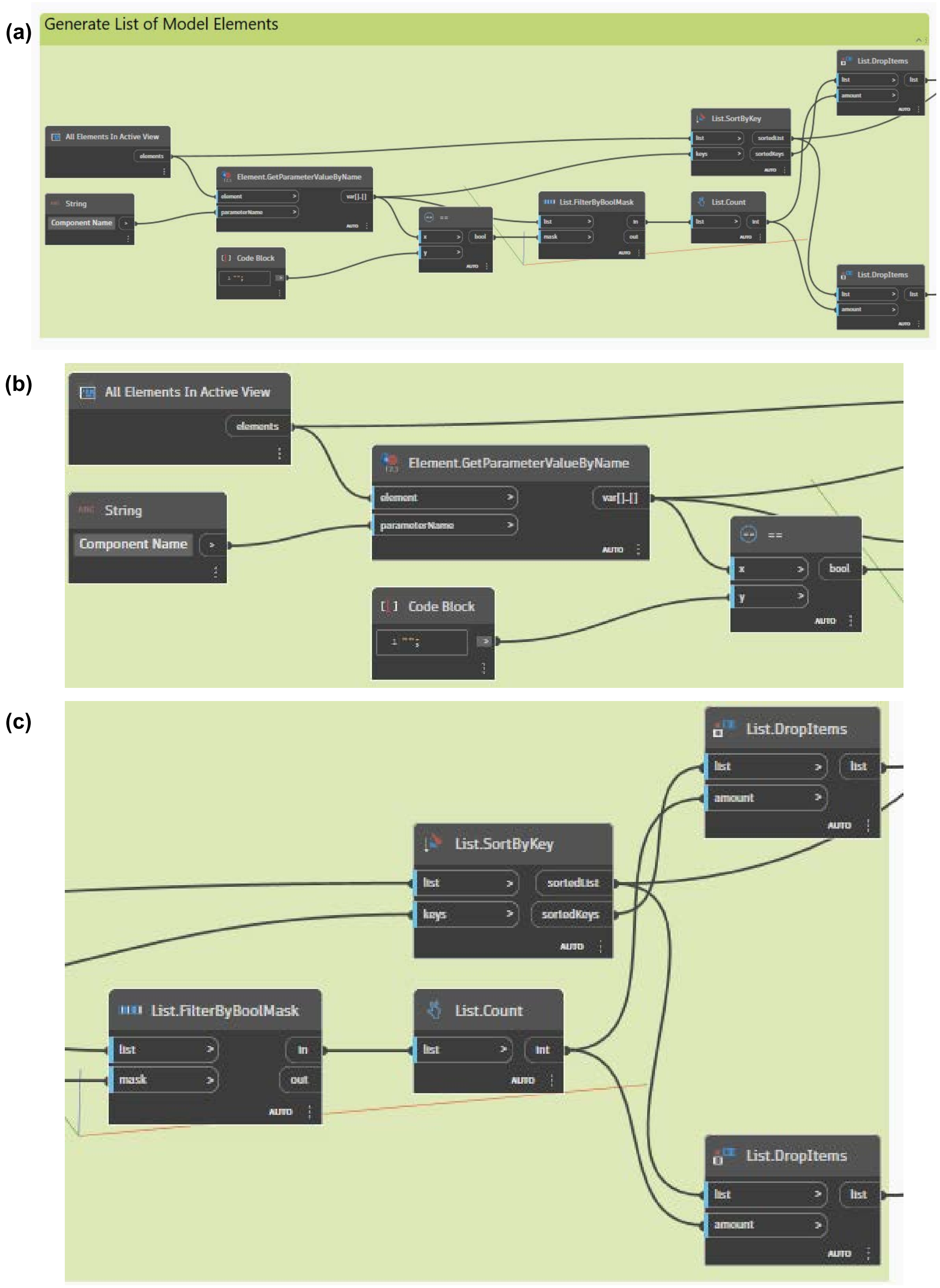
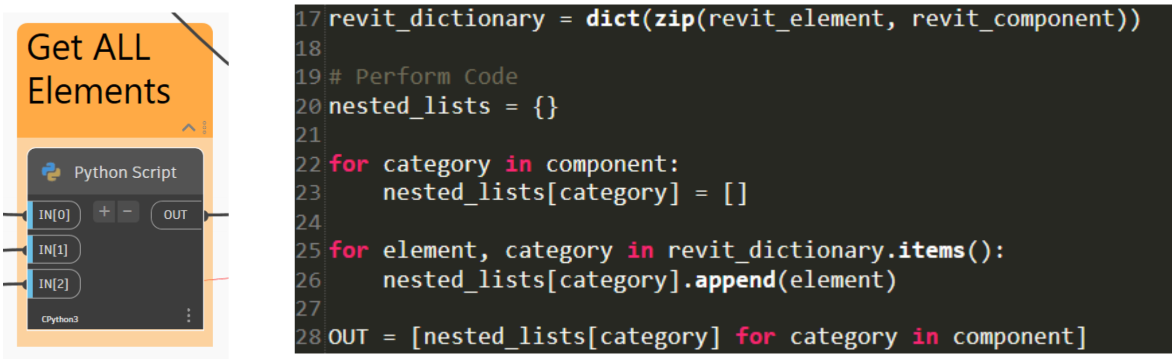


Figure 45: Nodes used for importing model elements from Revit. To enhance the visualization in (a), (b) provides a close-up view of the node's left side, while (c) does the same for the right side.

Additionally, an alphabetical sorting has been carried out to comply with the order of the elements in the Excel sheet. The model elements and their respective 'Component Name' are then stored for further use.

Group elements in lists

This operation is executed by one Python node (see left-hand side of Figure 46) and contains the code needed to create a structure which will be utilized as a key factor in generalizing the Dynamo script. From the previous groups of nodes, the Excel values for the condition states in the given year, the model elements, and the ‘Component Name’ parameter of these elements are expected as input into this node. These elements are then grouped into a nested list, which contains sublists of all the components available in the 3D model (refer to right-hand side of Figure 46). This way, a well-organized list will be produced, which keeps the same structure even if there is an entry in the Excel list, but the specific bridge model does not have any element of that category. In this case, an empty sublist is created, which does not disrupt the order of the elements.



```
revit_dictionary = dict(zip(revit_element, revit_component))

# Perform Code
nested_lists = {}

for category in component:
    nested_lists[category] = []

for element, category in revit_dictionary.items():
    nested_lists[category].append(element)

OUT = [nested_lists[category] for category in component]
```

Figure 46: Python node used for assigning elements into lists (left) and a snippet of the script underlying it (right).

Set condition states and interventions

The set of nodes in Figure 47 is the core part in terms of functionality of the script since it enables the dynamic assignment of values into the model elements, which will finally be used as a basis for further visualization. It is separated into two parts: allocating the (a) condition state values and (b) intervention values into the elements. The main difference between these units is that all bridge components are assigned to a certain condition state as well as a deterioration over time. Meanwhile, interventions are not always foreseen in the treated planning periods, creating the possibility of the components having no intervention information assigned to them. The contrast in information that needs to be assigned to the elements led to a slight variation in the approach to developing certain aspects in this section. The usage of transactions and unwrapped elements inside the Python node is also taken into consideration here to ensure the full functionality.

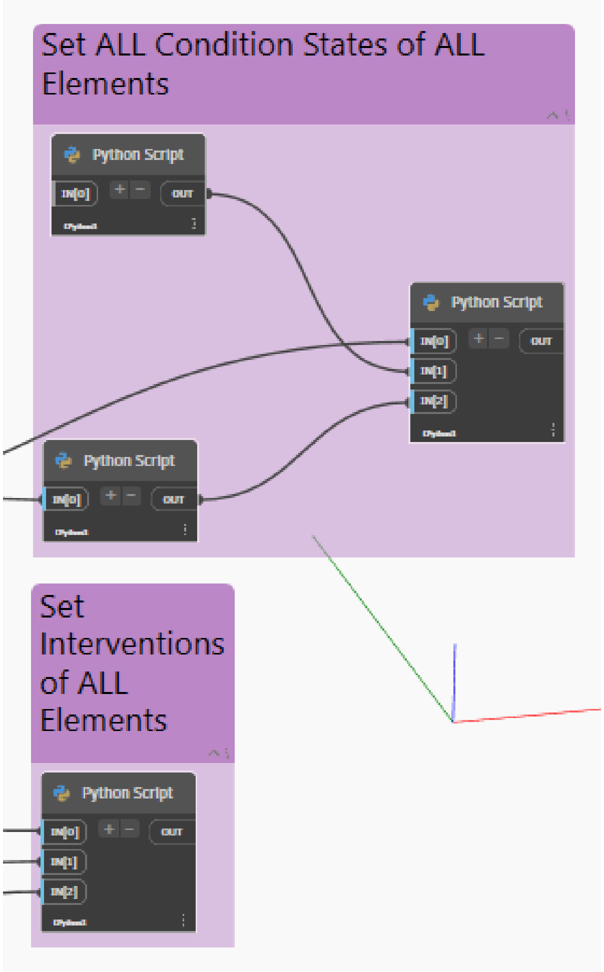
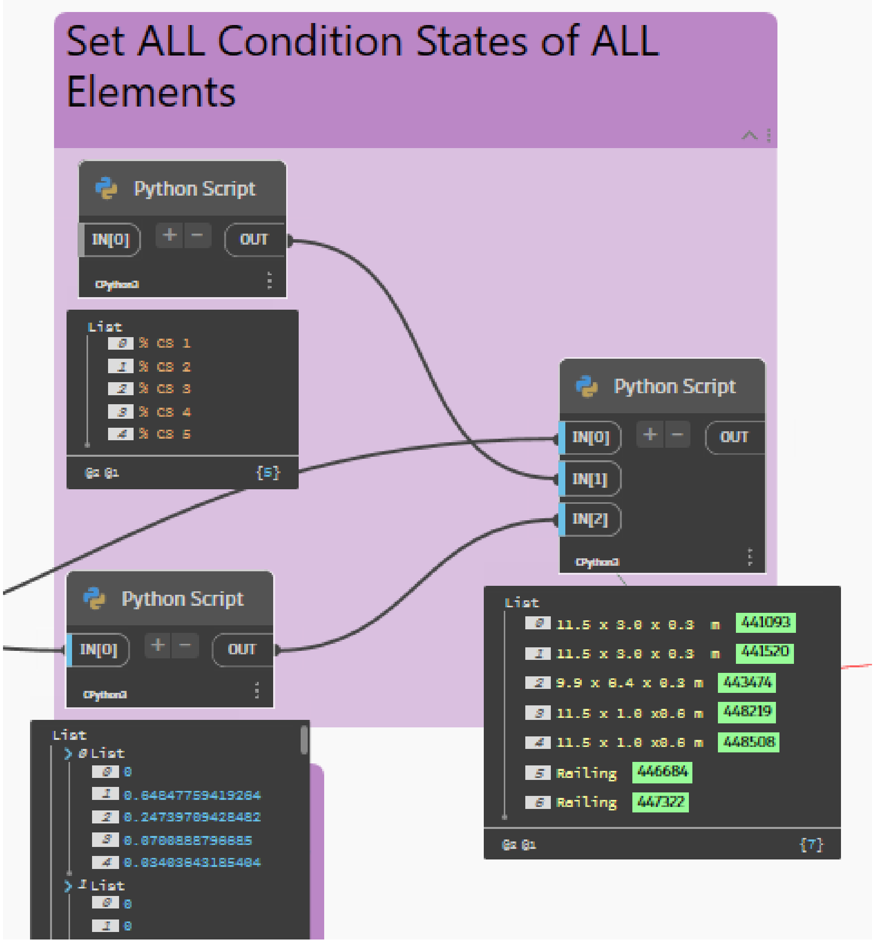


Figure 47: Nodes used for setting the parameter values for the condition states and interventions of the model elements according to the Excel values.

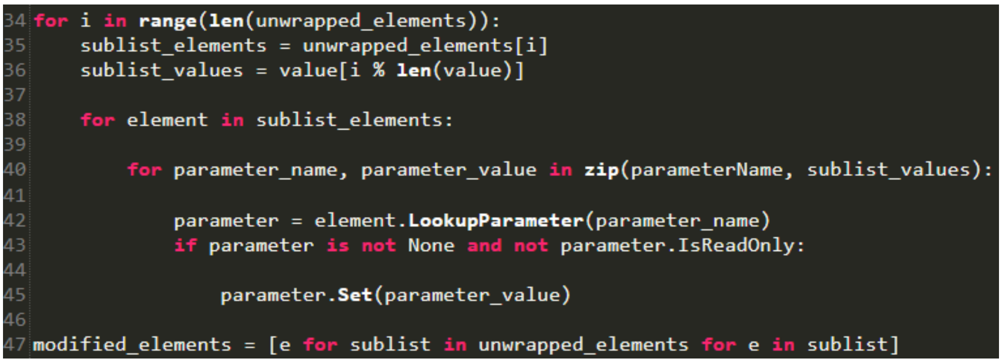
For the assignment of condition state values, a set of three Python nodes have been employed within Dynamo. An example output of the separate nodes can be observed at the top of Figure 48. In the first node, a list containing the parameters' names for each condition state is produced. In the second node, the specific values for the condition states depending on the year input are extracted from the Excel sheet. The returned values are then applied to the third node, part of which can be seen at the lower section of Figure 48. Together with the nested list containing the model elements in this node, all the condition state likelihoods for a certain year are assigned to their corresponding components by utilizing a series of successive loops.



```
List
  0 % CS 1
  1 % CS 2
  2 % CS 3
  3 % CS 4
  4 % CS 5
@2 @1                    {5}
```

```
List
  0 11.5 x 3.0 x 0.3  m  441093
  1 11.5 x 3.0 x 0.3  m  441520
  2 9.9 x 0.4 x 0.3 m  443474
  3 11.5 x 1.0 x0.6 m  448219
  4 11.5 x 1.0 x0.6 m  448508
  5 Railing  446684
  6 Railing  447322
@2 @1                    {7}
```

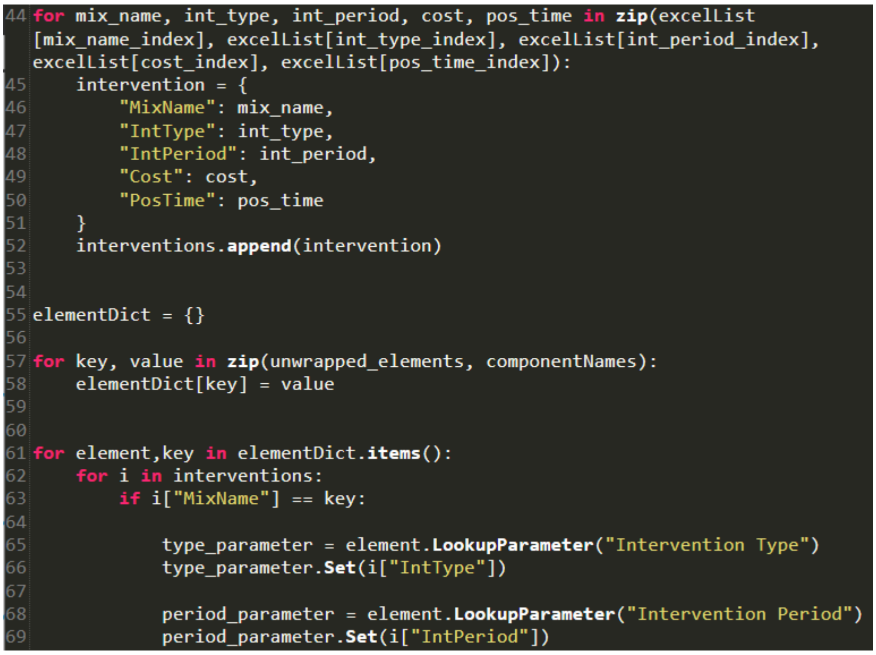
```
List
  > 0 List
      0 0
      1 0.64847759419264
      2 0.24739709428482
      3 0.0700868796685
      4 0.03403843185404
  > 1 List
      0 0
      1 0
```



```python
for i in range(len(unwrapped_elements)):
    sublist_elements = unwrapped_elements[i]
    sublist_values = value[i % len(value)]

    for element in sublist_elements:

        for parameter_name, parameter_value in zip(parameterName, sublist_values):

            parameter = element.LookupParameter(parameter_name)
            if parameter is not None and not parameter.IsReadOnly:

                parameter.Set(parameter_value)

modified_elements = [e for sublist in unwrapped_elements for e in sublist]
```

Figure 48: Nodes used to set the condition state values (top) and a snippet of the Python script where the parameter is obtained and set (bottom).

When it comes to assigning the correct values from the Excel sheet to the model elements regarding interventions, a single Python node has been developed. This node requires the intervention value sheet from Excel and the model elements together with their ‘Component Name’ parameter. In the code snippet from Figure 49, the approach can be interpreted. The aim is to create two lists: (a) one containing a pair of the model elements and their respective ‘Component Name’, as well as (b) the other list including the intervention information of each element where interventions are planned and the ‘Component Name’ identifier. The parameter acts as a key for the element since each element possesses only one ‘Component Name’. The said key is then compared to a value of the created dictionaries within Python, housing data from the Excel sheet, which is accessed through the ‘Component Name’. After comparing the values and finding a match between the identifiers, a loop of assigning all the values to the component parameters is executed.

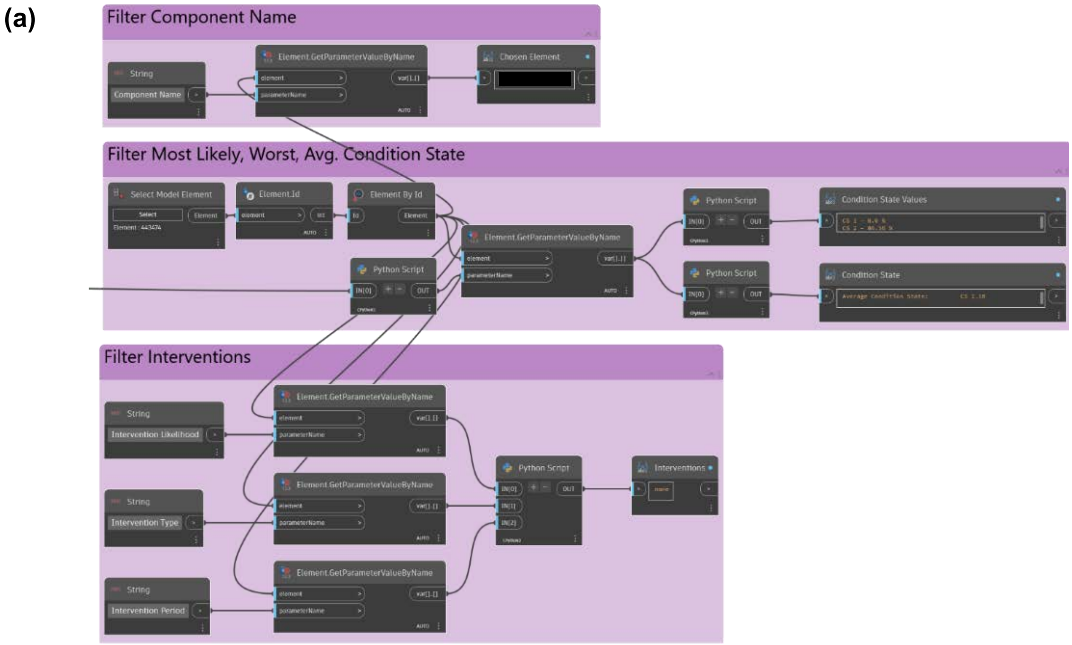


```
for mix_name, int_type, int_period, cost, pos_time in zip(excelList[mix_name_index], excelList[int_type_index], excelList[int_period_index], excelList[cost_index], excelList[pos_time_index]):
    intervention = {
        "MixName": mix_name,
        "IntType": int_type,
        "IntPeriod": int_period,
        "Cost": cost,
        "PosTime": pos_time
    }
    interventions.append(intervention)


elementDict = {}

for key, value in zip(unwrapped_elements, componentNames):
    elementDict[key] = value


for element,key in elementDict.items():
    for i in interventions:
        if i["MixName"] == key:

            type_parameter = element.LookupParameter("Intervention Type")
            type_parameter.Set(i["IntType"])

            period_parameter = element.LookupParameter("Intervention Period")
            period_parameter.Set(i["IntPeriod"])
```

Figure 49: Snippet of the Python node used to set the intervention values.

Visualize element condition states and interventions

Detailed information about a specific selected element has been added to the workflow as an additional feature to the basic visualization using color coding inside the model. It is illustrated by the set of nodes depicted in Figure 50. The 'Select Model Element' node serves as a basis for the other following nodes. It is also specified as an input node into the Dynamo Player to provide flexibility in selecting the component from which more detailed information is required.



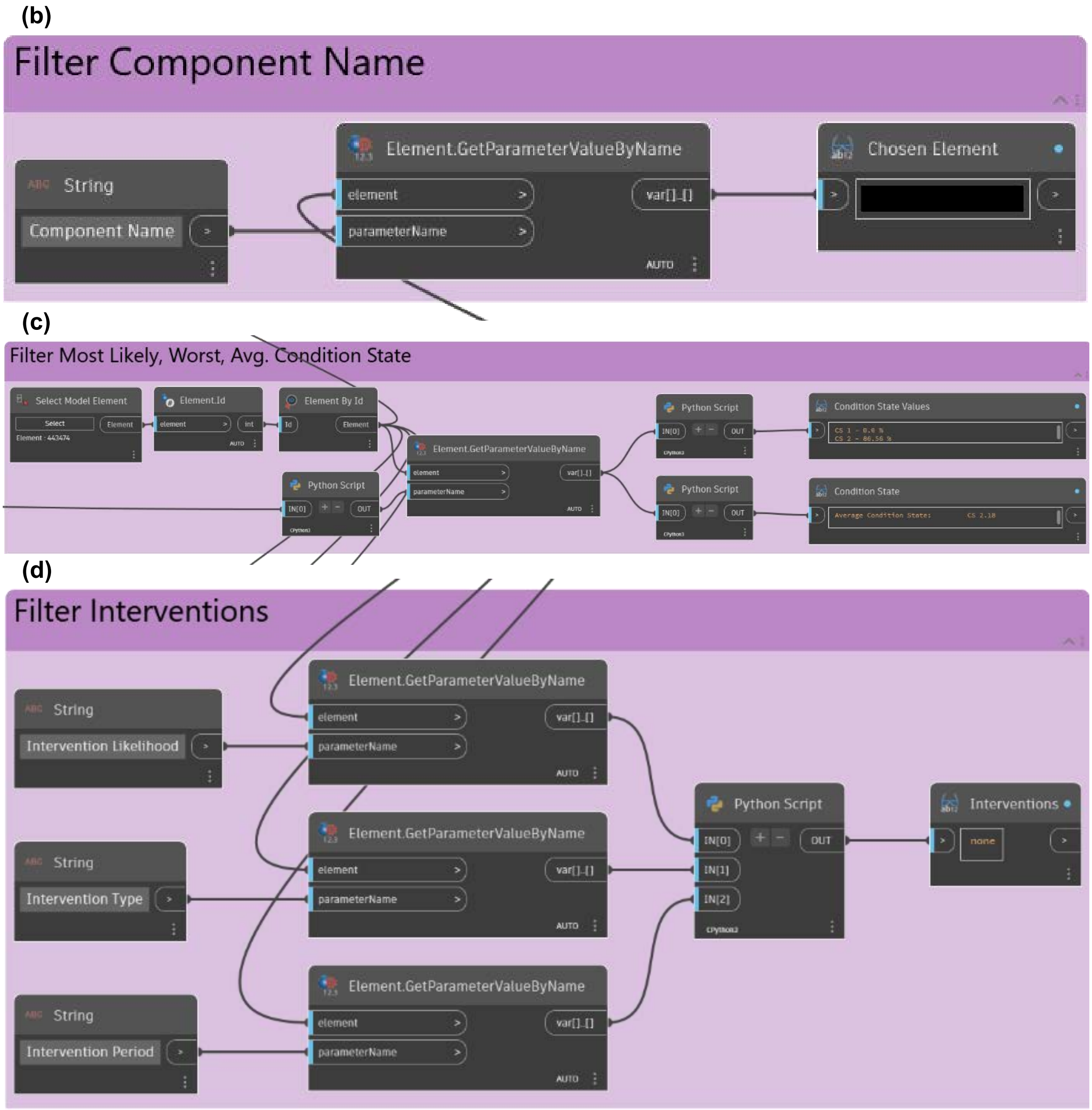


Figure 50: Nodes used to integrate a selection of elements in the Dynamo Player, returning values on the component name, condition states, and interventions. To enhance visualization, the node (a) is magnified at three distinct sections: 'Filter Component Name' (b), 'Filter Most Likely, Worst, Avg. Condition State' (c), and 'Filter Interventions' (d).

The ‘Component Name’ parameter (see Figure 50b) is extracted from the selected element in order to identify the current selection associated with the information shown. Figure 50c depicts the parameters for each condition state. Additionally, the ‘Average Condition State’, ‘Worst Condition State’, and ‘Most Likely Condition State’ for the selected element are calculated based on the likelihood values for each state, which are then output as numerical values. The average condition state is represented as a floating-point number, declaring the expected average number of all condition states based on their likelihood in the specific year. The worst condition state outputs the value of the first non-zero condition state starting from the lower end. In the case of the most likely condition state, the greatest likelihood is printed out. The string information has been modified inside the Python nodes for better readability.

The intervention information is provided in Figure 50d. The nodes extract the values from the given parameters concerning planned interventions in the form of ‘Intervention Likelihood’, ‘Intervention Type’, and ‘Intervention Period’. In the case of no interventions planned in the included planning periods, ‘none’ will be output in the view. A string modification has also been applied here to provide a more engaging visibility. All the mentioned output values have been included in the Dynamo Player as part of the workflow.

Visualize model condition state and interventions transparency

Unlike the previous set of nodes (see Figure 50), which provide more detailed information for specific asset components in the models, a general wide-scaled filtering of the values has been added to the Dynamo script, as depicted in Figure 51. These nodes are presented as an aid to obtain a better overview of the components by applying transparency filter parameters to all the elements. This part of the script is also separated into parts treating condition states and interventions separately.

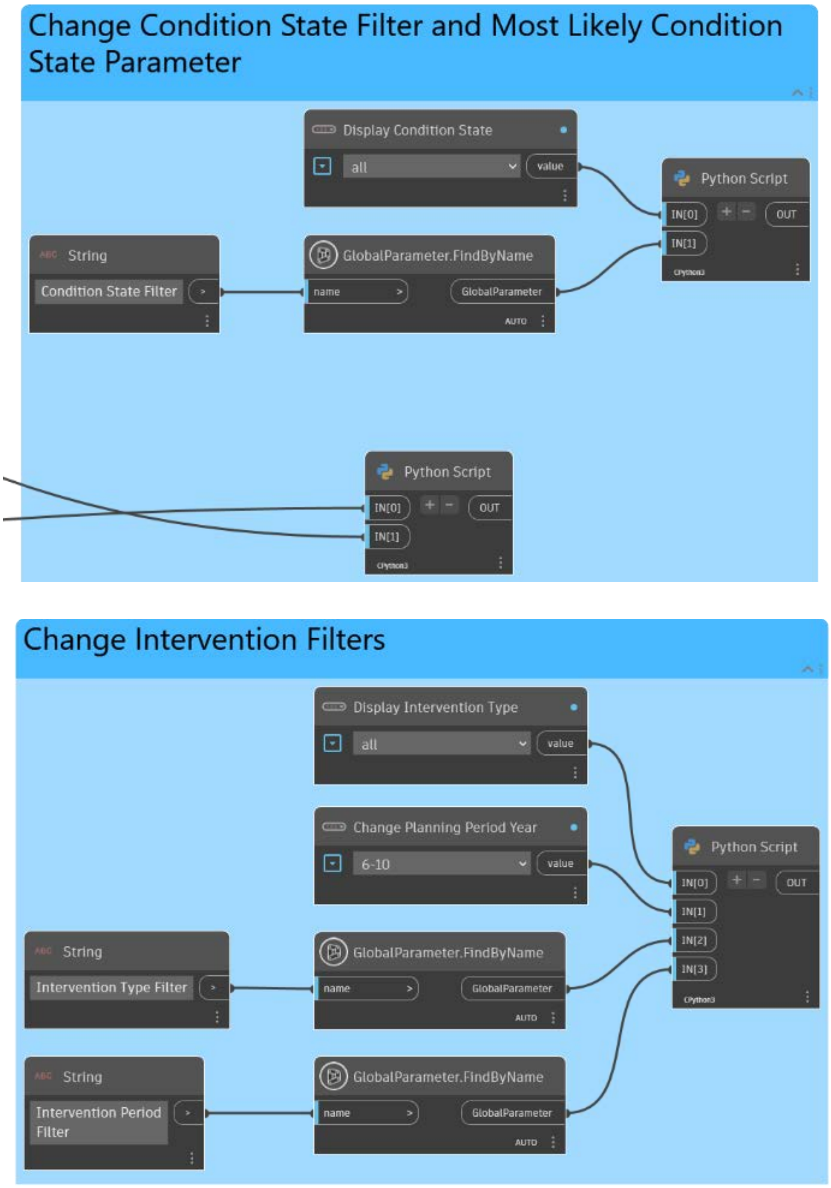


Figure 51: Nodes used for changing the global filter parameters in the Dynamo Player.

Generally, the desired values for a condition state are used as an input, where the selection between the condition states 1 to 5 is possible. Several threshold values have also been included. Therefore, filtering for components with ‘CS3 and worse’, for instance, is also feasible. These values will then be assigned to a global parameter with the same name, through which the view filter will be applied later on. In the lower Python node, the ‘Most Likely Condition State’ for all elements is calculated and assigned to the components. This value will then be used as a comparison inside the filters.

In the set of nodes handling the interventions, values for the filters regarding 'Intervention Type' and 'Planning Period' can serve as input, which will be assigned similarly to the explanation in the paragraph above. The data for the filters are additionally provided as an interactive input in the Dynamo Player.

Coloring between condition states and interventions

In order to use the same nodes for both visualizing condition states and interventions, an additional global parameter called 'Color override' has been created. In this way, separate scripts for both use cases have been developed, with a slight change in the 'Color override' parameter and the interactive input values for the element and model scale (see Figure 52). For the script regarding the condition states of the components, the approach includes setting the values depending on the Excel sheet and resetting the values for the interventions. This way, a coloring is provided only for one use case at a time. The same method is also applied when using the script to visualize interventions.

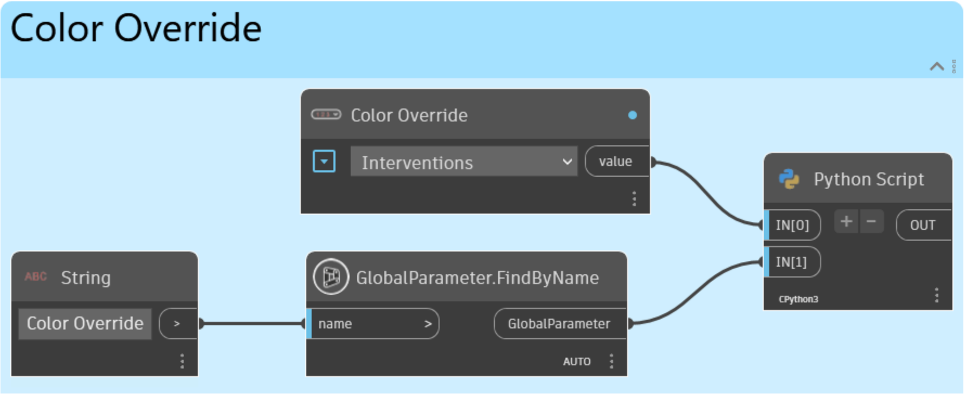
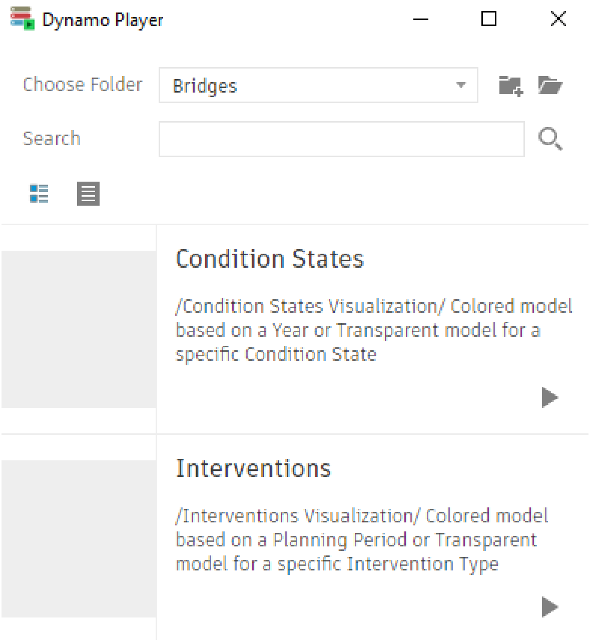


Figure 52: Nodes utilized to change the color override of the model elements.

2.2.4 Visualization

In addition to the dynamo visualization script presented in Section 2.2, the interactive user interface of the Dynamo Player in Revit has also been integrated into the workflow. This graphic interface is used to provide more detailed information, apart from the color overrides which can be seen in the 3D model, as well as a more intuitive way of handling the model and its parameters by introducing dropdown menus and action buttons. It supplies the user with input options, such as changing the evaluated year. Furthermore, it is separated into acquiring information at the component level, giving data about the condition states and interventions of one selected component, as well as filtering out all components that are in one named condition state or fulfil an intervention condition, on a more global scale.

Another feature of the Dynamo Player is the ability to create multiple Dynamo scripts and have them appear in one common place. This is accomplished by placing all the scripts in one directory and then selecting the said directory as a reference. A folder structure inside the mentioned location in the computer can also be created, providing a categorization of the scripts. As illustrated in Figure 53, a switch between the script for visualizing the condition states and the one for interventions is feasible, opening the possibility for the user to not even need to see the script in order to execute it. By using this folder structure, the number of Dynamo scripts can be extended to an unidentified number of scripts, making room for the scalability of the proposed methodology.



Choose Folder: Bridges

Search:

Condition States

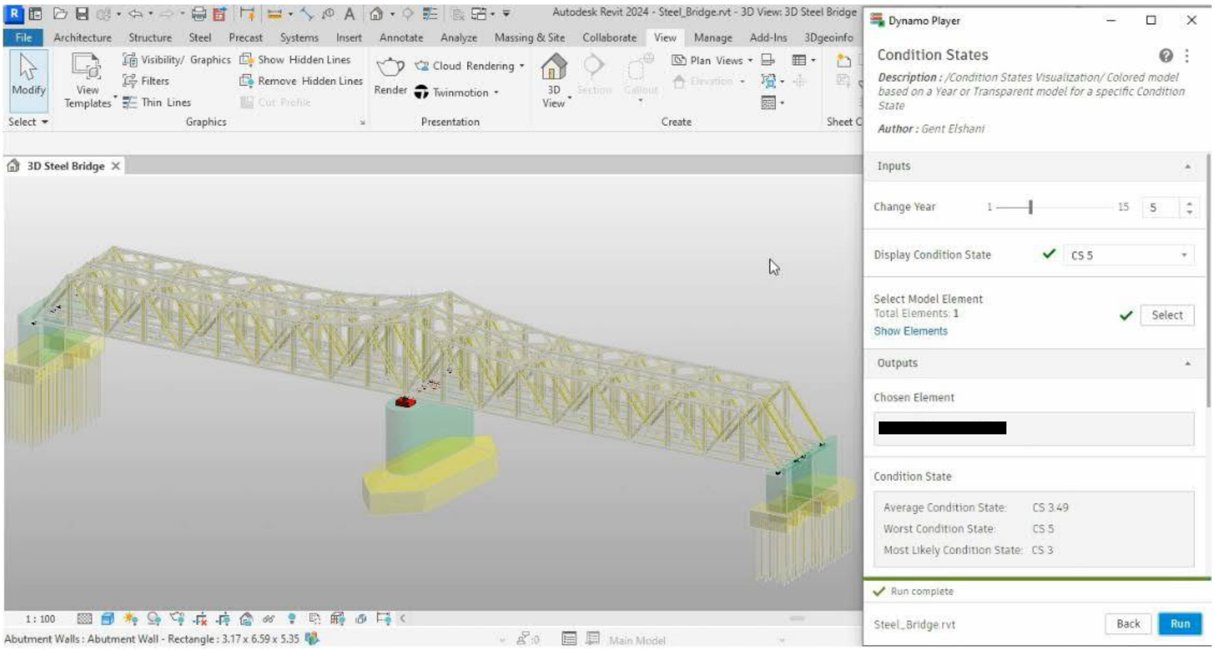
/Condition States Visualization/ Colored model based on a Year or Transparent model for a specific Condition State

Interventions

/Interventions Visualization/ Colored model based on a Planning Period or Transparent model for a specific Intervention Type

Figure 53: Two scripts in the Dynamo Player where the active script can be selected.

Finally, a visualization (see Figure 54) can result using the described workflow. In this screenshot, by selecting the global transparency for the CS5, only elements in the said condition state will be shown. This grants the ability to see elements even in a complex model where a lot of small elements are present. In addition, information about the selected element can be obtained from the output of the Dynamo Player, and insight about the condition states can be created utilizing the colored model as a reference.



Abutment Walls : Abutment Wall - Rectangle : 3.17 x 6.59 x 5.35

Condition States

Description : /Condition States Visualization/ Colored model based on a Year or Transparent model for a specific Condition State

Author : Gent Elshani

Inputs

Change Year: 1 – 15, 5

Display Condition State ✓ CS 5

Select Model Element
Total Elements: 1 ✓ Select
Show Elements

Outputs

Chosen Element
[redacted]

Condition State

Average Condition State: CS 3.49
Worst Condition State: CS 5
Most Likely Condition State: CS 3

✓ Run complete

Steel_Bridge.rvt Back Run

Figure 54: The visualization through transparency, color coding, and additional numerical information on a 3D model.

2.3 Proposed workflow

As part of a more extensive process known as the intervention planning process, the proposed methodology operates within the asset level of this process [2]. The focus is specifically on bridge asset management and track asset management because the condition of the tracks passing on top of the bridges makes the assets of both domains inevitably connected. By introducing automation in digital asset generation and providing an easier way to visualize the said data, decision-making can be considerably helped. In Figure 55, the initial part of the bridge, respectively the track asset management, is shown as part of the intervention planning process. The aim is to facilitate the management of these assets with regard to their predicted future condition states, deterioration over time, and the predicted future interventions resulting from the specific condition state. This proposed methodology intends to get a better overview of the asset information that was previously captured in tables and databases.

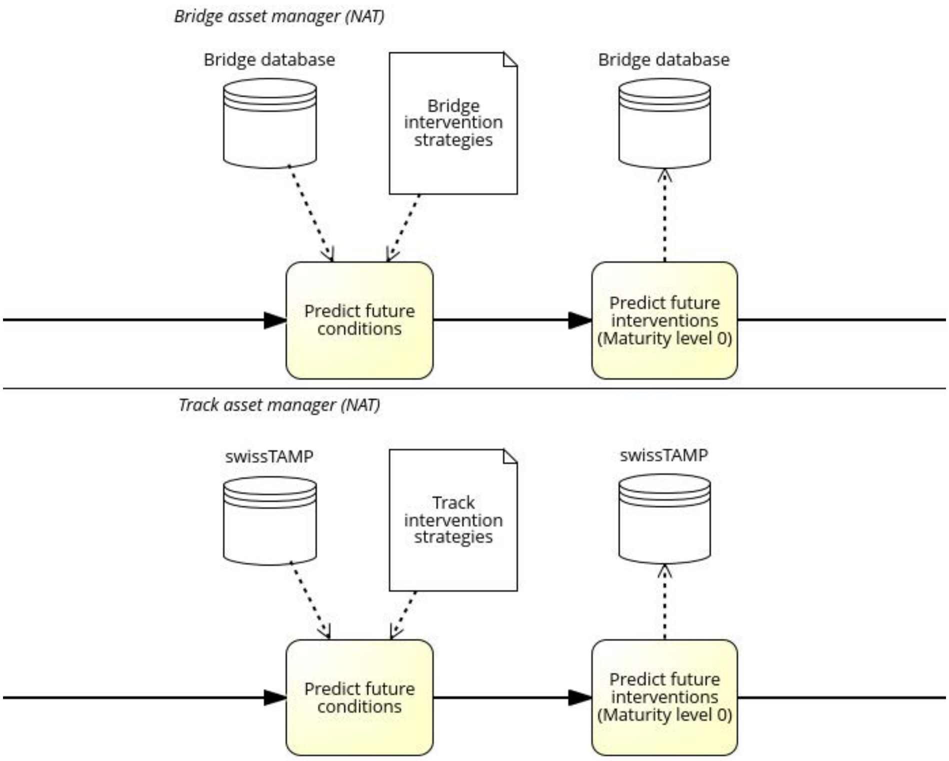

Bridge asset manager (NAT)
Bridge database
Bridge intervention strategies
Bridge database
Predict future conditions
Predict future interventions (Maturity level 0)
Track asset manager (NAT)
swissTAMP
Track intervention strategies
swissTAMP
Predict future conditions
Predict future interventions (Maturity level 0)

Figure 55: First processes of the bridge and track asset managers as part of the intervention planning process [2].

The goal of executing interventions precisely within allocated time slots is advanced by leveraging the advantages of employing a 3D parametric BIM model. It is worth mentioning, however, that this methodology does not aspire to fully substitute data captured in tables and reports. It rather seeks to provide a general overview of the asset condition and enhance the decision-making process for the necessary interventions on the named assets.

A continuous update and enrichment of the digital models is also foreseen, mirroring the advancements and modifications made to the planned interventions for separate assets. Figure 56 illustrates the processes succeeding the bridge and track asset management, where requirements of a higher maturity level are identified. By using the proposed methodology, the fulfilment of these processes can be facilitated, providing additional aid to the coordination with the different asset managers.

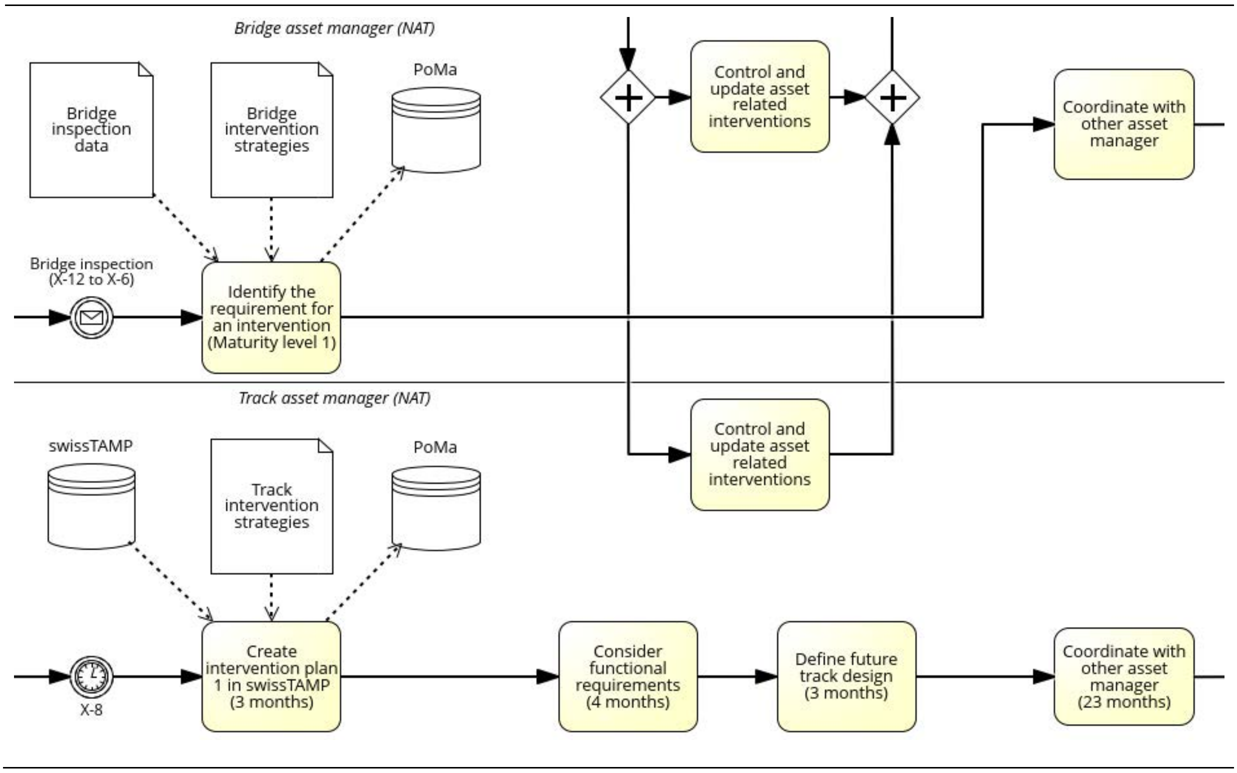

Bridge asset manager (NAT)
Bridge inspection data
Bridge intervention strategies
PoMa
Bridge inspection (X-12 to X-6)
Identify the requirement for an intervention (Maturity level 1)
Control and update asset related interventions
Coordinate with other asset manager
Track asset manager (NAT)
swissTAMP
Track intervention strategies
PoMa
Control and update asset related interventions
X-8
Create intervention plan 1 in swissTAMP (3 months)
Consider functional requirements (4 months)
Define future track design (3 months)
Coordinate with other asset manager (23 months)

Figure 56: Processes involving higher maturity and coordination between asset managers as part of the intervention planning process [2].

By improving the data quality over time and updating the BIM model according to the planned interventions, a single source of truth can emerge from these processes. This source lays the foundations of a better management of the assets over time.

By using the created 3D BIM models as a basis, bridge and track asset managers can finally collaborate more effectively. These models can serve as a common visual platform for joint assessments, planning, and decision-making. With data integrated into these models, both asset managers can analyze the interdependencies between bridges and tracks, allowing for more informed, coordinated decisions on maintenance, repairs, and optimizations. This shared digital environment enhances communication between teams while maximizing the overall efficiency, safety, and performance of the rail network.

3 Network view

As a crucial part of broader scale coordination, especially among the different managers participating in the intervention planning process, a common display of both bridge and track components in a unified view is of utmost importance. This issue has been addressed as part of this Thesis, trying to advance the previously mentioned asset level to a more comprehensive scale, with the objective of improving collaboration between different participants in the intervention planning process, each managing their own assets.

For this reason, in this section, the so-called network view is introduced to approach the mentioned restrictions and advance the decision-making process for interventions on a larger scale. The proposed network view is described as a system of assets containing bridges and tracks. Its usefulness manifests through applying color coding and transparency to the components depending on different conditions, ensuring a more straightforward overview of the assets and their synergies in applying interventions to them. Furthermore, the clustering of interventions can be facilitated by observing the components that have common features regarding interventions. This is explained in more detail in the following parts.

One principal benefit of this view is that it should be regarded as an additional resource to the more detailed data and it is not intended to entirely substitute them. Instead, it provides further visual aid to better plan the interventions performed on the assets. The chosen visualization is similar to the one presented in the asset level using the filter overrides in Revit. Since the envisioned usage time of this proposed data model spans several years before the actual interventions are scheduled, refinement and enhancement with newer data over time have to be considered as well. In this way, a single source of truth can be achieved, promoting collaboration from a unique origin as a basis, instead of communication based on different sources between the stakeholders.

Figure 57 suggests what the result of this approach for the network view could look like, combining the assets of the bridges and railway tracks into one view. Further visualization is described in more detail in Section 3.3.

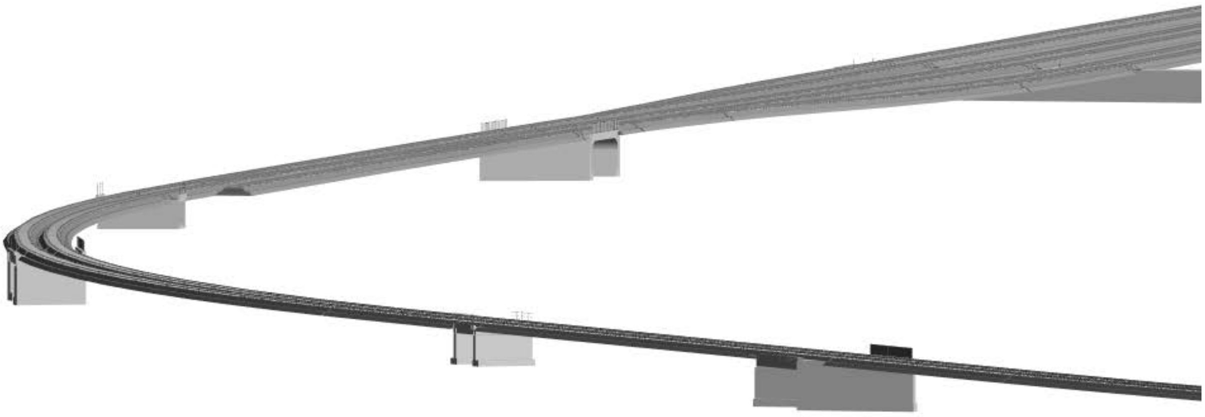


Figure 57: Visualization of the network view containing bridge and track assets.

3.1 Data exchange

3.1.1 File formats

Efficient data exchange and interoperability are indispensable to achieve effective railway asset intervention planning. Railway systems usually rely on various software tools and data formats to manage maintenance schedules, track asset performance, and plan interventions. These systems often come from different vendors and may use proprietary file formats or data structures.

Interoperability ensures that these different software tools can seamlessly exchange data without manual conversion or data loss. This saves time and reduces the risk of errors associated with manual data entry or format conversions. The aim of ensuring easy access and integration of historical data and records into the planning process, while also providing flexibility to harness opportunities in fields such as data analytics, predictive maintenance, and asset management, can be achieved through this approach.

To provide an efficient data exchange between the stakeholders, it is crucial to understand and utilize different data structures of the source and target files. Analyzing the structure, format, and semantics of the data in each system, including documentation and collaboration with stakeholders, is essential in gaining insights into these schemas. Establishing a mapping between these schemas entails the formulation of a set of rules and instructions that determine how data elements from the source schema should be mapped to their corresponding elements in the target schema. Depending on the file formats, finding a good match can be challenging for several elements.

Especially when handling geometric data paired with CAD design, proprietary formats such as DXF, OBJ, and STL are used. In the aspect of integrating the processes into BIM, IFC is commonly used. It employs an inheritance data model with classes being structured in a hierarchic way, as shown in Figure 58, where only a part of the whole schema is depicted.

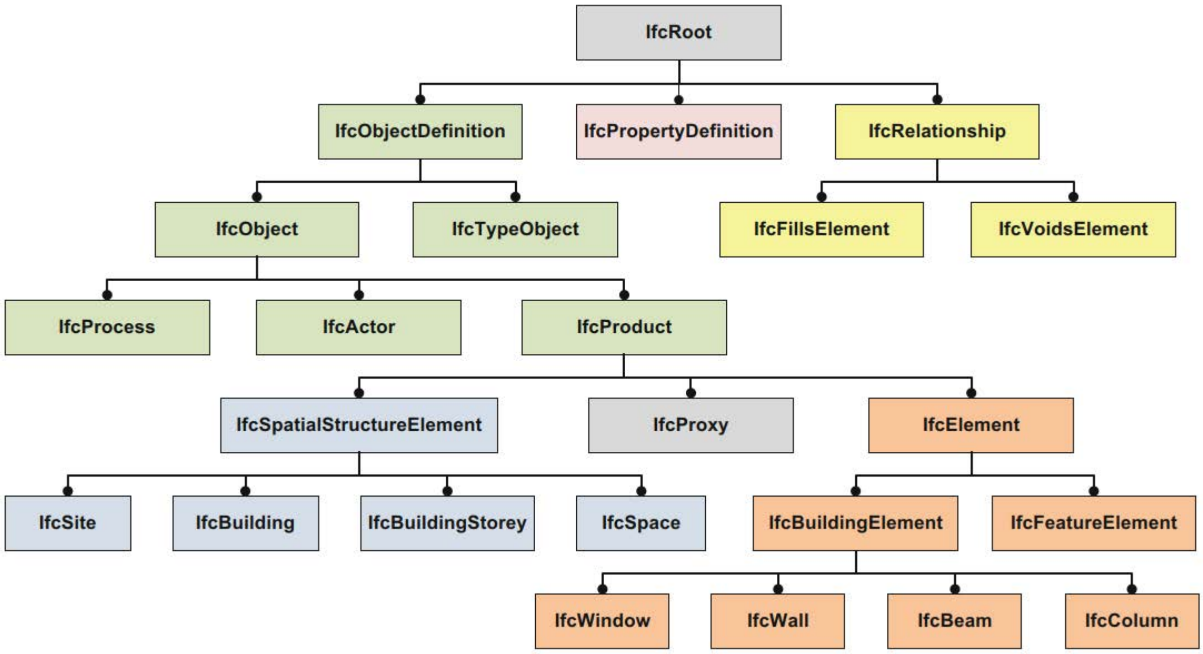


Figure 58: Part of the data structure from the IFC file format [50].

In comparison to the complex data structure of IFC, simpler file formats can be utilized based on the covered data and the expected outcome. In this way, when dealing with computer-aided design without the inclusion of semantics and parametric data, DWG might be a better choice. On the other hand, when only listing the data in a table format, an even simpler approach using comma separated values (CSV) can be employed. This is because the data is stored as plain text, significantly facilitating the data exchange process.

3.1.2 Geodata vs building data

The difficulties when exchanging data between different domains are also encountered when trying to convert between the larger-scaled geodata and the finer-grained building data. Some of the most used file formats are shown in Table 6, making a clear distinction between the file formats used in the respective sectors.

Table 6: File formats commonly used to represent geodata and building data.

Geodata	Building data
Shapefile	IFC
GeoJSON	DWG
KPX	DXF
GPX	BCF
CityGML	FBX

While geodata extend over a larger area, hence conditioning the generation of big data, building data is far smaller in the aspect of data size. This makes the approach of handling both types of data more challenging. Moreover, conversions and data losses must be taken into consideration. Especially when using a BIM authoring tool, such as Revit, larger scales are not conditionally taken into account because of how Revit approaches the modeling space by placing a 2D working plane over the project extent. Small distortions in the model can also be expected since the projection from the earth's curved surface into the planar working space is usually calculated in GIS software, which is not the case in other BIM software [51].

3.1.3 Network view approach

The aim of utilizing a network of bridge and track assets into a single view to gain a general overview of the interventions planned in a specific railway section requires the integration of both domains into a single interface. Revit has been selected as the Common Data Environment (CDE) where the network will be located for several reasons. Firstly, one factor is the much more efficient way in which the dynamic parameter assignment and modification, as well as the visualization are performed compared to other BIM authoring software. Furthermore, the visualization scheme utilized at the asset level and explained in Section 2 can be reused. This means that a new visualization does not need to be learned by the user, and considerable parts of the code and scripts can be reused with only a few slight adjustments.

The specific assets included in this network view are assets, at the component level, of bridges and railway tracks. These tracks contain five components: rails, ballast, subgrade, sleepers, and fastening system. Since some of the bridge models were already existing 3D models, the approach was to create the remaining bridges in Revit as separate models and generate the track assets using their alignment as a reference.

Figure 59 presents the different input sources for the railway tracks. The extent of example line has been extracted from the SBB online database [52]. Since the data was less detailed and consisted of only one line representing the alignment with no track number information, this source was considered to set the extent of the line, where the bridges and tracks have been evaluated only inside this area. More reliable data have been extracted from Swisstopo where the whole railway network of Switzerland is available [53]. Since the track alignments contained each separate track as a distinct line and multiple lines in the case of sidelines and train stations, this dataset has been utilized to form the basis of the track asset generation.

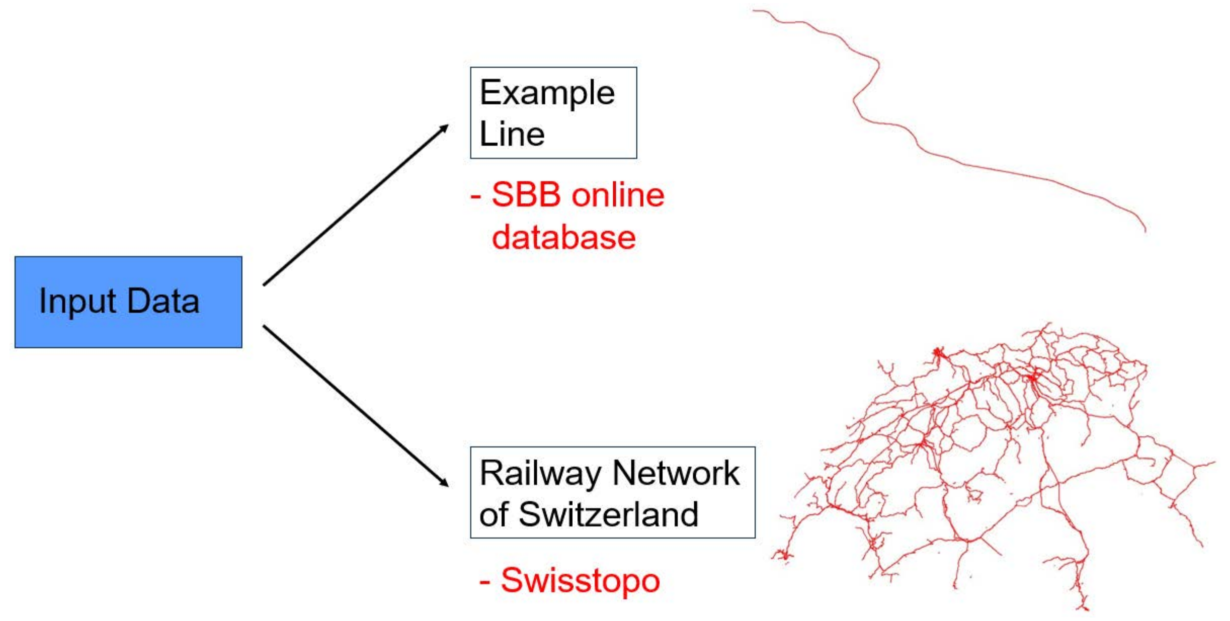


Figure 59: Input data of railway track alignments of the example line and the railway network of Switzerland.

While the family catalog introduced in Section 2.1 enables the creation of the remaining bridge models, the challenge remains in exploring possibilities to include the provided track data into Revit. Since the import options of Revit are rather limited (see Figure 60) and accept only a small number of file formats (i.e., IFC, DWG, and PDF), the main challenge was to transform the already existing track alignments into a fitting format which is usable in the aspects needed to remodel the tracks again in the BIM authoring tool. This transfer needs to be carried out with as little data loss as possible, as well as in an efficient manner, with not too many steps and conversions.

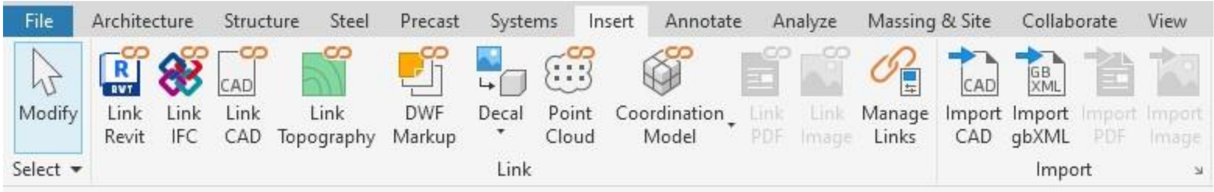


Figure 60: File insert options in Revit.

The file formats in which the input data for the railway tracks was available were limited to the below-presented formats:

Example Line:

- Shapefile
- GeoJSON
- KML
- GPX

Railway Network of Switzerland:

- Shapefile

The ESRI shapefile has presented itself as more reliable in the aspect of representing geodata because of the simplicity of it. Features such as annotation, attribute relationships, topology relationships, attribute domains, and subtypes cannot be represented in shapefiles. However, they are not needed for the specific treated use case, making the usage of this file format more of an innate choice [54]. In addition, the fact that the railway network of Switzerland was available only in this format, reinforced this choice.

Many routes have been explored of how to transfer the given data to a usable format for this Thesis, always with a special focus on the interoperability in the aspects of BIM [34]. Figure 61 shows the concepts of BIM regarding the usage of software tools, open formats, and different disciplines. The terms little, big, closed, and open BIM are introduced, yielding a given number of combinations between these expressions.

A distinction between closed BIM and open BIM exists concerning data exchange. The concept of closed BIM prioritizes vendor internal data transfer, whereas integration from third-party software suppliers is done in a very limited matter. This way of sharing information can lead to an optimized workflow inside vendor-specific software. It faces, however, a lot of limitations when transferring between different software providers. Meanwhile, open BIM gives greater importance to open formats, which can be exchanged with little to no issues between different software. The drawback of these types of data representation lies in the limited possibility of including many specific features due to the global availability that must be provided.

The overall aim of the industry is to remodel its processes to achieve big open BIM, an approach where data exchange is carried out using open formats, reducing the importance of distinct software vendors since the data will be indistinguishably interpreted over all platforms [34]. Moreover, the same digital assets can be used across all stages and different disciplines when implementing this aspect, supplying the processes with coverage regarding the data format over the whole life cycle of the project.

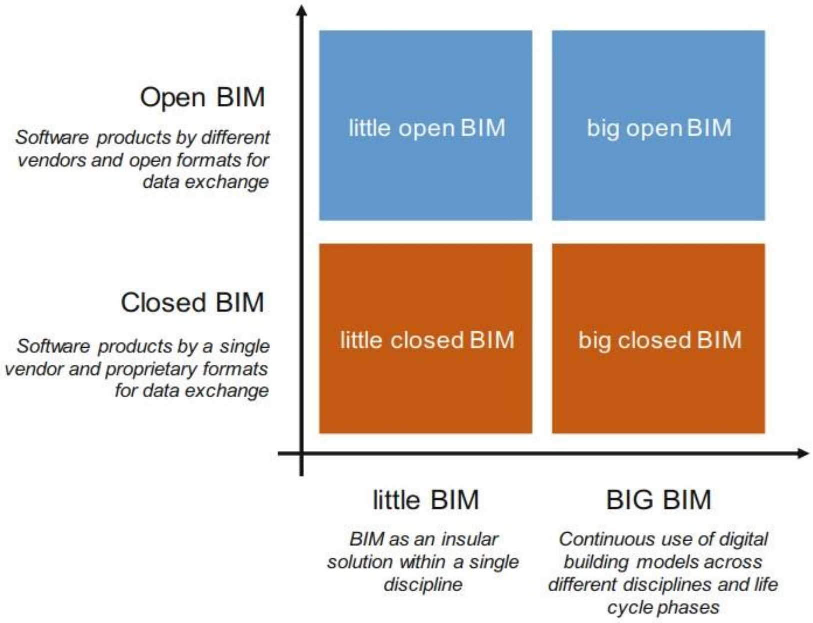
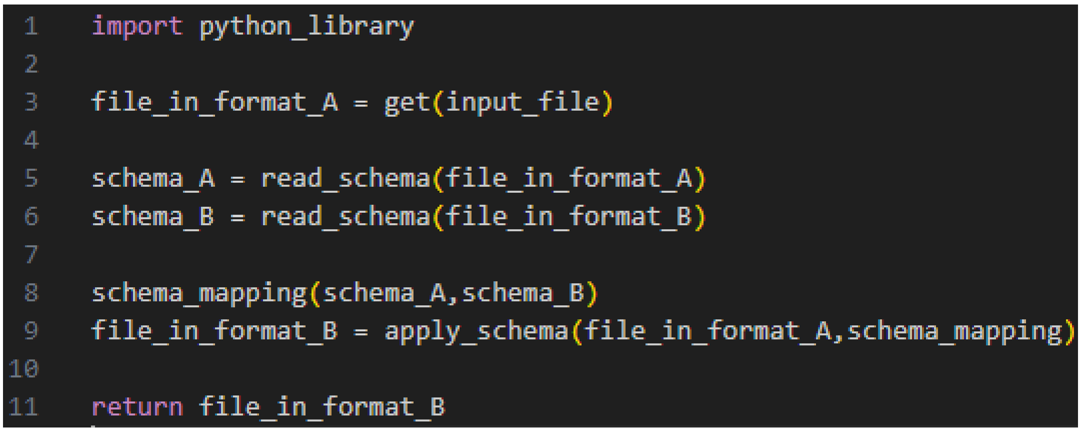

Open BIM
Software products by different vendors and open formats for data exchange
little open BIM
big open BIM
Closed BIM
Software products by a single vendor and proprietary formats for data exchange
little closed BIM
big closed BIM
little BIM
BIM as an insular solution within a single discipline
BIG BIM
Continuous use of digital building models across different disciplines and life cycle phases

Figure 61: The concepts of ‘open BIM’, ‘closed BIM’, ‘little BIM’, and ‘big BIM’ [34].

To elaborate on the selected data transfer methods in this methodology, the following section separates into three main approaches for data exchange. The aim of converting the given data from different sources of the railway track alignment into a format compatible with the tools offered by this methodology is explored. Conversion between different file formats has been done with the ultimate goal of making use of the existing data. Moreover, the identified limitations and potential future improvements are described in this section.

Code routine

The approach of developing and utilizing a code routine to provide an exchange between different file formats has proven the most versatile way compared to other techniques. There are several different ways in which data conversion can be done using this approach. The most elementary path is to write a script that will perform the file format adjustments and results in a new format. Figure 62 illustrates a pseudo-code of how such a conversion could look like using already existing Python libraries. By reading the information of how the data schemas are built and inducing a mapping between these data structures, the resulting file can be extracted in a new file format.



```
import python_library

file_in_format_A = get(input_file)

schema_A = read_schema(file_in_format_A)
schema_B = read_schema(file_in_format_B)

schema_mapping(schema_A,schema_B)
file_in_format_B = apply_schema(file_in_format_A,schema_mapping)

return file_in_format_B
```

Figure 62: Pseudo-code for file conversion using Python.

Usually, the procedure of directly developing a script to perform the conversion requires a deeper knowledge and insight into the data schemas of the distinct file formats. Consequently, this approach is generally not a natural choice.

Another way of utilizing the flexibility of writing a code directly is by using already developed scripts for particular use cases. In this way, several scripts and frameworks have been built to address the conversion between building data and geodata, while focusing more on the larger scale data. Some of the most relevant tools that facilitate data conversion are captured in Table 7.

Table 7: Selected tools to handle data conversion and management.

Tool	Main developer
3DCityDB	Chair of Geoinformatics (TUM)
Citygml-tools	Chair of Geoinformatics (TUM)
IFCCityJSON	Open source
IFCOpenShell	Open source

The main drawback of these tools is their focus on very specific data formats and the requirement for a deeper understanding of databases and web services. For instance, the 3DCityDB provides an online database by utilizing PostgreSQL or Oracle's spatial database, both of which request a detailed setup [55].

Recent advancements with the aim to provide the same functionalities of the proposed conversions by executing a code routine, by using a more user-friendly interface have been made. Worth mentioning are the software Speckle and FME. While the focus on promoting open BIM is mentioned, the latter does not provide open access. Nonetheless, the usage of these tools can advance productivity by providing seamless data conversion.

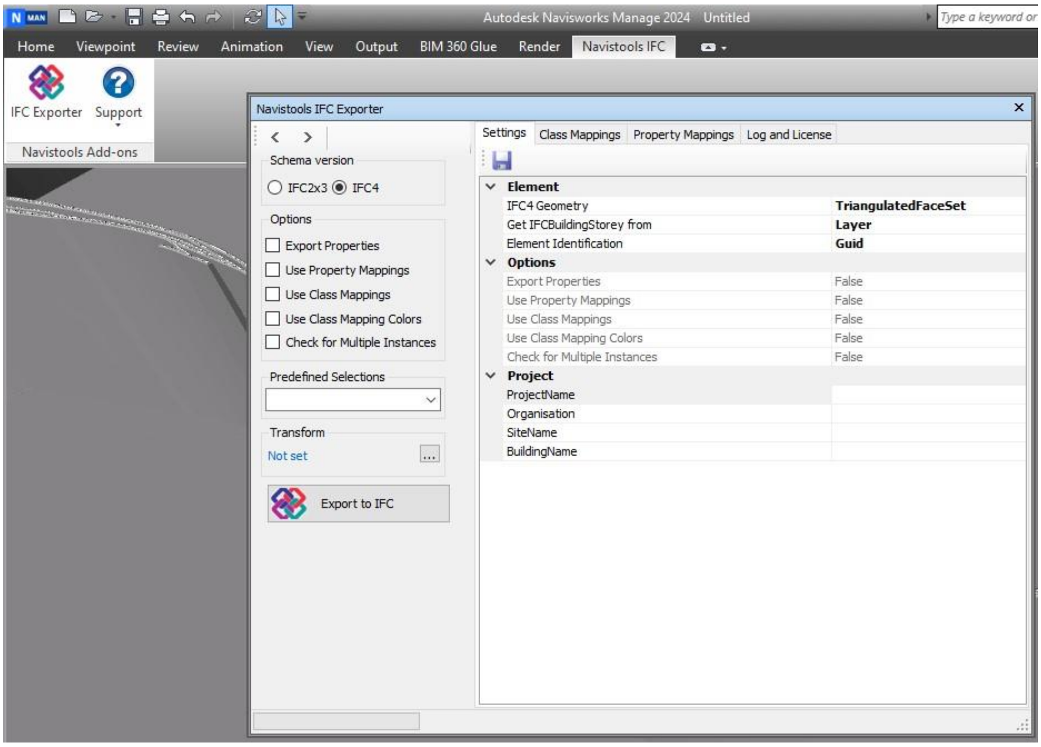
Because of the versatile configuration of these methods for data conversion, open-source formats are usually preferred, enhancing the concepts of open BIM. While using this type of file exchange, a shift towards non-proprietary communication between stakeholders can be implemented.

Software plugins/Online converters

Following this data conversion approach, a variation of both open and closed BIM can be achieved depending on what the handled data represents. Several software plugins have been explored throughout the course of this Thesis. Increased emphasis has been placed on the two following add-ons:

- Navistools IFC Exporter
- Revit CityJSON Importer

The Navistools IFC Exporter is a software plugin that makes the export from Autodesk Navisworks to IFC possible (see Figure 63). It is developed by an application provider which makes the accessibility more challenging. The export is feasible in the formats IFC2x3 and IFC4. Two geometry representations are also available. The converted geometry can be expressed as a triangulated face set and a faceted boundary representation, both providing implicitly created geometry. The two possibilities of exporting an IFC file are explored in more detail in the following parts.



Schema version

○ IFC2x3 ● IFC4

Options

□ Export Properties
□ Use Property Mappings
□ Use Class Mappings
□ Use Class Mapping Colors
□ Check for Multiple Instances

Predefined Selections

Transform

Not set

Element	
IFC4 Geometry	**TriangulatedFaceSet**
Get IFCBuildingStorey from	**Layer**
Element Identification	**Guid**
Options	
Export Properties	False
Use Property Mappings	False
Use Class Mappings	False
Use Class Mapping Colors	False
Check for Multiple Instances	False
Project	
ProjectName	
Organisation	
SiteName	
BuildingName	

Figure 63: Navistools IFC Exporter interface in Navisworks.

Unlike the previously shown plugin, the Revit CityJSON Importer supports open formats and their integration into proprietary tools. It was developed by the 3D geoinformation group at Delft University of Technology, which made it publicly accessible. In Figure 64, the included tool inside Revit can be observed, making the import of CityJSON files possible. One of the drawbacks here is that open format CityJSON is not used a lot in real-world projects because it is still under development by the scientific community.

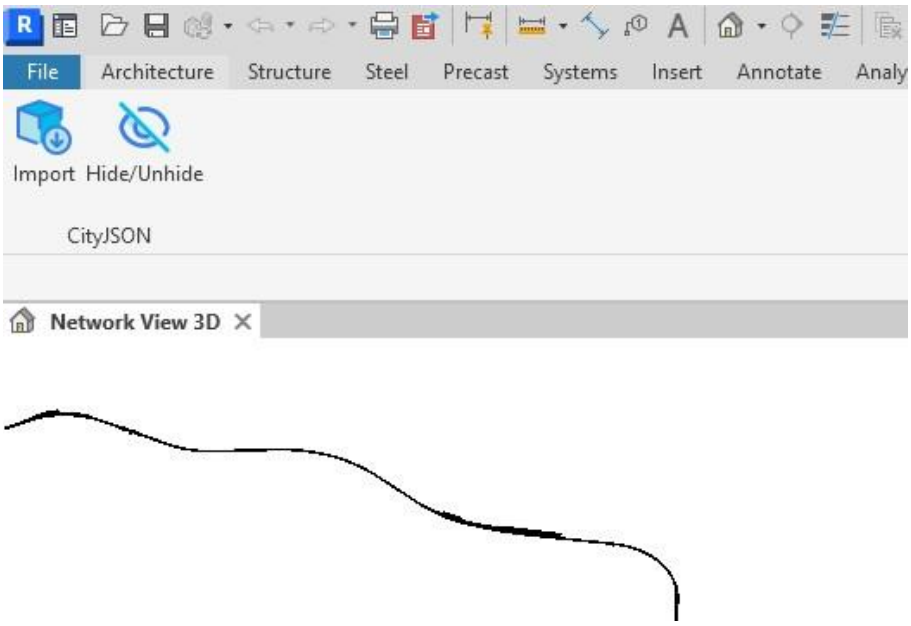


Figure 64: CityJSON Importer in Revit.

The general limitations of using software plugins entail the need for optimizing the workflow between software applications to accommodate the required conversions effectively. Also, the accessibility of these tools might be limited depending on the case. Since they might not be openly accessible, this necessitates the allocation of additional resources in the operations. Furthermore, their value relies on their current state of development at the time of utilization. For instance, the Navistools IFC Exporter toolcan only produce IFC files up to version 4.0.

Another plugin that is worth exploring is an IFC importer into Blender that is provided by the already mentioned IFCOpenShell library. With the integration of this plugin, it becomes feasible to insert and modify IFC files in Blender, which was not possible before. Blender provides a visual approach for editing the IFC file, as opposed to other libraries that follow the route of directly modification through code.

Regarding services proving data conversion through the access of an online service, several alternatives have been investigated. A lot of online converters are available, but the exchange between a lot of file formats that are not used as much does not generate consistent results. Due to the typical lack of access to file schemas and the inability to manage object mapping between instances using these tools, detailed data management becomes unfeasible, thus limiting their utility.

Software internal exchange

This form of exchanging data utilized the converting file formats between software usually provided by the same software vendor. Usually, companies that provide more than one software solution also offer a standard way of exchanging the data between those solutions. By utilizing this link, the data quality will be much better and data loss does not occur that often. As an example, from the software Autodesk Navisworks, the NWD format can be produced, and the data type FBX is recommended to be used from Autodesk Infraworks to replicate the information captured inside the file as correctly as possible (see Figure 65).

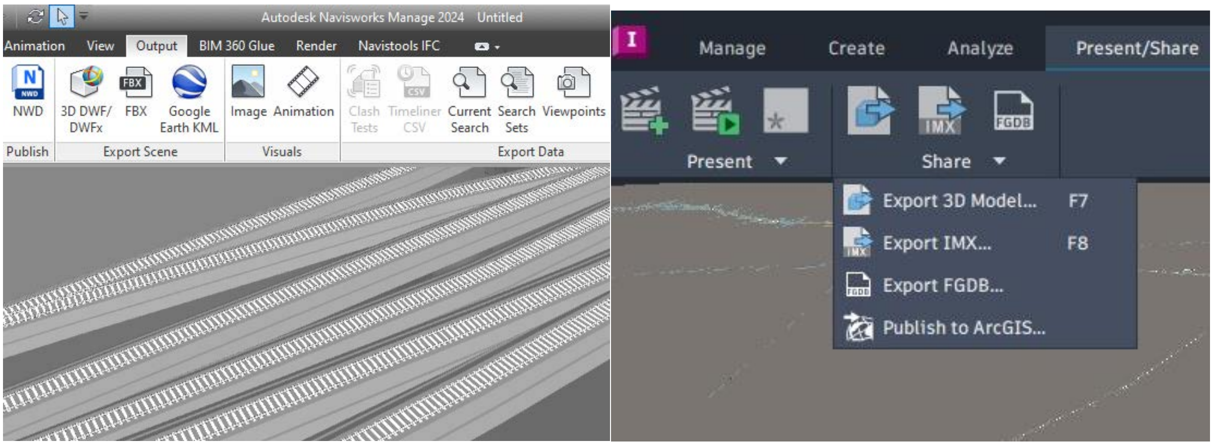


Figure 65: Export options of Navisworks (left) and Infraworks (right).

Although this method is limited in using products from the same vendors, which can lead to difficulties especially when trying to transfer data between different organizations, the most accurate representation can be produced. Limitations to this approach are revealed in constraining the workflows to the tools of a single software vendor, making aim of implementing big open BIM infeasible, since this strategy connects to the concept of closed BIM.

Implemented approaches

Out of the previously mentioned approaches, several conversions from the ESRI Shapefile into a file that can be imported and efficiently used in Revit have been implemented. Following the concepts of efficient data exchange and considering the aspects of BIM, two pathways have been followed in more detail, providing the desired data quality.

Firstly, a conversion from shapefile to IFC has been addressed in greater detail, following the steps outlined in Figure 66. The data goes through several steps of conversion, focusing on the data that is transferred and laying out the exchange in the most optimal way.

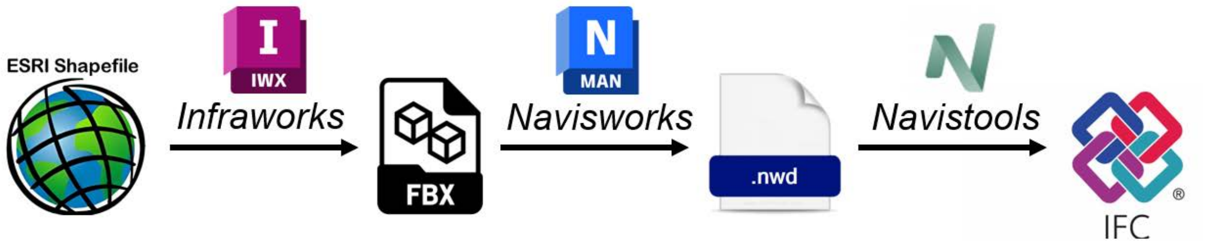


Figure 66: The process utilized for the data conversion from ESRI Shapefile to IFC.

Initially an import of the file into Autodesk Infraworks has been performed. This software is specifically laid out to handle geodata and large-scaled projects in the most optimal way. The data could be directly integrated, providing a visualization of the whole railway tracks of the line, as shown in Figure 67. It is also possible to incorporate the bridge models created in Revit to achieve a network view within Infraworks. However, this approach was discontinued due to its constraints in dynamically assigning and visualizing parameters for the separate track and bridge components and simultaneously having an intuitive user interface that allows input in a more straightforward way. Infraworks does offer the capability to highlight certain components and assign parameter values to them, but it significantly restraints the workflow when compared to how Revit manages these features.

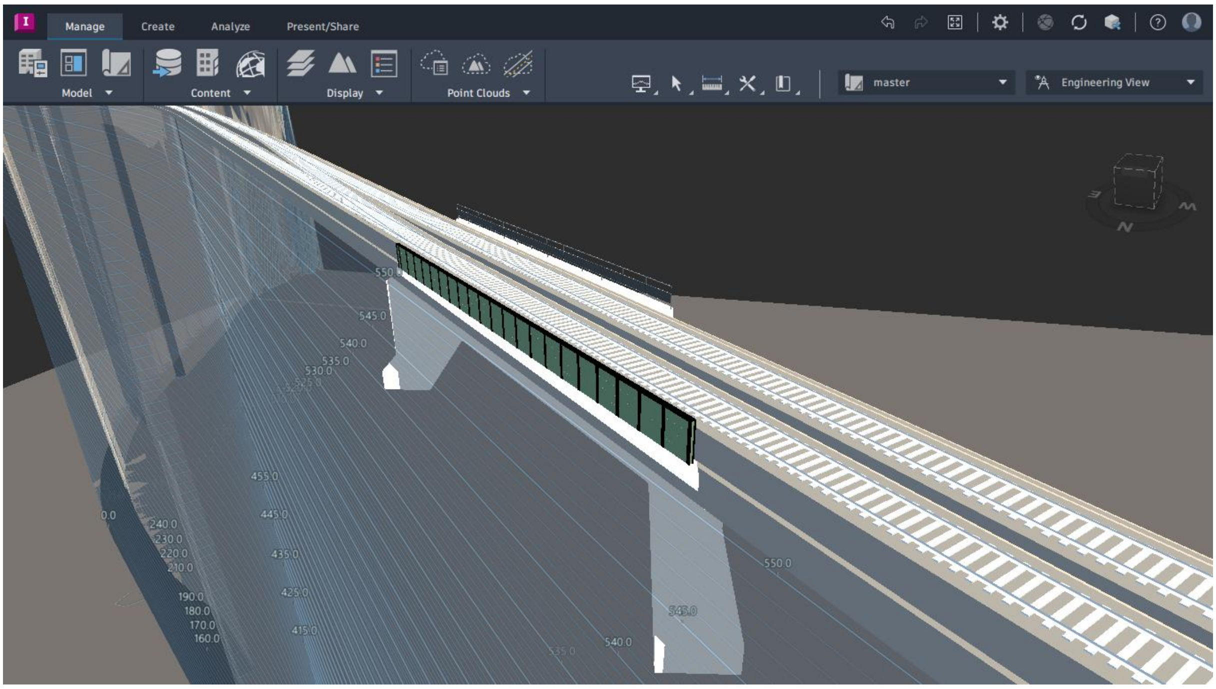


Figure 67: Railway track and bridge assets in Infraworks.

After saving the model of the railway tracks in Infraworks as a FBX file, an import to Autodesk Navisworks has been performed (see Figure 68). Utilizing the aspects of closed BIM, the information and geometric representation was not affected by any form of data loss, making this step of the workflow go through a seamless conversion. Following this step, two approaches of including the file into the Revit user interface have been implemented, including the generation of a so-called Coordination model in Revit and exporting the file into the IFC format, with the goal of including it in the BIM authoring software.

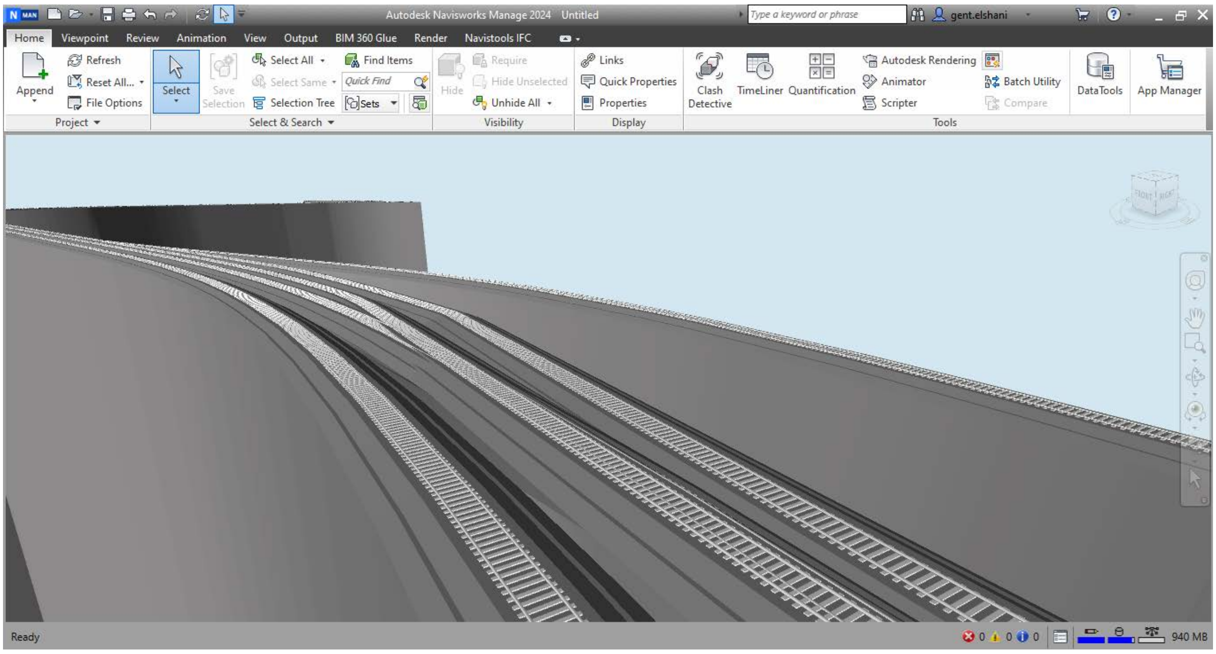
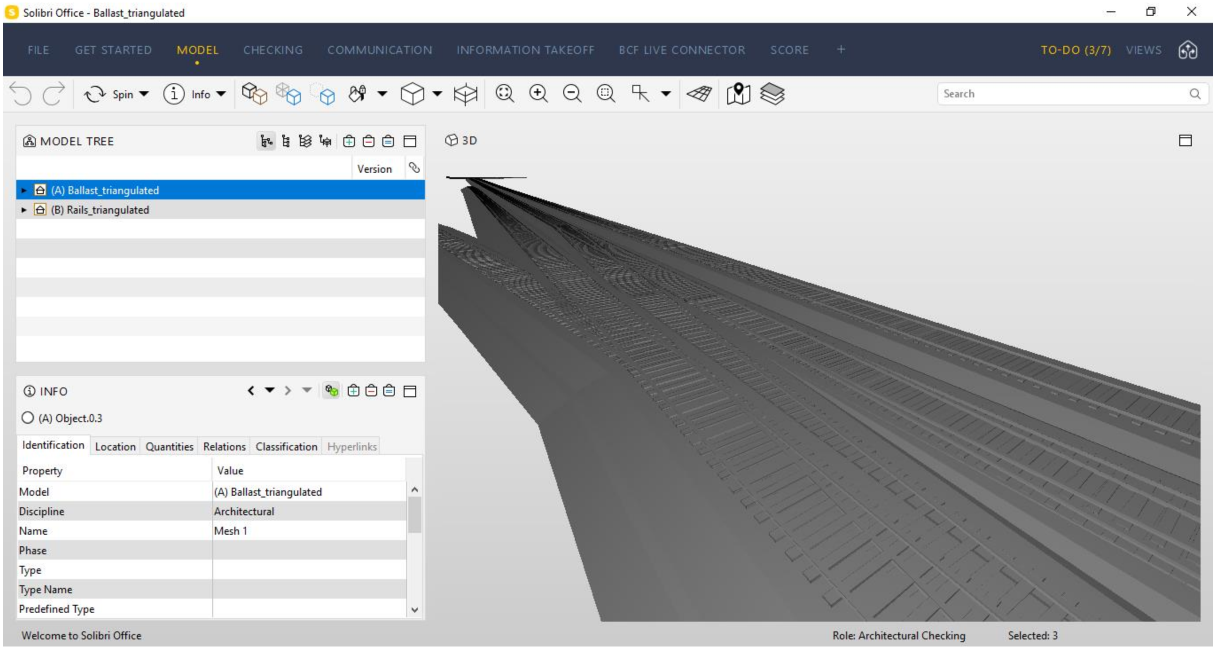


Figure 68: Railway track assets in Navisworks.

The import of the coordination model in Revit is done by using the Navisworks proprietary file format NWD. Although inclusion is completed effortlessly, it presents limitations in the aspect of usability. As the name implies, the main purpose of the coordination model is to provide a basis for project coordination; thus, it is very constrained in providing the features of a conventional 3D model. Consequently, the conversion to the IFC file format has been further explored.

In order to facilitate the exchange of the file containing railway tracks, a plugin has been integrated into Navisworks to enable this functionality. The utilization of the Navistools IFC Exporter enabled exporting an output up to the format IFC4. When exporting the track file using the approach of generating faceted boundary representations, it results in consistent geometries, albeit with a large file size, reaching almost 1 GB for the tracks within the treated line (see Figure 69). Meanwhile, storing the geometry as triangulated face sets significantly decreased the file size into roughly 500 MB.



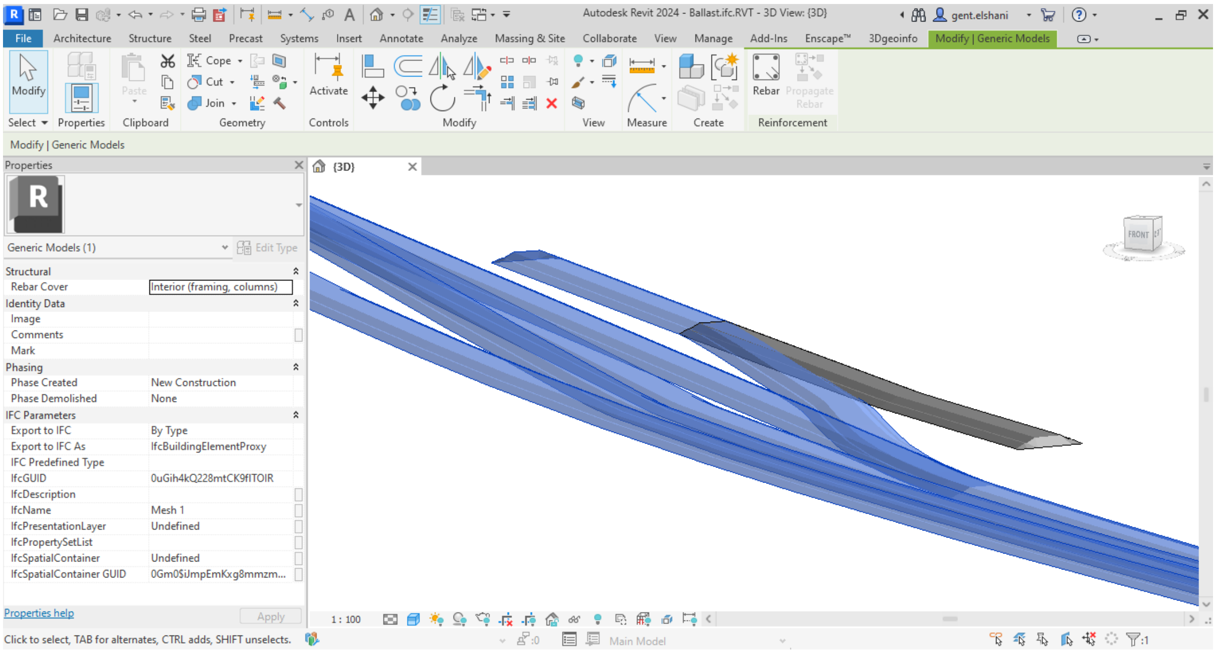
(A) Ballast_triangulated
(B) Rails_triangulated

(A) Object.0.3

Property	Value
Model	(A) Ballast_triangulated
Discipline	Architectural
Name	Mesh 1
Phase	
Type	
Type Name	
Predefined Type	

Figure 69: The IFC file of the railway track assets opened in Solibri.

The IFC file following the geometric representation of the faceted boundary representation is then imported into Revit. In Figure 70, the imported ballast of the railway tracks can be seen. An uncertain clustering of the components was noticed, with the model elements also not having any attributes or the possibility to modify them. These features indicated an inadequate setup of the elements, and the use of objects as a guide to create Revit elements was also not possible.


Autodesk Revit 2024 - Ballast.ifc.RVT - 3D View: {3D}

Modify | Generic Models

Properties

Generic Models (1)

Structural	
Rebar Cover	Interior (framing, columns)
Identity Data	
Image	
Comments	
Mark	
Phasing	
Phase Created	New Construction
Phase Demolished	None
IFC Parameters	
Export to IFC	By Type
Export to IFC As	IfcBuildingElementProxy
IFC Predefined Type	
IfcGUID	0uGih4kQ228mtCK9fITOIR
IfcDescription	
IfcName	Mesh 1
IfcPresentationLayer	Undefined
IfcPropertySetList	
IfcSpatialContainer	Undefined
IfcSpatialContainer GUID	0Gm0$iJmpEmKxg8mmzm...

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 70: Railway track ballast imported in Revit.

The main limitations of the presented workflow are related to the export capability of Navisworks, which is constrained to generating IFC files up to version 4, as of the current time. Further advancements in enabling the production of IFC4x3 files could enhance this workflow, as this release incorporates railway and bridge-specific objects in the IFC data schema. Modifications to the current workflow can be implemented by directly altering the IFC file to tailor it to the specific needs of the use case. This can be achieved using libraries that provide a basis on adjusting an IFC file using code, some of which are provided in Table 8 [34].

Table 8: Selected libraries that provide IFC programming, with their additional information [34].

Library	Programming language	Binding type
IFCOpenShell	C++ Python	Early
IFCOWL	Ontology Web Language	Early
IFCPlusPlus	C++	Early
xBIM	C#	Early

An alternative approach to the one previously discussed involved extracting only the track alignment from the shapefile and then generating the model geometry directly within Revit. To attain this outcome, the conversions depicted in Figure 71 have been executed. With a CSV file as the result, integration into Revit can be accomplished using a Dynamo script.

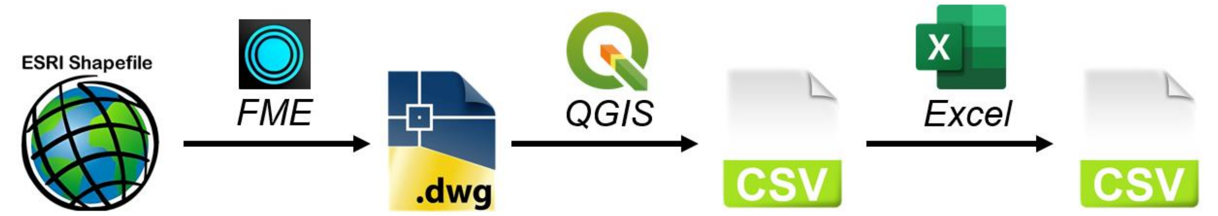
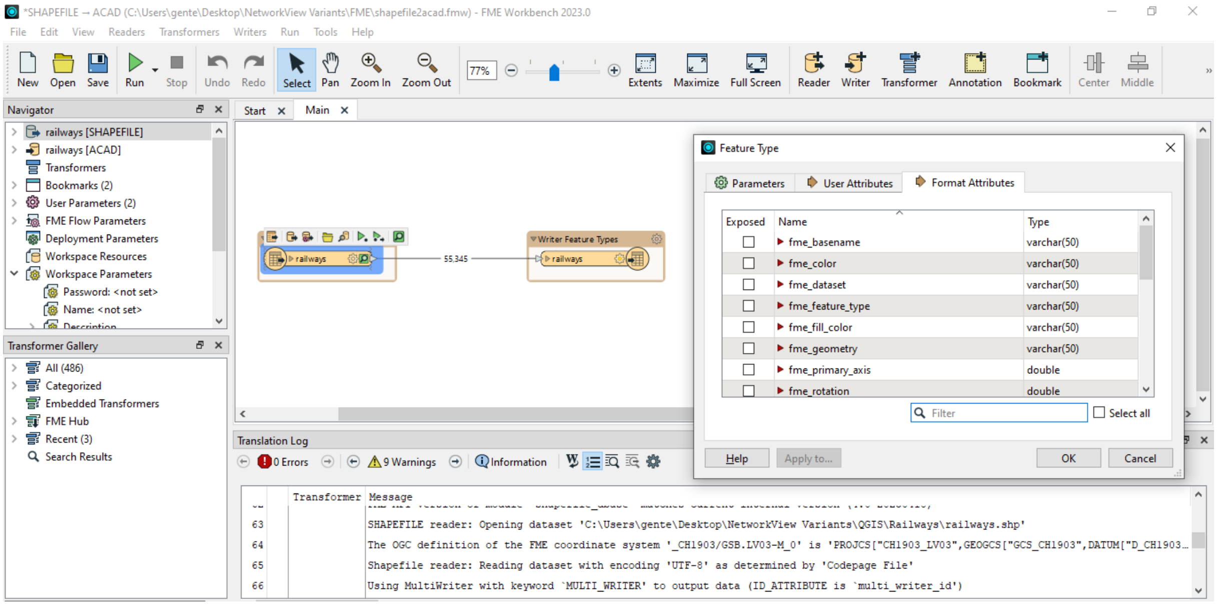


Figure 71: The procedure followed for the data conversion from ESRI Shapefile to CSV.

Firstly, the conversion from shapefile to DWG has been made. This is was done using a software called FME (Feature Manipulation Engine), which maps the instances from the data schemas to the corresponding matches of the target file format. The user interface and the attributes of the shapefile can be seen in Figure 72.



Exposed	Name	Type
□	fme_basename	varchar(50)
□	fme_color	varchar(50)
□	fme_dataset	varchar(50)
□	fme_feature_type	varchar(50)
□	fme_fill_color	varchar(50)
□	fme_geometry	varchar(50)
□	fme_primary_axis	double
□	fme_rotation	double

```
   Transformer Message
63             SHAPEFILE reader: Opening dataset 'C:\Users\gente\Desktop\NetworkView Variants\QGIS\Railways\railways.shp'
64             The OGC definition of the FME coordinate system '_CH1903/GSB.LV03-M_0' is 'PROJCS["CH1903_LV03",GEOGCS["GCS_CH1903",DATUM["D_CH1903…
65             Shapefile reader: Reading dataset with encoding 'UTF-8' as determined by 'Codepage File'
66             Using MultiWriter with keyword `MULTI_WRITER' to output data (ID_ATTRIBUTE is `multi_writer_id')
```

Figure 72: The conversion conducted in FME to transfer from Shapefile to DWG.

The conversion resulted in a DWG file containing the alignment of the railway network of Switzerland. By using a standard CAD software, such as AutoCAD, the DWG file can be opened. The leftmost part of Figure 73 displays the whole file, while the 3D lines represent the railway tracks. The area of interest has been filtered out by using the extents of the line from the SBB online database as a basis. In this way, the alignment depicted in the right side of Figure 73 is created. Considering that the file is available to be edited in AutoCAD, a segmentation of the track sections has been performed. Since the meaning of the particular track segments from a provided database by SBB was not known, a subdivision based on the intersections, curvatures, stations, and bridges has been performed. After acquiring the information about the location for the specific track segments, this segmentation can be further improved.

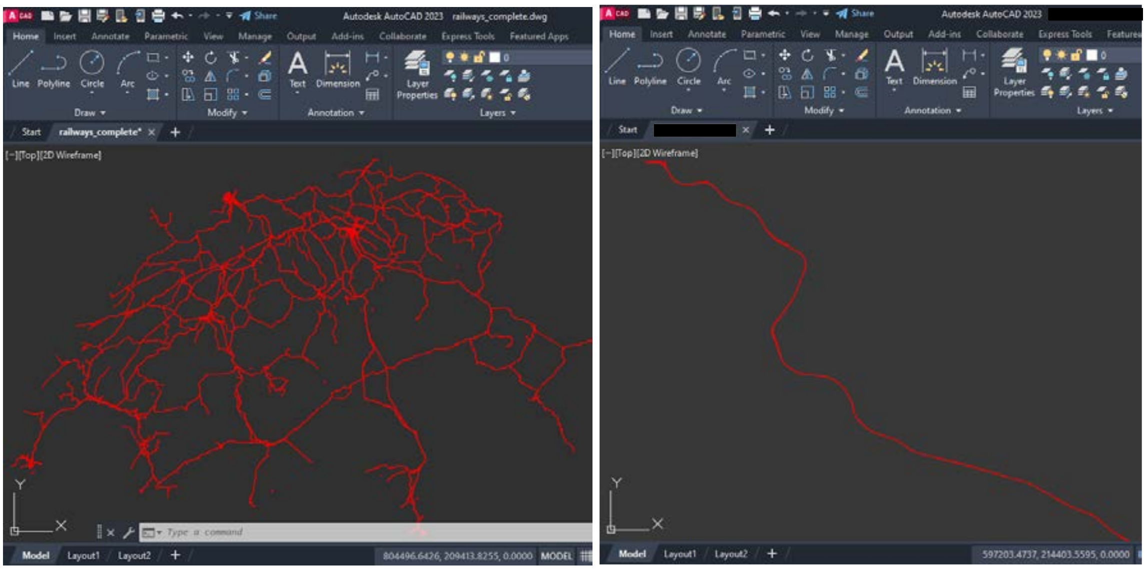
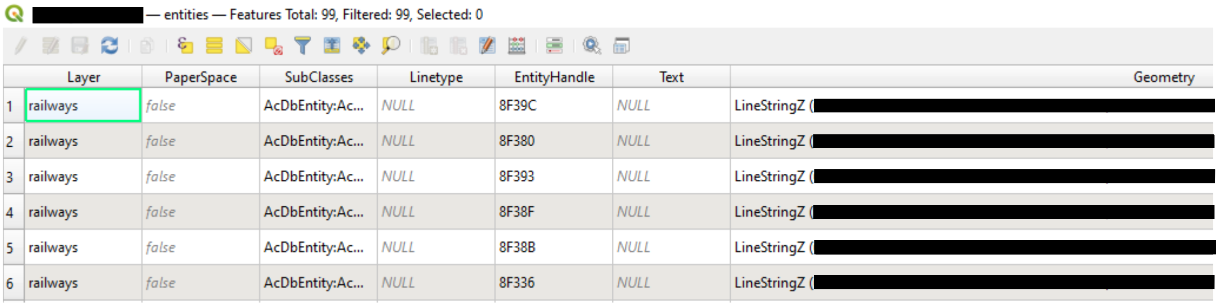

Autodesk AutoCAD 2023 railways_complete.dwg
Home Insert Annotate Parametric View Manage Output Add-ins Collaborate Express Tools Featured Apps
Line Polyline Circle Arc Text Dimension Layer Properties
Draw Modify Annotation Layers
Start railways_complete*
[–][Top][2D Wireframe]
Type a command
Model Layout1 Layout2
804496.6426, 209413.8255, 0.0000 MODEL
Autodesk AutoCAD 2023 [redacted]
Start [redacted]
597203.4737, 214403.5595, 0.0000

Figure 73: Filtering and segmentation in AutoCAD: before (left) and after (right).

The next step includes the integration of the data into a GIS software, for which the open-source program QGIS has been chosen. Since QGIS does not support the import of DWG files, a DXF file was generated from AutoCAD, which can then be included. A slight horizontal translation of the alignment in comparison to the underlying map can be observed, which presumably occurs as part of the file exchange.

Afterwards the attribute table of the imported DXF file has been accessed, including the geometry information of the alignment curves. Here (refer to Figure 74), every single row represents one polycurve of the track alignment. Additional information through a new field named ‘Geometry’ has been added, where the coordinates of each vertex of the alignment curves were intended to be saved. By entering the keyword ‘geom_to_wkt(@geometry)’ to the expression field, the information about the precise coordinates is resulted. Through the QGIS interface, this information could then be exported to a CSV file, and additional modifications have been planned for it.

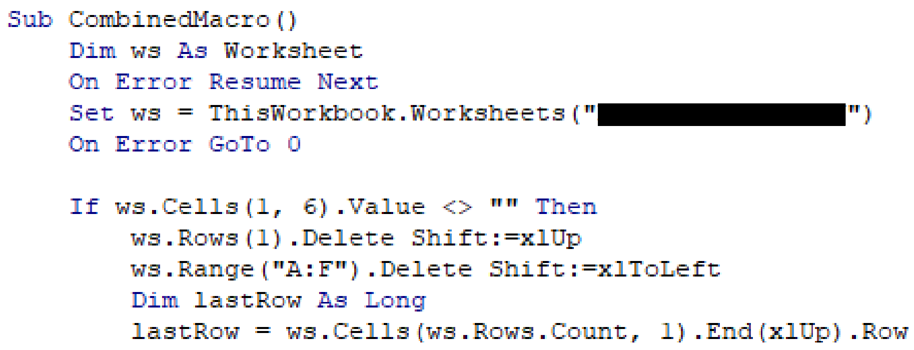


[redacted] — entities — Features Total: 99, Filtered: 99, Selected: 0

	Layer	PaperSpace	SubClasses	Linetype	EntityHandle	Text	Geometry
1	railways	false	AcDbEntity:Ac...	NULL	8F39C	NULL	LineStringZ ([redacted]
2	railways	false	AcDbEntity:Ac...	NULL	8F380	NULL	LineStringZ ([redacted]
3	railways	false	AcDbEntity:Ac...	NULL	8F393	NULL	LineStringZ ([redacted]
4	railways	false	AcDbEntity:Ac...	NULL	8F38F	NULL	LineStringZ ([redacted]
5	railways	false	AcDbEntity:Ac...	NULL	8F38B	NULL	LineStringZ ([redacted]
6	railways	false	AcDbEntity:Ac...	NULL	8F336	NULL	LineStringZ ([redacted]

Figure 74: Attribute table of the lines in QGIS.

Considering that the data stored in the CSV file was rather unorganized and included unnecessary string values, the file has been imported into Microsoft Excel. Using the macro functions of Excel and a custom developed macro (see Figure 75), the information in the file can be converted to a certain structure. This layout separates the coordinates for the x, y, and z values into different columns, and each distinct curve is divided by a blank row. The specific configuration is employed while generating the assets using these alignment curves. A more detailed description is provided in the following section.



```
Sub CombinedMacro()
    Dim ws As Worksheet
    On Error Resume Next
    Set ws = ThisWorkbook.Worksheets("[redacted]")
    On Error GoTo 0

    If ws.Cells(1, 6).Value <> "" Then
        ws.Rows(1).Delete Shift:=xlUp
        ws.Range("A:F").Delete Shift:=xlToLeft
        Dim lastRow As Long
        lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row
```

Figure 75: Part of the Excel VBA macro used for string manipulation.

The approach proposed in this Thesis encounters a limitation due to the available data during the case study. Therefore, the application of an alternative approach at the SBB might render the presented workaround obsolete, depending on the internal data structures and available resources.

3.2 Asset generation

The digital asset generation is divided into two parts: creating the railway tracks and modeling the bridges along the line. Figure 76 illustrates the approach to create digital asset models for both sectors. The previously prepared CSV file is used in the track generation by running a custom-made Dynamo script. In the case of creating digital bridge models, modeling them directly in Revit facilitates the data transfer process, as a simple Revit import can be used. Of particular interest are the distinct coordinate systems of the assets. In the case of the railway tracks, the data provided by Swisstopo is available in the Switzerland-specific CH 1903/LV 03 coordinate system, contrary to the bridge assets where the information about their geoposition is accessible in the commonly used WGS 84 system. A conversion between these two systems must be achieved to provide consistent results.

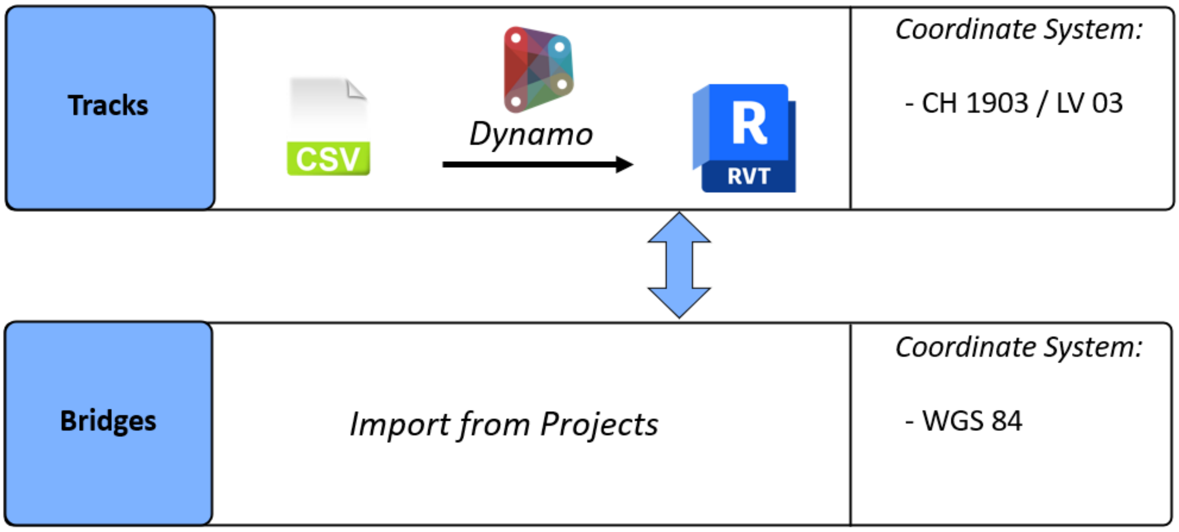


Figure 76: The selected import options of the assets with different coordinate systems in Revit.

Track generation

The following Dynamo script has been created to generate the tracks in Revit by using their alignment as a basis (refer to Figure 77). The grouping of nodes in individual blocks has also been applied in this Dynamo code for a better overview of the script. In the next part, the code is clarified in greater detail.

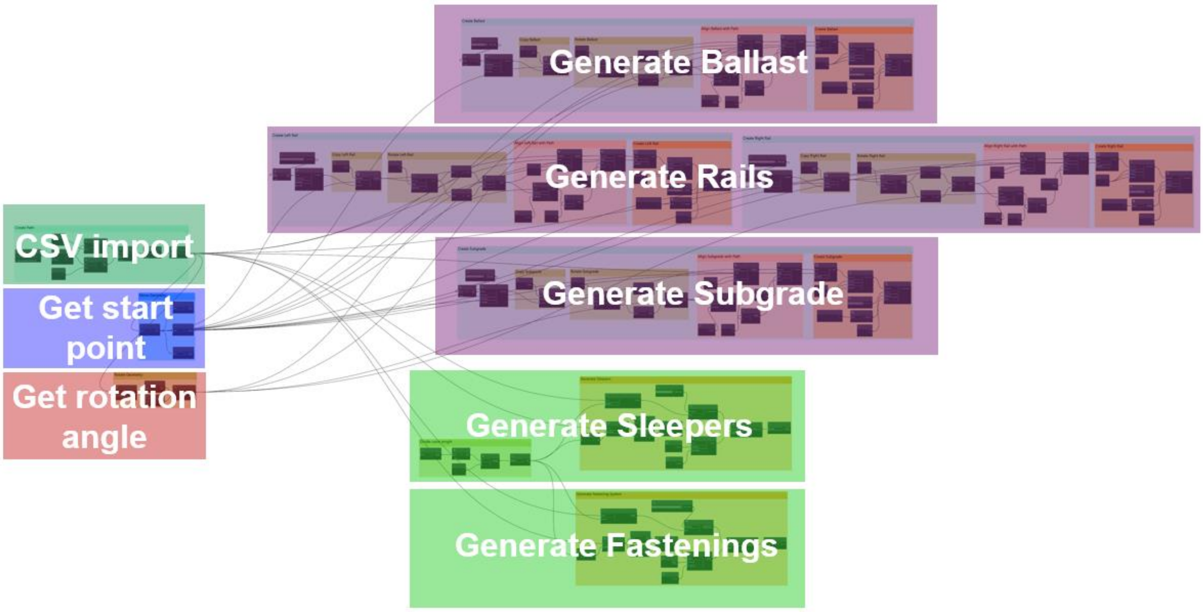


Figure 77: A simplified view of the Dynamo script used to generate the track assets.

Two different approaches have been followed to generate the named assets. While the Ballast, Rails, and Subgrade have an extrusion-based creation, the sleepers and Fastening systems are constructed as instance-based geometry; thus, a grouping based on these concepts is presented.

CSV import

The first step of generating the assets using the script is to import the track alignment data into Dynamo (refer to Figure 78). This can be done by navigating to the file location containing the coordinates of all points of the polycurves. By getting the first and second index of these points, the x and y coordinates can be integrated into the script. Since importing the z coordinate into the model provided inconsistent results of the generated assets in the end, the track generation was limited to a 2D representation of the network. However, this approach produced results sufficient for the specific use case of the network.

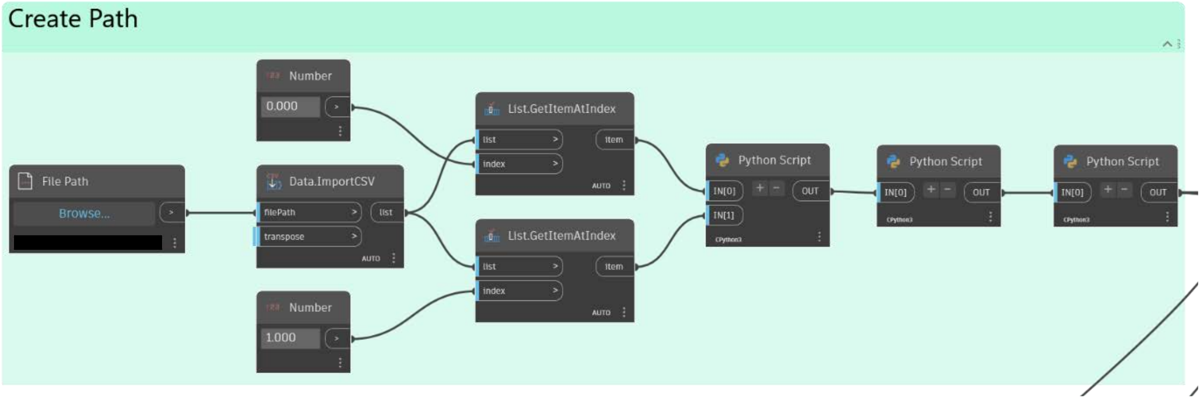
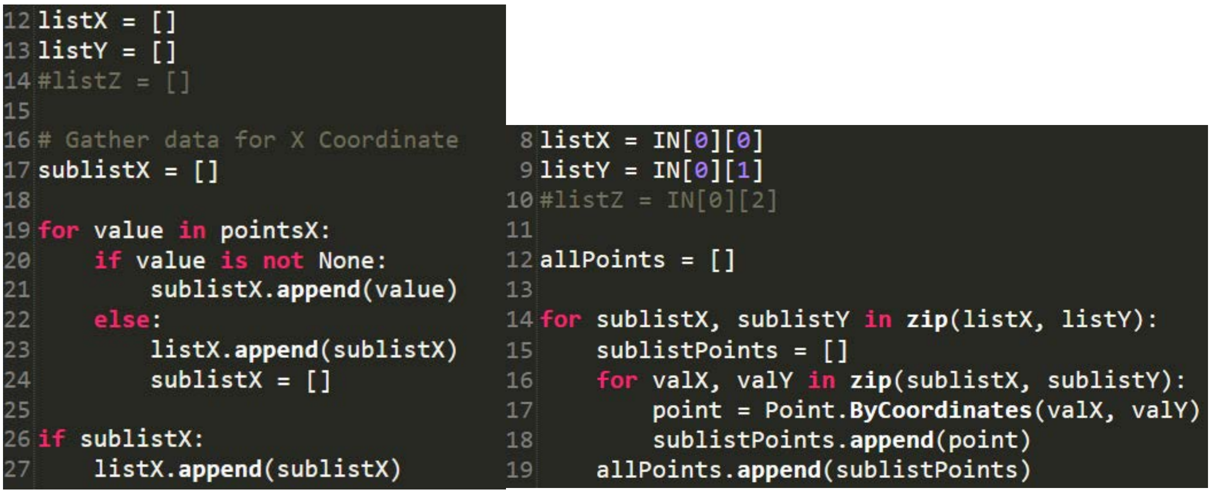


Figure 78: Nodes used to create the alignment path from the CSV file.

After obtaining all the coordinate data of every single vertex of the curves as floating-point number information, three separate Python nodes have been deployed. The aim to generate a set of points inside Dynamo has been achieved in the first two Python nodes (see Figure 79), where first, the distinct curves have been separated into individual lists by using the blank entries in the CSV file as a guide. Afterwards, in the second Python node, the Autodesk.DesignScript.Geometry has been used to create Dynamo instances of the points grouped in the same manner as in the previous node.

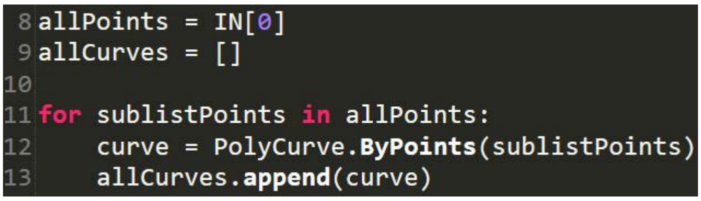


```python
listX = []
listY = []
#listZ = []

# Gather data for X Coordinate
sublistX = []

for value in pointsX:
    if value is not None:
        sublistX.append(value)
    else:
        listX.append(sublistX)
        sublistX = []

if sublistX:
    listX.append(sublistX)
```

```python
listX = IN[0][0]
listY = IN[0][1]
#listZ = IN[0][2]

allPoints = []

for sublistX, sublistY in zip(listX, listY):
    sublistPoints = []
    for valX, valY in zip(sublistX, sublistY):
        point = Point.ByCoordinates(valX, valY)
        sublistPoints.append(point)
    allPoints.append(sublistPoints)
```

Figure 79: Snippet of Python nodes employed to extract the single coordinates (left) and to create a set of Dynamo points using the coordinates (right).

Once all the points are created in Dynamo by using the PolyCurve object, the final alignments can be created using the node, part of which is shown in Figure 80. An approach utilizing curves of higher polynomial degrees, such as NURBS has been explored. However, based on the given data for the nodes, only an approximation could be achieved, which is not certain to represent the actual state of the tracks. Also, since the continuity requirement for segmenting the tracks does not require a continuity higher than C1, the method of using polycurves for the alignment has been selected.



```python
allPoints = IN[0]
allCurves = []

for sublistPoints in allPoints:
    curve = PolyCurve.ByPoints(sublistPoints)
    allCurves.append(curve)
```

Figure 80: Snippet of the Python nodes used to create a polycurve along the points in Dynamo.

Finally, the alignment has been generated in order to be further used in the script. In Figure 81, a part of the alignment can be observed. The leftmost side of Figure 81 shows the generated points in Dynamo, whereas the right part depicts the polycurves created from them.

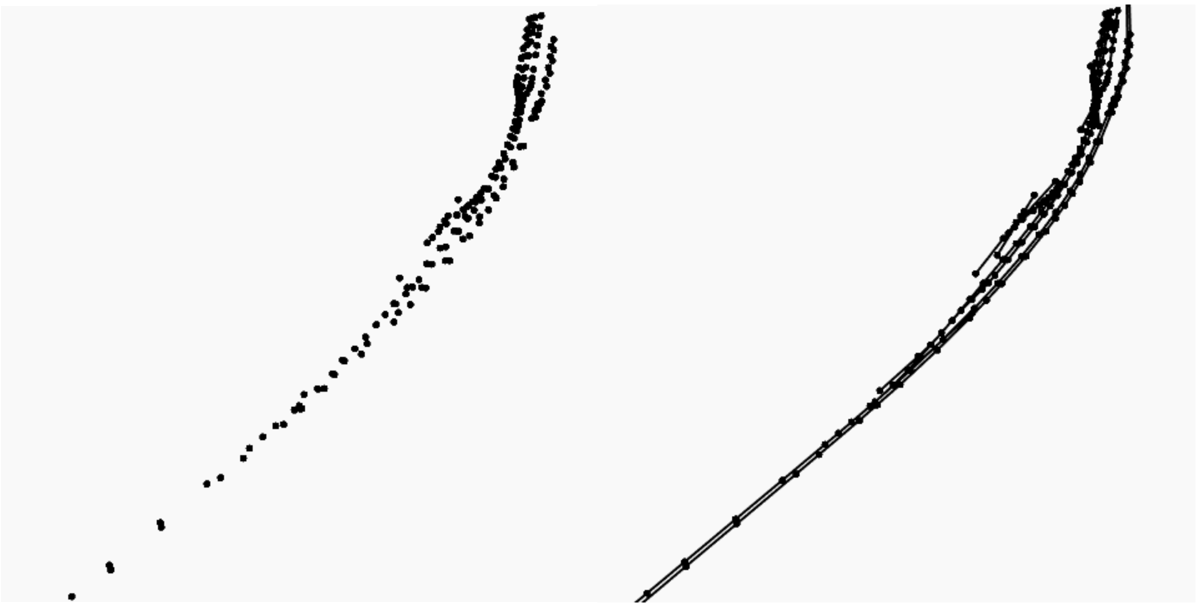


Figure 81: A set of Dynamo points (left) and curves (right).

Get start point

The group of nodes in Figure 82 take every start point of the separate alignment curves and stores their respective x, y, and z coordinates. This information is further used as the location in which the profiles and family instances get copied. More details are provided in the separate groups of nodes where the assets are generated.

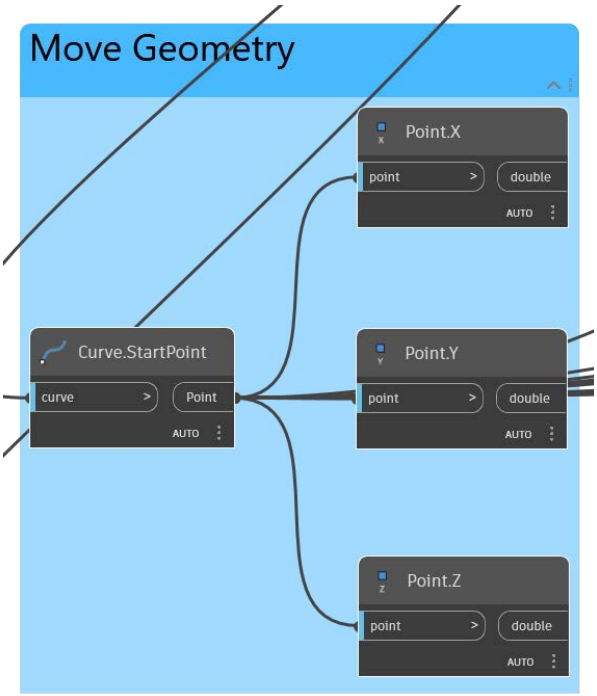


Figure 82: Nodes used to store the start point of every alignment curve.

Get rotation angle

In the group of nodes in Figure 83, a line tangent to the start point of each alignment curve is created. By constructing a vector using this line as a reference, a basis can be established which is utilized to extract the angle in which the profiles and family instances must be rotated when generating the assets.

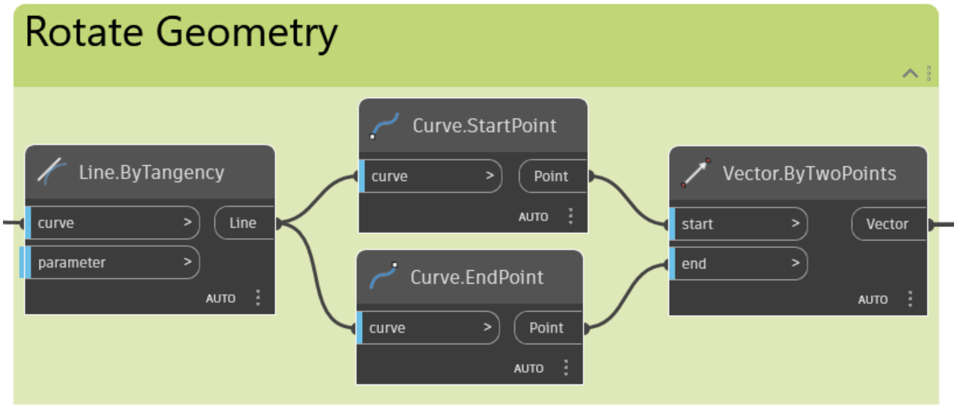


Figure 83: Nodes used to store the vectors tangent to the start point of every alignment curve.

Generate ballast, rails, and subgrade

In order to create the digital models for the railway track assets representing the ballast, rails, and subgrade, the group of nodes shown in Figure 84 has been used. These assets have been grouped together because of the similar approach used to create them: creation through extrusion. The geometry can be produced by defining the cross-section and the path over which it will be generated. This methodology employs the alignment as a reference for the extrusion path.

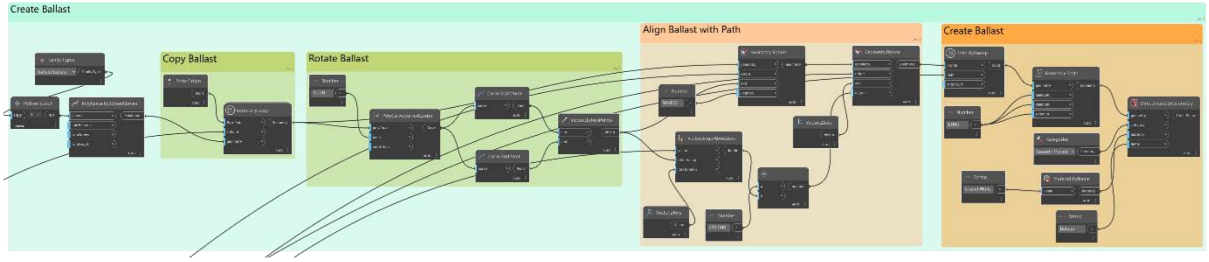
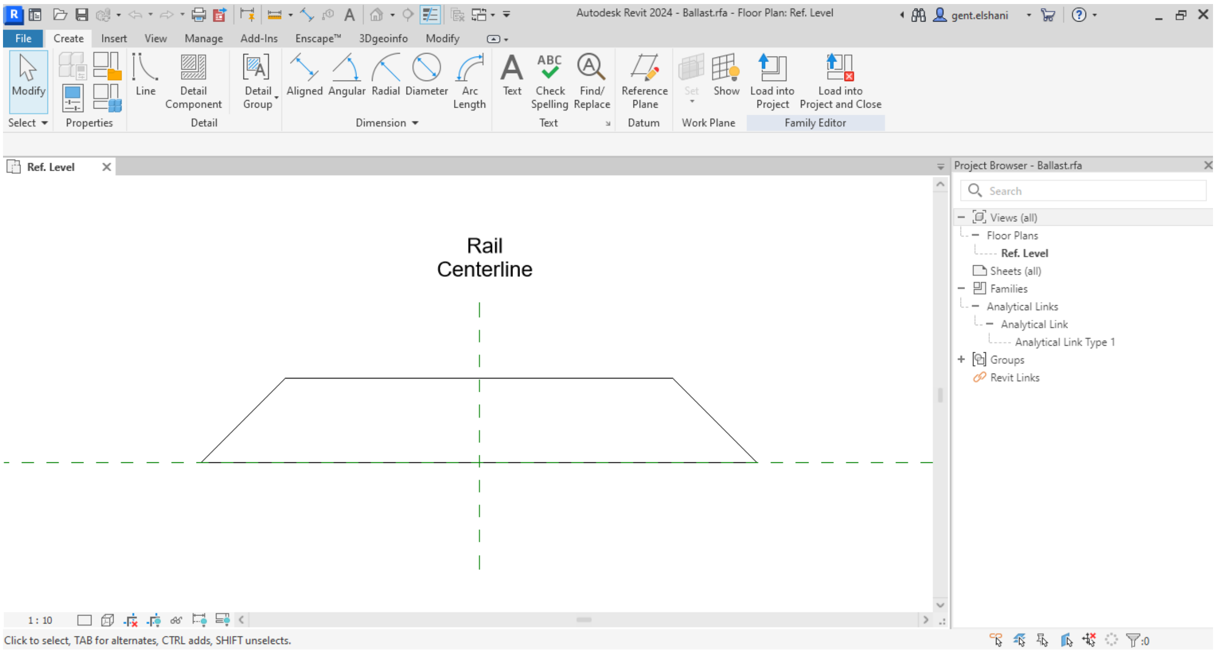


Figure 84: Nodes utilized to create the extrusion-based model elements.

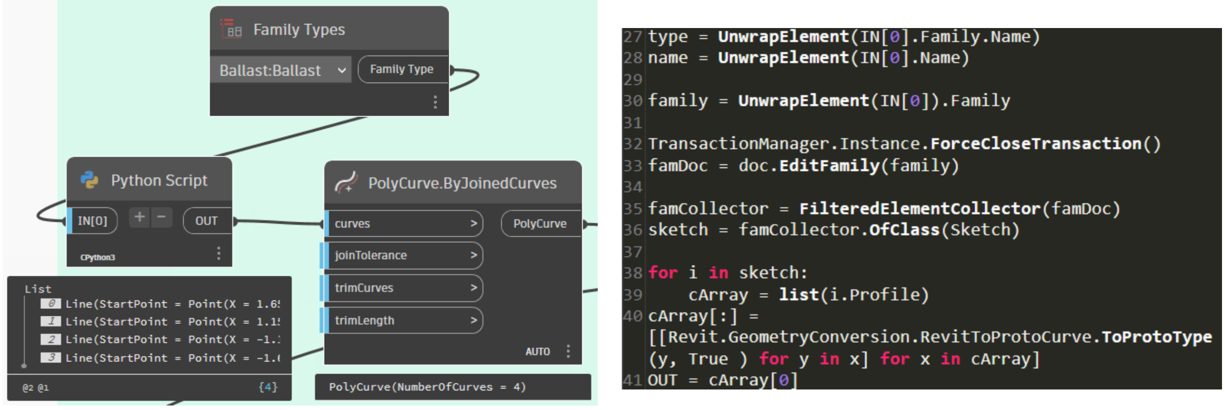
As a basis for the cross-section, the Revit profile families have been used. These families contain only 2D drawings and can be used only in specific settings and not as geometric instances on their own in the BIM model. Figure 85 shows the profile family employed for the track ballast.



Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 85: A view of a profile family in Revit.

The first part of the nodes used in Figure 86 imports the Revit profile family into Dynamo using a custom-made Python script. This script extracts the single lines from the profile by unwrapping them and then storing them into a list. The list elements are then used to form a polycurve in Dynamo.



```
List
  0 Line(StartPoint = Point(X = 1.65
  1 Line(StartPoint = Point(X = 1.15
  2 Line(StartPoint = Point(X = -1.1
  3 Line(StartPoint = Point(X = -1.6
@2 @1                          {4}
```

```
PolyCurve(NumberOfCurves = 4)
```

```python
type = UnwrapElement(IN[0].Family.Name)
name = UnwrapElement(IN[0].Name)

family = UnwrapElement(IN[0]).Family

TransactionManager.Instance.ForceCloseTransaction()
famDoc = doc.EditFamily(family)

famCollector = FilteredElementCollector(famDoc)
sketch = famCollector.OfClass(Sketch)

for i in sketch:
    cArray = list(i.Profile)
cArray[:] =
[[Revit.GeometryConversion.RevitToProtoCurve.ToProtoType
(y, True ) for y in x] for x in cArray]
OUT = cArray[0]
```

Figure 86: Nodes used to create a Dynamo polycurve from the Revit profile family (left) and a snippet of the Python node (right).

The next step involves copying the profile instance from its original location to the starting point of each alignment curve (see Figure 87). The coordinates previously stored from the nodes used to obtain the starting points have also been applied here. In this way, a physical connection between the profile instances and the start point of the extrusion paths is established.

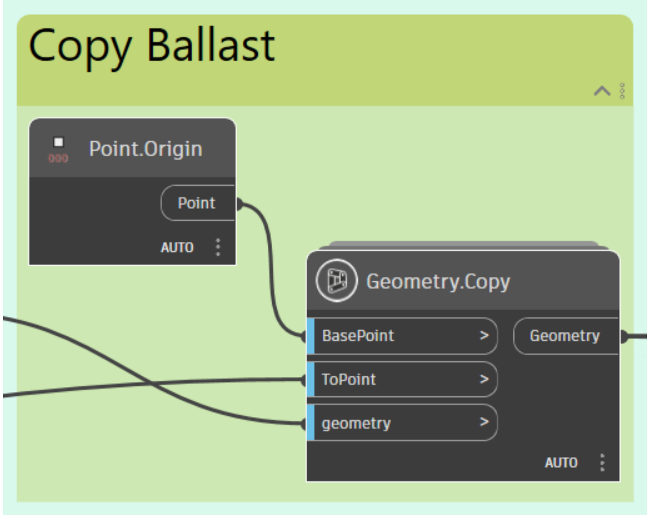


Figure 87: Nodes used to copy the profile to the start point of each alignment curve.

After placing one instance of the Revit profiles on each curve, the shape rotation needs to be carried out to position the profile perpendicularly to each alignment. The position of the profile family can be obtained by extracting the line at the base of the ballast section and creating a vector that serves as a reference (see Figure 88).

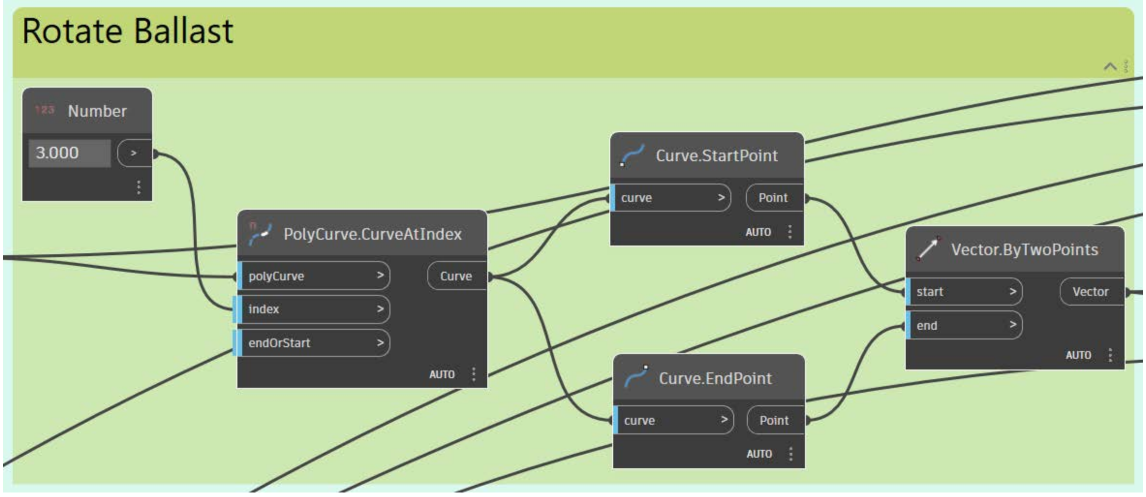


Figure 88: Nodes utilized to acquire the information about the profile rotation angle.

In the next set of nodes shown in Figure 89, the rotation of the profile families has been performed. By using the vector of the profile and the vectors tangent to the start point of every alignment, the angle between them is calculated. This has been accomplished employing the Vector.AngleAboutAxis node in Dynamo using the z-axis as the reference axis. Using the mentioned node is crucial rather than the similar node called Vector.AngleWithVector, which expects only two vectors. The latter one can only return angles up to 180 degrees, leading to inconsistent results when used. An additional rotation around the y-axis has also been performed.

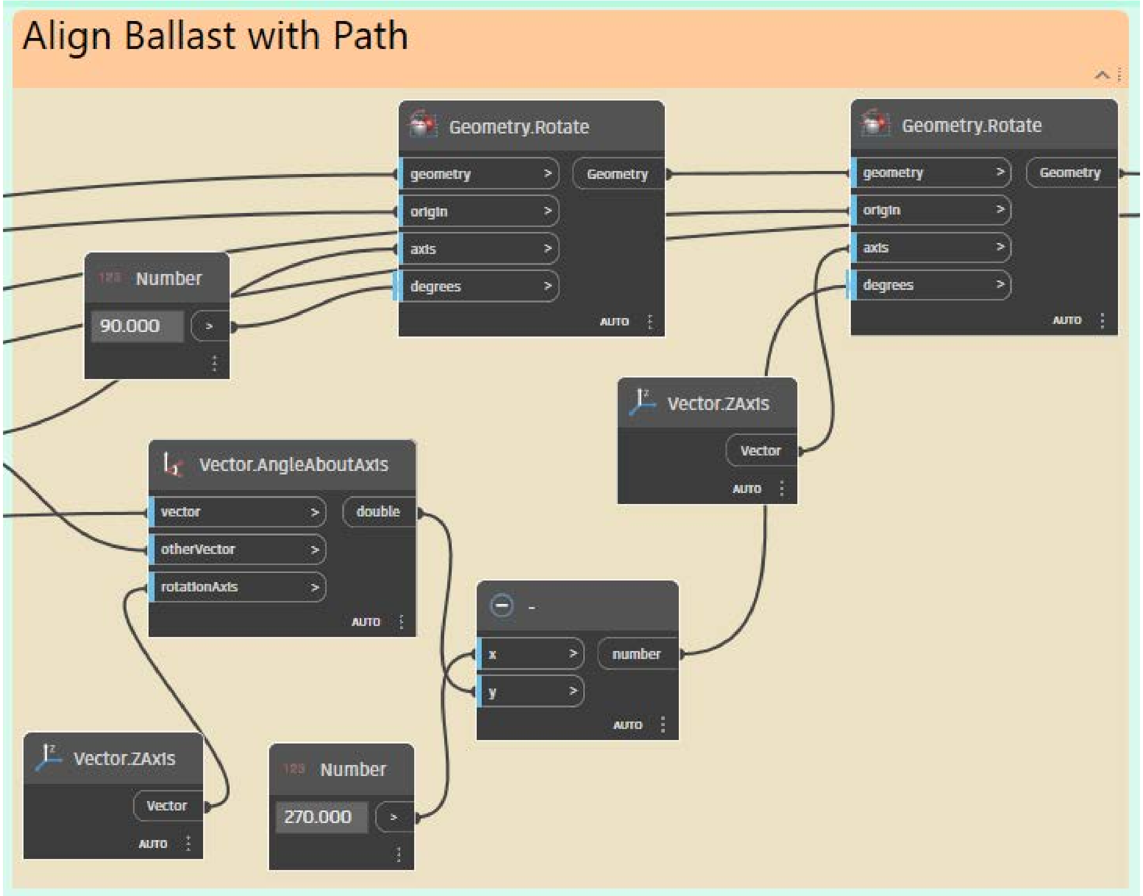


Figure 89: Nodes used to rotate the profile according to the vector tangent to the start point of every alignment curve.

Using the correctly positioned and rotated profile family and the alignments of the tracks, the final geometry generation can be carried out in the set of nodes illustrated in Figure 90. Solid bodies have been generated by creating a sweep using these instances. A scaling option has also been included, making it possible to modify the expansion size. By including other parameters, such as the material category and name, a DirectShape has been produced in the BIM authoring tool.

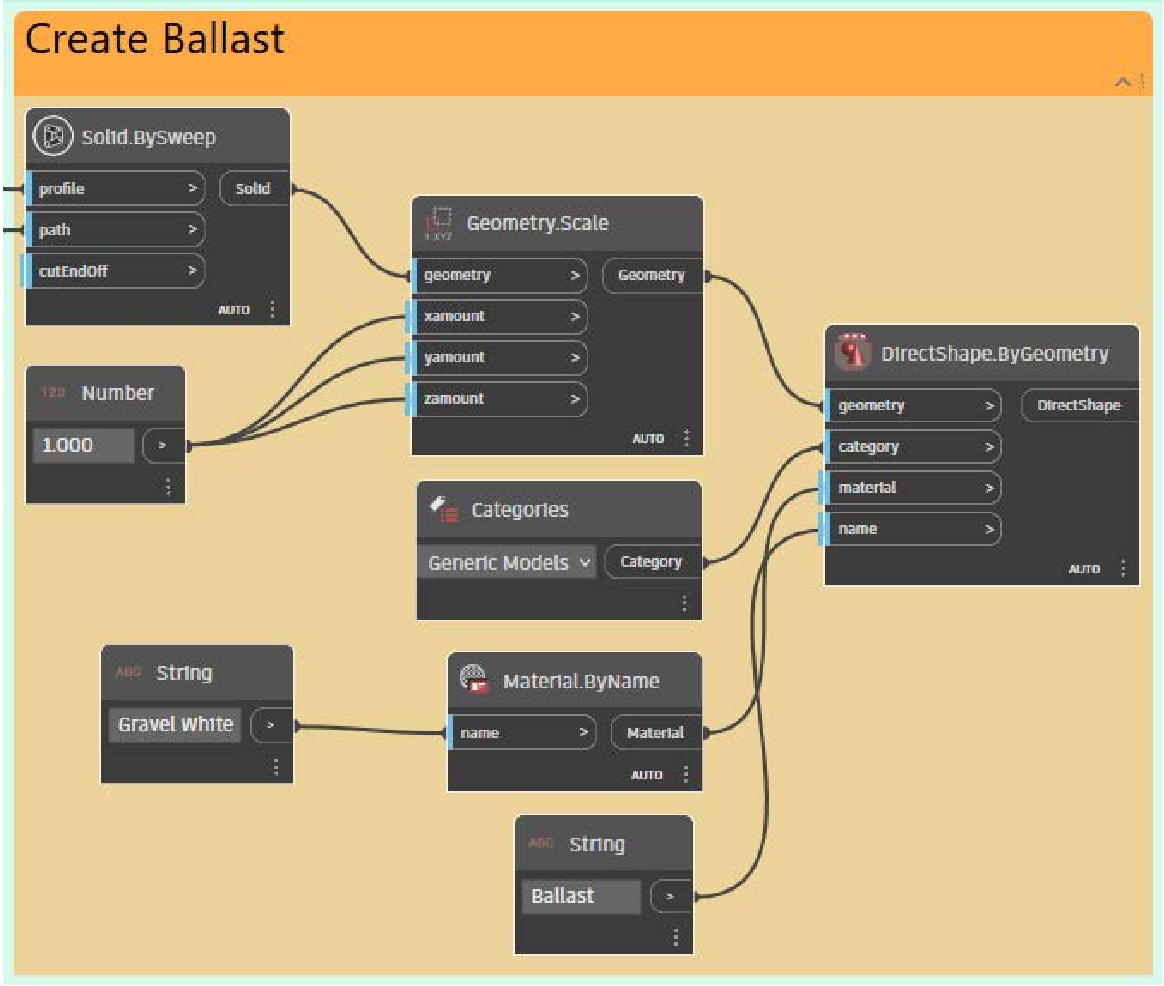


Figure 90: Nodes applied to create a model element through the profile and path.

A similar approach has been used to create the subgrade and both rails of a track section. Since all these components use the same alignments as a basis, the complete track assets can be generated by changing only the Revit profile family. Since the created components represent an implicit form of geometric modeling, the data size will not significantly extend compared to other methods of representing geometry.

Generate sleepers and fastening systems

A different approach has been followed when creating the geometry for the sleepers and fastening systems of the track section. Since these components are not continuous over the alignment but are rather placed over a set distance, creating an instance of the objects every 0.6 meters has been selected. Because the sleepers and fastening systems are placed at the same location in relation to each other, their generation has been grouped together. Figure 91 illustrates the asset creation of the sleepers as part of the railway tracks.

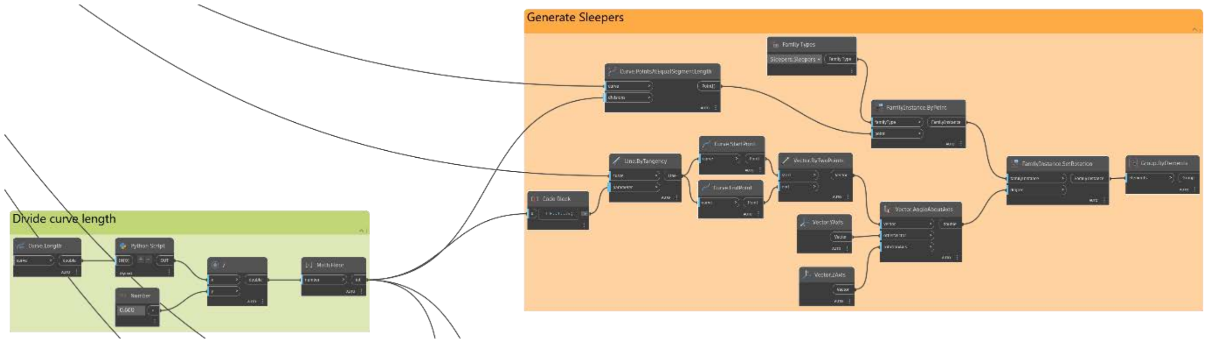
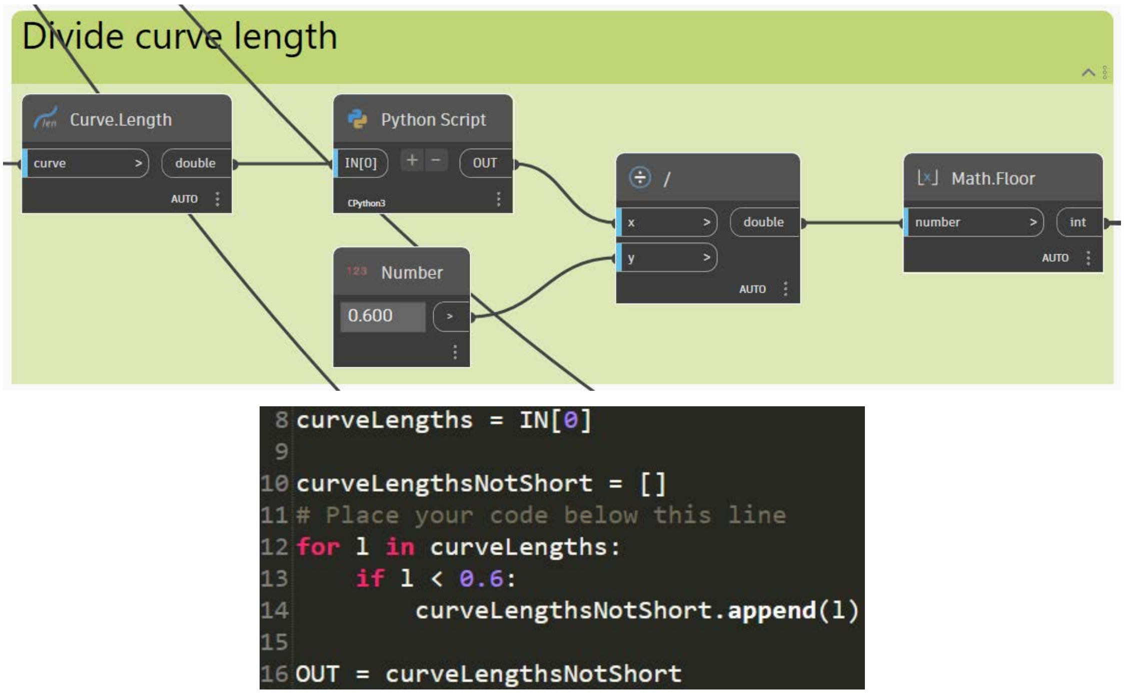


Figure 91: Nodes utilized to create the instance-based model elements.

The first group of nodes in Figure 92 introduces the division of the alignment curves into separate parts using the set distance of 0.6 meters. Before this dissection, a Python script filters out curves that are smaller than the given distance to avoid unexpected behavior later on. This same division has been used for the sleepers and the fastening systems.



```
curveLengths = IN[0]

curveLengthsNotShort = []
# Place your code below this line
for l in curveLengths:
    if l < 0.6:
        curveLengthsNotShort.append(l)

OUT = curveLengthsNotShort
```

Figure 92: Nodes employed to divide the alignment curves (top) and a snippet of the Python node to remove short lines (bottom).

In this section, the asset generation of the sleepers is explained in more detail (refer to Figure 93). Firstly, a number of points have been placed on the alignment curves based on the division performed earlier. Afterwards, a family instance of the sleeper was created on those points. A more simplified family than the one presented in the family catalog was selected because of the large number of instances that needed to be included in the network view. Following the placement, the rotation angle between the vector tangent to each point and the y-axis has been measured, similarly to the approach on the extrusion-based asset generation. In this way, the correct placement of the sleepers could be ensured along the alignment. A grouping of the sleepers along the different alignments has been done to facilitate the handling of the created geometry. Analogously, the fastening systems have also been generated with the only distinction of using a different family instance as a basis.

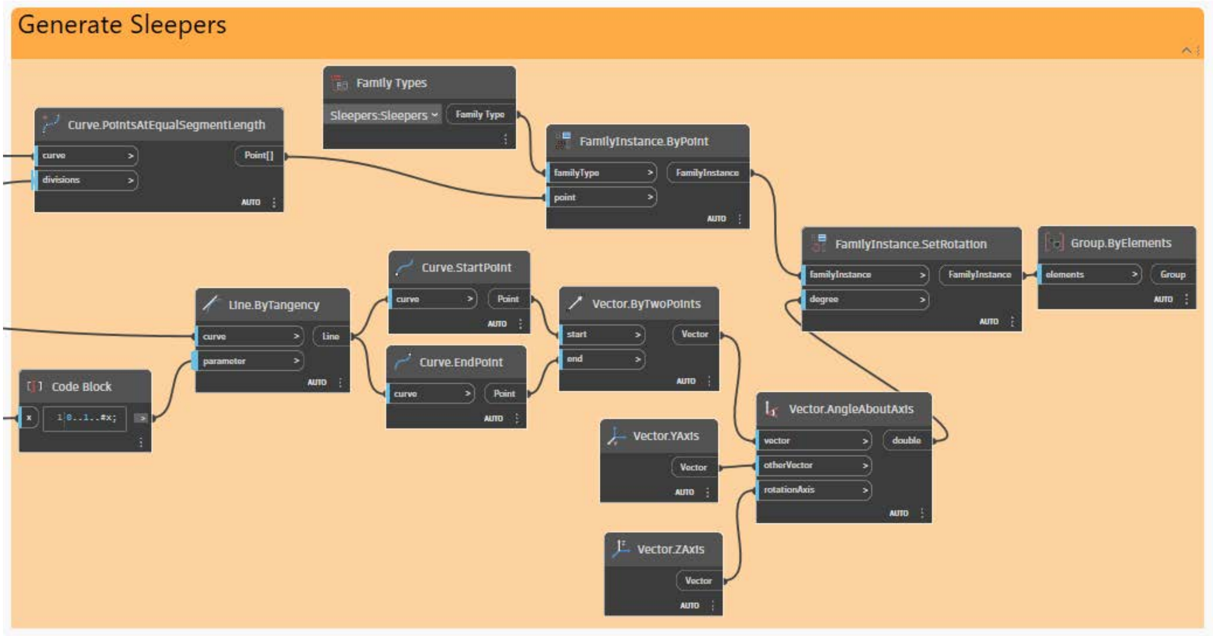


Figure 93: Nodes used to generate the family instances based on the separation and angle of the alignment curves.

The process of generating separate instances for each sleeper and fastening system can be optimized in the future, especially when considering the exponential growth in data size when modeling multiple railway lines to create a comprehensive digital representation of Switzerland ‘s whole railway network.

Lastly, the detail of the generated components should comply with the use case of this Thesis, leading to a complete coverage of all the track assets in the treated line. A part of the resulting railway track assets created with the proposed methodology can be found in Figure 94. It should be noted that the representation of the separate components has been achieved using simplified shapes. Additionally because of the high demand of computational power when generating the instance based components, a separate execution of parts of the script was performed. With more detailed information at hand, the cross-section of the tracks can also be optimized, leading to a more accurate outcome that can be used in further use cases of BIM.

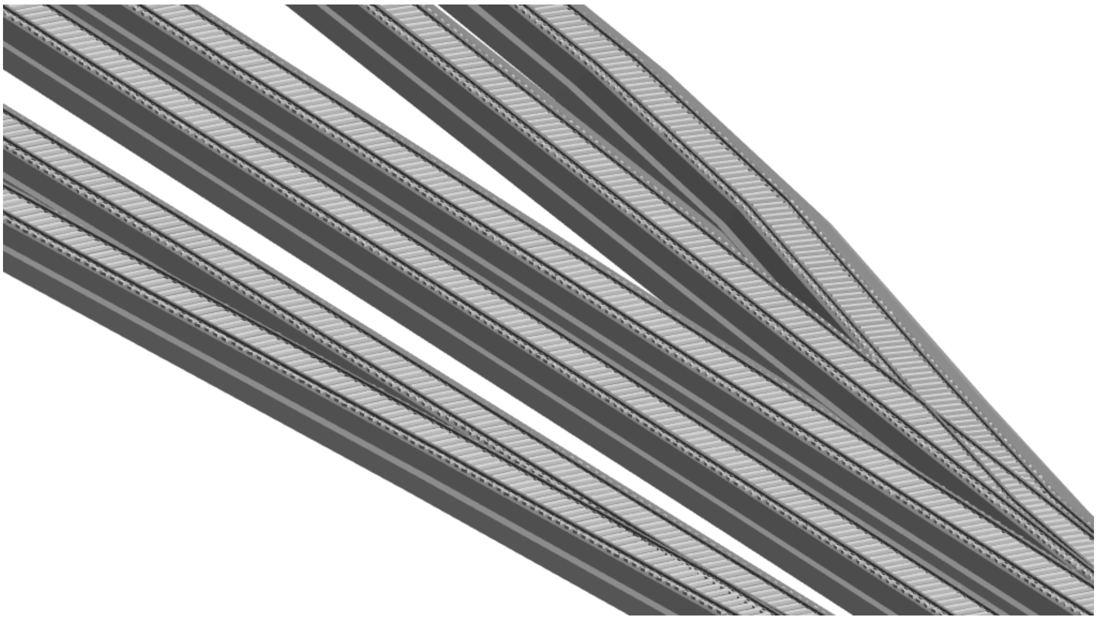


Figure 94: A view of the generated railway track components in Revit.

Bridge generation

To incorporate all the bridge assets existing on the line into the network view, direct modeling in Revit has been employed. In this way, no data conversion had to be performed between different authoring tools, and the same parameters could be used.

Since four bridge models have been already created (see Figure 95) using the available construction drawings as a reference and applying the family catalog to model them, the remaining bridges have also been included in the scope of this Master's Thesis.

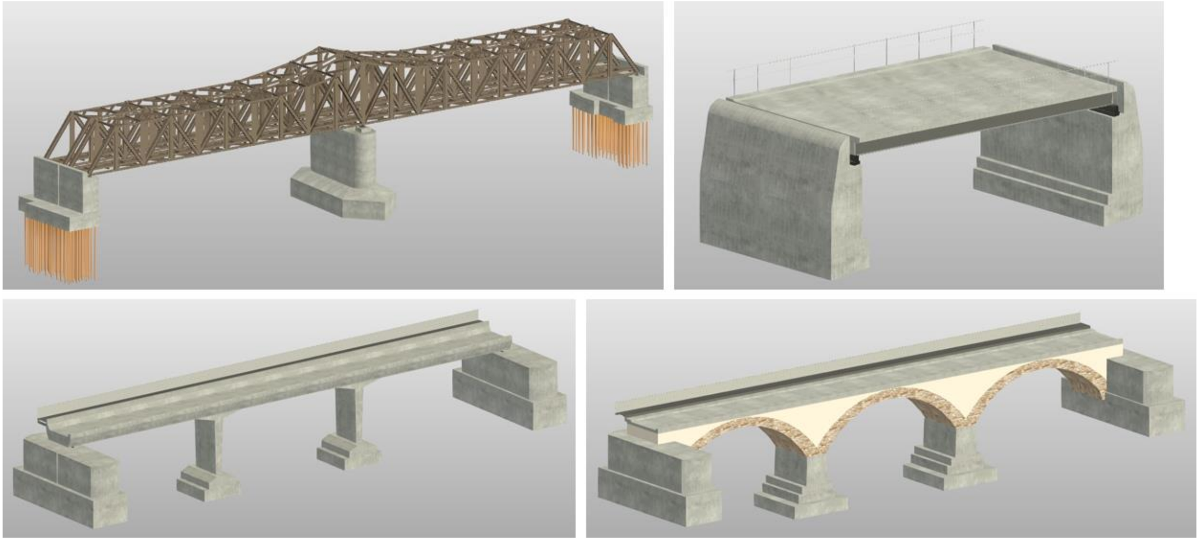


Figure 95: Visualization of bridge models with a higher level of detail.

Considering the lack of further geometric information regarding the rest of the bridges, the assets have been modeled using satellite and Google Street View images. Since this was the only data available for the bridges, a LOD of 200 can be assumed, providing a rough representation of the geometry. However, this level of development can facilitate the presented use case to represent the network as completely as possible. The given assets can be additionally refined and modified, assuming the information is available. In Figures 96 and 97, the further modeled concept bridges of the network can be observed.

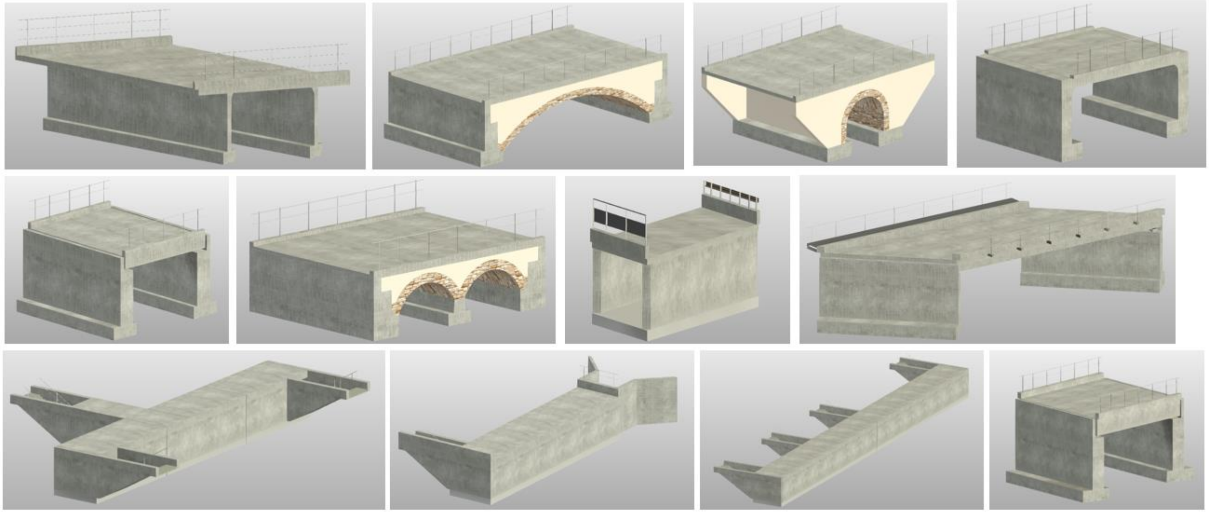


Figure 96: Visualization of 12 bridge concept models.

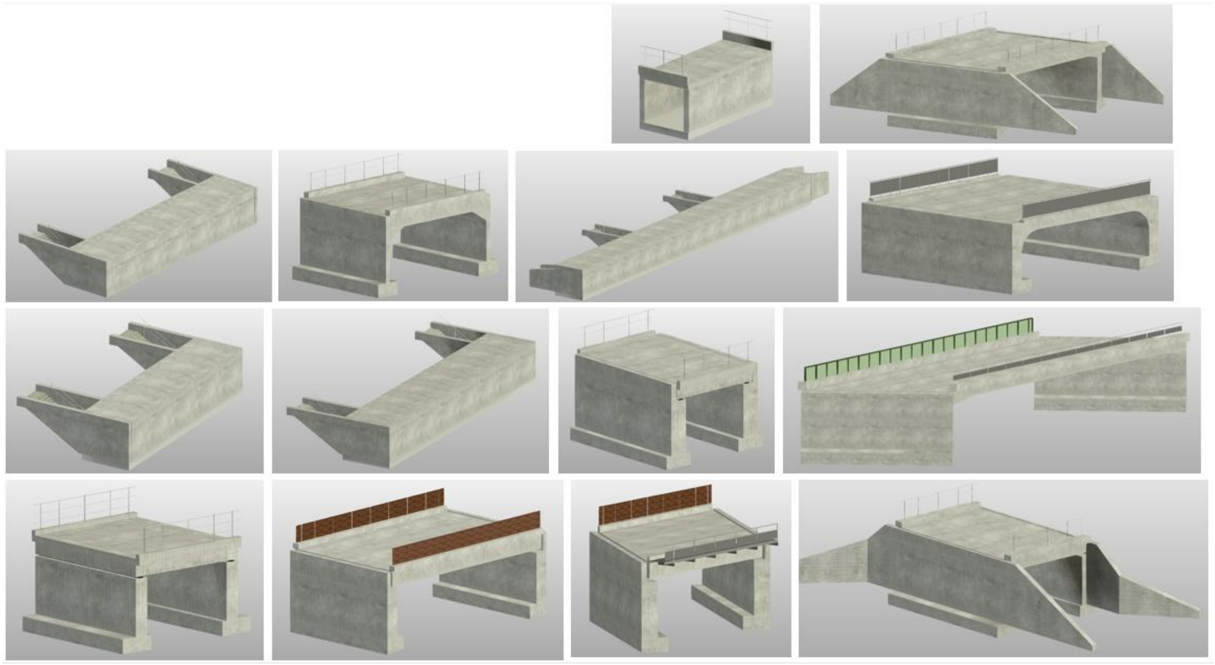


Figure 97: Visualization of 14 bridge concept models.

After generating the bridge assets, the assembly of the network view has been completed by inserting all the bridge components into the file containing the railway track assets. In order to place the bridges in the correct position, the geoposition information was obtained from the SBB online database. This was performed for all bridges where slight translations and rotations have been induced. Automation can be further established by using the shared coordinates in Revit to include the rotation and accurate placement of the bridges into the network.

The small displacements concerning the track positions may be attributed to model's relatively significant distance from the project base point in Revit. Other possible reasons include the data quality of the railway tracks or bridge coordinates which might not be consistent. A misinterpretation while mapping between different file formats or coordinate systems could also be the case.

3.3 Visualization

For visualizing the information embedded in the assets of the network view, a similar approach to the asset level visualization of the separate bridges has been adopted. This includes showing the data through colors and transparency directly on the 3D model and integrating the Dynamo Player into the workflow.

A distinct separation of the Dynamo scripts has been done according to the division in Figure 98. Here, separate scripts were created depending on the assets treated. The general categorization of the scripts includes the bridges and tracks of the line separately, where an analysis at the asset level can be conducted. In order to achieve this, the scripts enabling the visualization of the condition states and interventions have been incorporated. Following this approach, a separate category of the network has been implemented, where all the assets can be evaluated at the same time.

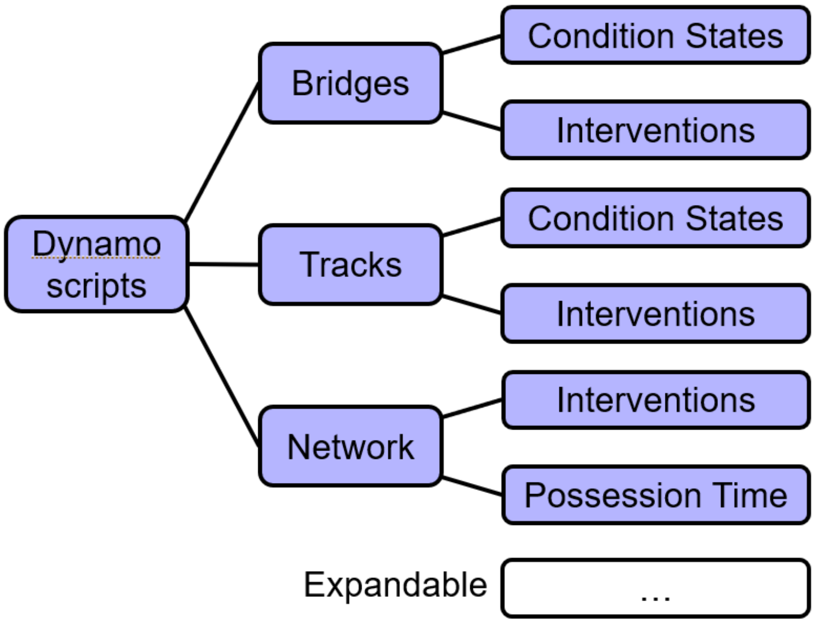
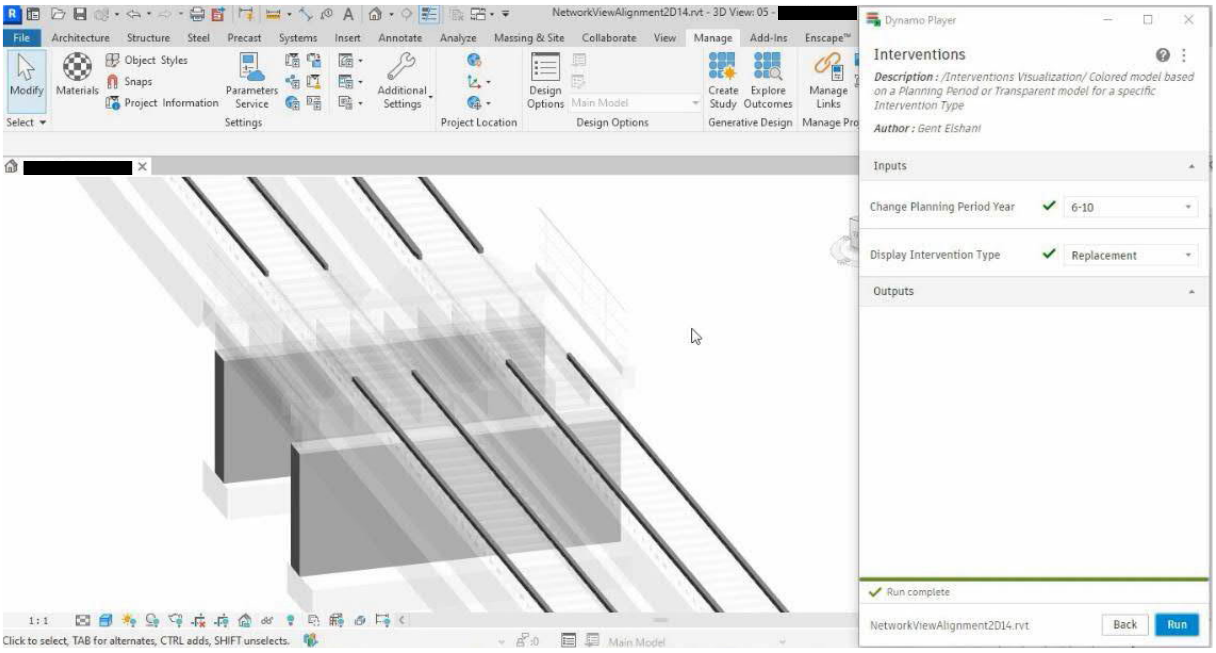


Figure 98: Dynamo script categorization of the network view.

While incorporating the script for visualizing necessary interventions on the components, the condition state script has not been added at the network level. This decision was made with the intention that the distinct asset managers would initially be responsible for their respective assets in the early stages. Collaboration using the network will be then done while planning interventions on a larger scale. With the inclusion of the network view, a clustering of the interventions can be performed in an easier way. Additionally, a script to visualize the possession time of each component has been added. Following this approach, the workflow can accommodate any number of Dynamo scripts based on the information integrated within the components and the specific use case being addressed.

Regarding the project setup and the transfer of the given setup, the same approach mentioned in Section 2.3.1 has been chosen. This configuration includes the integration of project and global parameters, as well as view filters. A differentiation can be made in the view setup selected for the network view, where a site view from above and an orthographic 3D view for the bridges and tracks separately, and the whole network were included in Revit. A station-to-station separation has been additionally incorporated to provide a better overview of the assets in a specific region.

Figures 99 and 100 depict the visual representation achieved after implementing the interventions script. The highlighted components represent all objects where interventions are being scheduled in the planning period of six to ten years in the future. A better perception of the interventions that must be performed can be obtained by providing a combined visual aid of both bridges and railway tracks. Clustering the interventions into groups, which can be performed in the same time slot, is also a very important topic that can be accomplished in a simpler way through the usage of the network view.



NetworkViewAlignment2D14.rvt - 3D View: 05 - [redacted]

[redacted]

Interventions

Description : /Interventions Visualization/ Colored model based on a Planning Period or Transparent model for a specific Intervention Type

Author : Gent Elshani

Inputs

Change Planning Period Year ✓ 6-10

Display Intervention Type ✓ Replacement

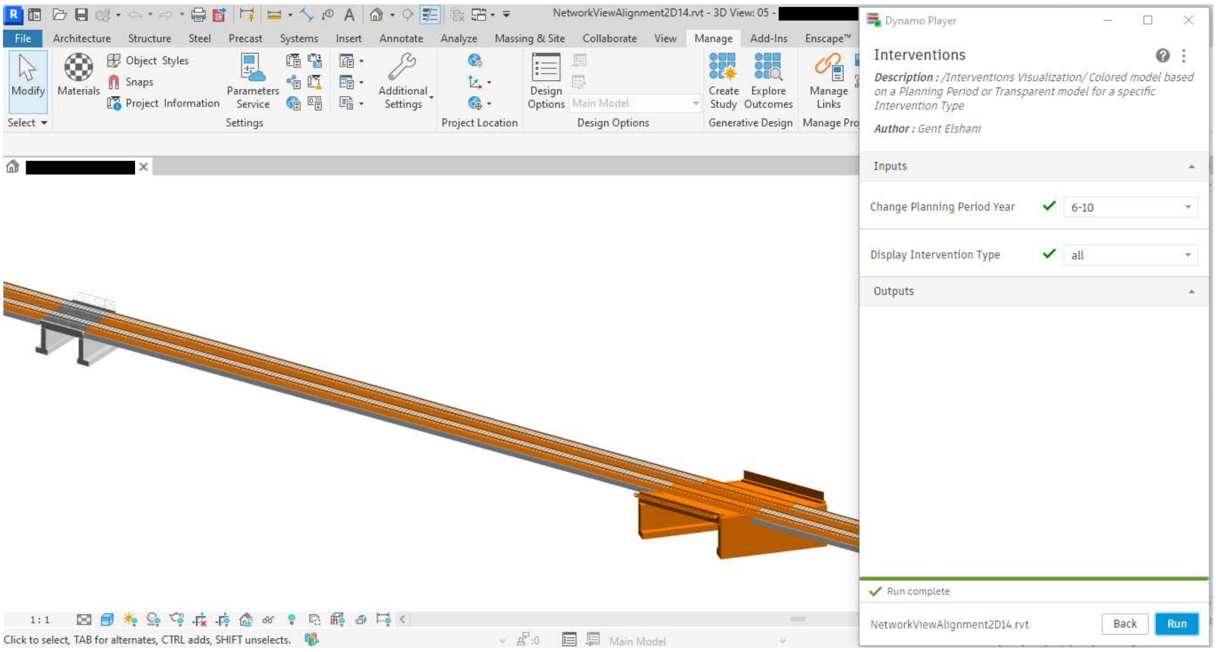
Outputs

✓ Run complete

NetworkViewAlignment2D14.rvt Back Run

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 99: The visualization through transparency of the intervention type and period at the network level.



NetworkViewAlignment2D14.rvt - 3D View: 05 - [redacted]

[redacted]

Interventions

Description : /Interventions Visualization/ Colored model based on a Planning Period or Transparent model for a specific Intervention Type

Author : Gent Elshani

Inputs

Change Planning Period Year ✓ 6-10

Display Intervention Type ✓ all

Outputs

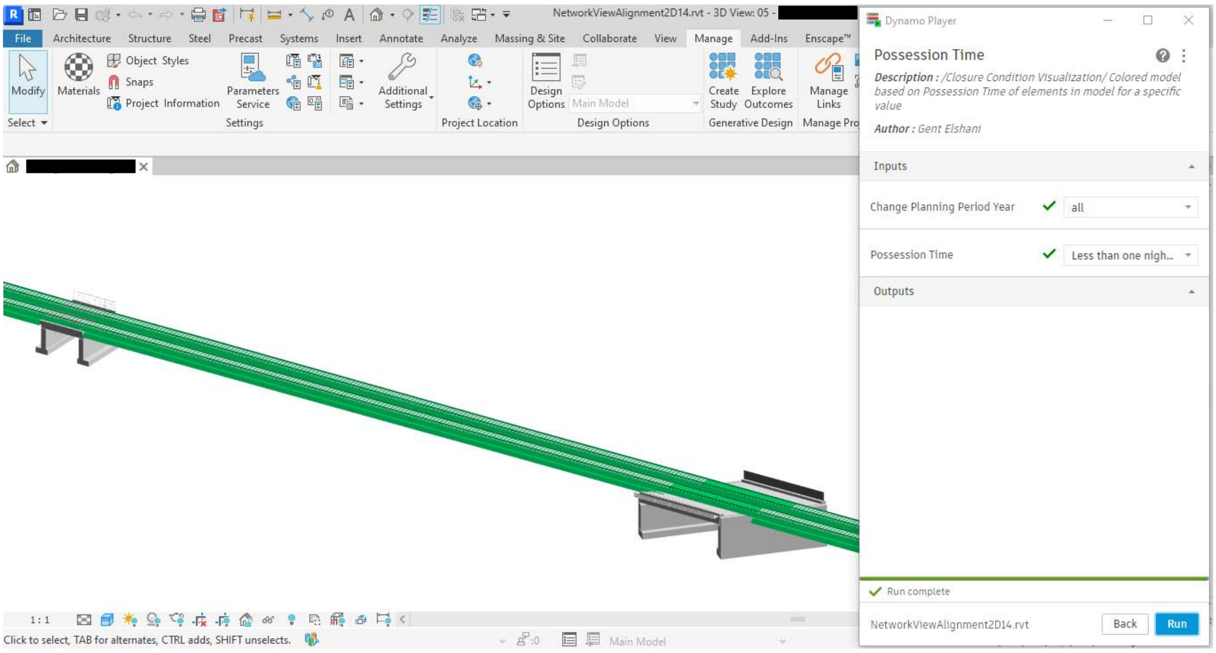
✓ Run complete

NetworkViewAlignment2D14.rvt Back Run

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 100: The visualization through color coding of the intervention planning period at the network level.

The following visualization in Figure 101 has resulted from applying the possession time script in the same section. This script is set up in a way that it compares the calculated possession time for carrying out given interventions on components with several threshold values. In this methodology, the included values of time frames greater than one night and one week, as well as values smaller than them have been incorporated. These intervals were selected to comply with the usual disruption time on service hours of the infrastructure involved with the interventions. In Figure 101, all components where the performance of the planned intervention lasts longer than one night (six hours) are highlighted in the provided color. In this specific case, only track components are shown since the interventions on the bridges would last significantly longer.



NetworkViewAlignment2D14.rvt - 3D View: 05 - [redacted]

[redacted]

Dynamo Player

Possession Time

Description : /Closure Condition Visualization/ Colored model based on Possession Time of elements in model for a specific value

Author : Gent Elshani

Inputs

Change Planning Period Year ✓ all

Possession Time ✓ Less than one nigh...

Outputs

✓ Run complete

NetworkViewAlignment2D14.rvt

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

Figure 101: The visualization through color coding of the intervention possession time at the network level.

The intervention cost has been included as a semantic information in the components, opening the possibility to further extend the script to visualize this information. Currently, because of the big data size mainly contributed by the large amount of sleeper and fastening system instances, the execution of the mentioned scripts on the network level takes more time than anticipated (about two minutes). Further research is necessary to improve this. One possibility includes the application of all view filters in Revit at once in the beginning, altering only the visibility of these filters through the Dynamo script. This approach could shorten the application time to an instant performance, resolving this limitation of the workflow.

3.4 Proposed workflow

The network level of the proposed methodology aims to also contribute at the asset level of the intervention planning process [2]. Emphasis is placed on fostering coordination between different asset managers, each responsible for their respective assets, considering the potential influence of specific interventions on the overall network's functionality. Figure 102 shows this coordination between the bridge and track asset managers as part of the intervention planning process, emphasizing the common decision-making that arises from their respective qualified activities. In this part, the network view can be proven useful by serving as a shared source of information.

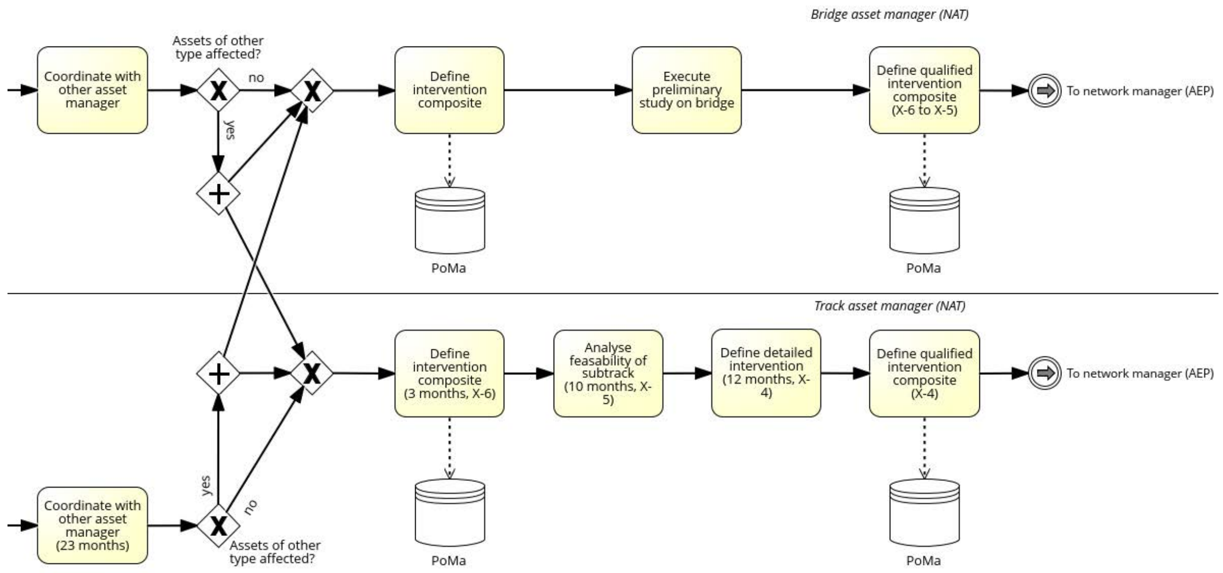

Bridge asset manager (NAT)
Coordinate with other asset manager
Assets of other type affected?
no
yes
Define intervention composite
Execute preliminary study on bridge
Define qualified intervention composite (X-6 to X-5)
To network manager (AEP)
PoMa
PoMa
Track asset manager (NAT)
Define intervention composite (3 months, X-6)
Analyse feasability of subtrack (10 months, X-5)
Define detailed intervention (12 months, X-4)
Define qualified intervention composite (X-4)
To network manager (AEP)
Coordinate with other asset manager (23 months)
yes
no
Assets of other type affected?
PoMa
PoMa

Figure 102: Processes involving more extensive coordination between asset managers, as well as collaboration with the network manager, as part of the intervention planning process [2].

Additionally, the coordination between the separate asset managers and the network manager can be enhanced using the network view, where initial ideas on ordering the interventions can be completed more efficiently. Patterns and common features can be identified using this view, providing further direct information about the network in a single perspective. Figure 103 displays the processes where the network manager is intended to be involved. All these processes can utilize the network view for securing further information without disregarding the usual data sources, which will provide a more comprehensive level of detail compared to the network view.

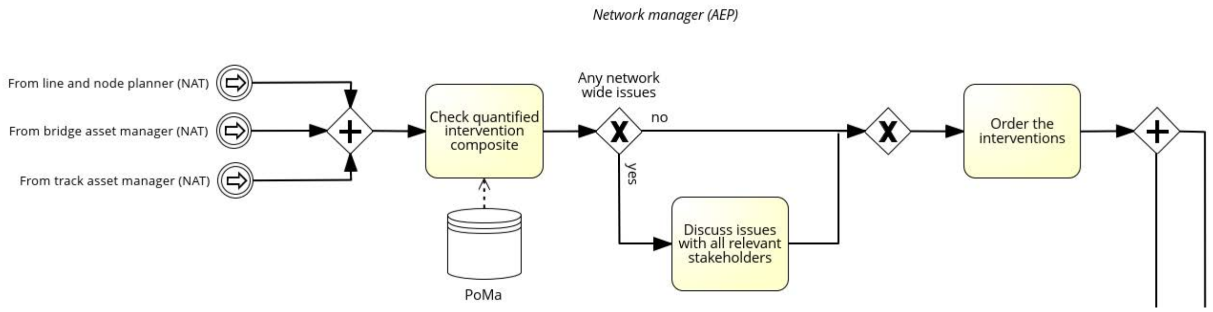

Network manager (AEP)
From line and node planner (NAT)
From bridge asset manager (NAT)
From track asset manager (NAT)
Check quantified intervention composite
PoMa
Any network wide issues
no
yes
Discuss issues with all relevant stakeholders
Order the interventions

Figure 103: Processes included in the domain of the network manager as part of the intervention planning process [2].

4 Construction planning

In this part of the Thesis, concepts on how the proposed methodology, the developed 3D models, and accompanying codes can be integrated into the construction planning phase, will be discussed. It includes the asset-level approaches presented in Section 2, as well as the network level shown in Section 3.

The benefit of integrating BIM into this phase of an asset's lifecycle results in an enrichment of the processes in multiple dimensions. By efficiently incorporating real-time information regarding the condition states and required interventions for railway assets, construction planners gain valuable insights that enhance the construction phase.

One of the most notable advantages is the early detection of issues. Potential problems can be identified before they escalate into critical issues through predictive maintenance. This approach minimizes construction delays and cost overruns, contributing to a more efficient project execution. Maintenance scheduling also receives a significant improvement by informing construction planners about the exact intervention needed on the railway assets. By combining predictive data with historical records, planners can optimize construction schedules, reduce downtime, and enhance the overall efficiency of railway infrastructure projects. Also, the need of having a 3D model to integrate BIM is not always necessary. Pure information models can advance the construction processes as well by informing managers on the specific needs through connected data. By setting the requirements in terms of information and semantics, they can advance the process by almost the same quality.

In term of models from various disciplines communicating with one other, researchers have also explored how to achieve the most efficient workflow. Figure 104 illustrates one potential strategy, where the use of an information bus as part of the CDE can facilitate updates with the separate instance models and necessary notifications [56]. The approach can advance the goal of reaching a BIM of Level 3 with integrated data spanning the whole life cycle.

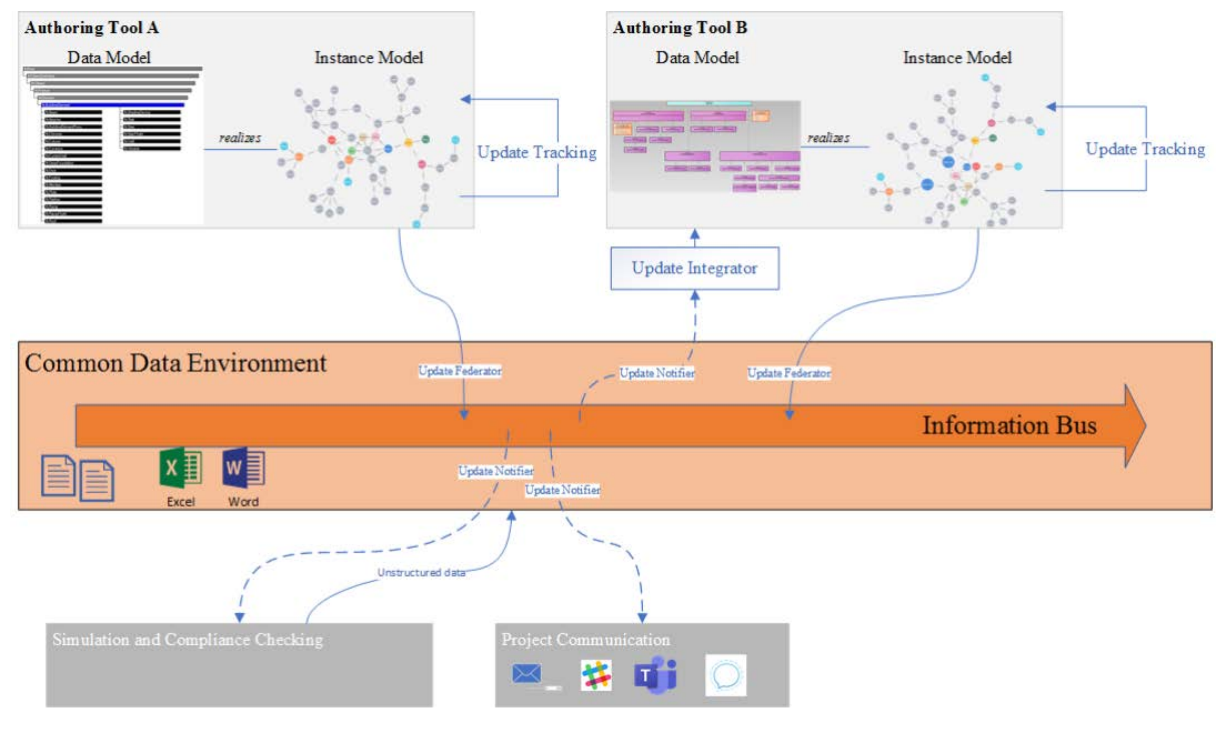


Figure 104: The proposed BIM Level 3 system architecture [57].

In order for the mentioned resources and methodologies to conform the specific needs that appear in construction planning phase, the following distinct measures can support the achievement of this goal.

By providing a further division of the elements, in comparison to the already established component-based approach, the specific perquisites can be fulfilled. The visualization through the Dynamo script is carried out by grouping all elements of the bridge according to which of the treated component they belong to. This approach allows for a representation of the properties of the components, however provides limited properties of the individual elements. Special attention could be given to the 'Component Name' parameter, which served as a common identifier for the components in this Thesis. By introducing individuality in the assignment and the identifiers, a much more accurate and meaningful outcome can be generated.

To expand the use cases of the current digital models, especially in terms of deriving construction drawings from the BIM models, it is necessary to introduce a finer granularity to the modeled assets. The proposed methodology did not necessitate a high level of geometric detail, as a greater emphasis has been placed on the information contained within the components, rendering a basic representation satisfactory. There are several ways of achieving more detailed models, where two of them might be applicable in the part of construction planning.

One solution to the encountered problem is to employ a central model through which the different disciplines and asset managers can communicate (see Figure 105). Since a lot of different stakeholders communicate and exert influence over the central model, it is necessary for each participant to establish requirements regarding the specific information that should be covered within the central model to address their domain. This holds particular importance in the context of the information models mentioned, as it enables a direct association between the output information and the model elements.

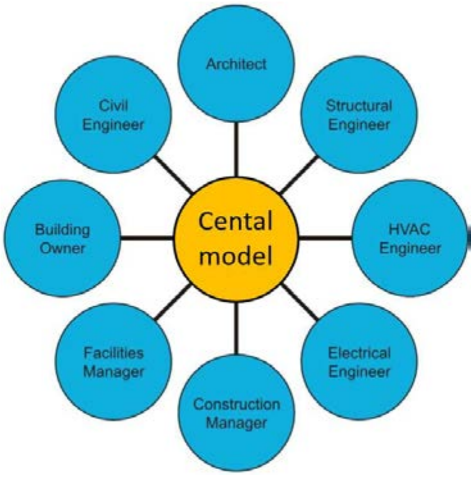


Figure 105: The central model between different disciplines, adapted from [36].

Another method involves refining the process of data exchange between different domains. Efficiency in addressing specific disciplines can be attained by employing multiple communicating models, ensuring that the necessary information is readily accessible to other fields at the same time. By introducing a standardized data exchange, this can be achieved (see Figure 106). Particular emphasis is placed on the utilization of open formats, such as the newly developed IFC4x3 format. These formats are globally understood and ensure a safe and reliable transfer of data between domains.

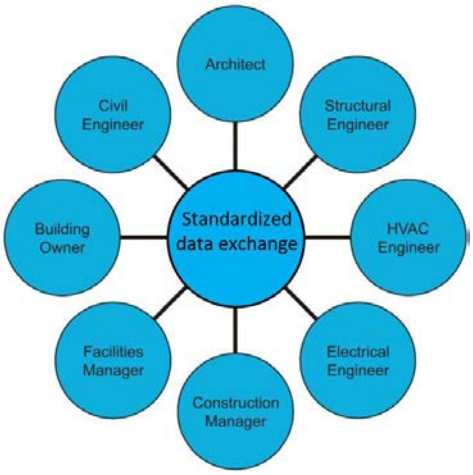


Figure 106: The standardized data exchange between different disciplines, adapted from [36].

Since the required information during the construction planning phase differs substantially from the previous phases, the workflow should be adapted. A stronger emphasis lies on concepts such as material flow, execution planning and phases, scheduling and cost, among others. This new set of data that the 3D models must contain might change the way data is being governed compared to previous approaches. Also, involvement of different stakeholders has to be foreseen and processes have to be modified accordingly. Since the execution work can be conducted both internally and through external companies, such as design offices or construction companies, an adapted procedure may need to be considered.

With concepts such as intervention schedules, construction sequencing, resource allocation, and cost estimations being introduced in this phase, an elevation of the dimensionality on the BIM development of the projects might be considered. In such ways, by reaching 5D BIM, key components of time and cost will be included, closely related to the mentioned concepts in the construction planning.

The integration of BIM workflows into the realm of 4D models has been the subject of extensive research over the past years. For instance, a study conducted at Stanford University involved a quantitative comparison between task execution using the conventional paper method and 4D BIM models [57]. The results indicated improved accuracy and task duration, confirming the practical benefit of these methods. Similar research methodologies for managing, visualizing, and delivering data with the utilization of 4D models for railway assets have also been investigated [29], [58].

4.1 Proposed workflow

The proposed approaches are also planned to facilitate the integration of digital models in the intervention planning process [2]. In this context, the production and project planning of this process could be significantly impacted. The primary focus would be on the earlier parts of this section, involving the corridor manager and project manager, and highlighting the early decision-making aspects of the intervention planning process (refer to Figure 107).

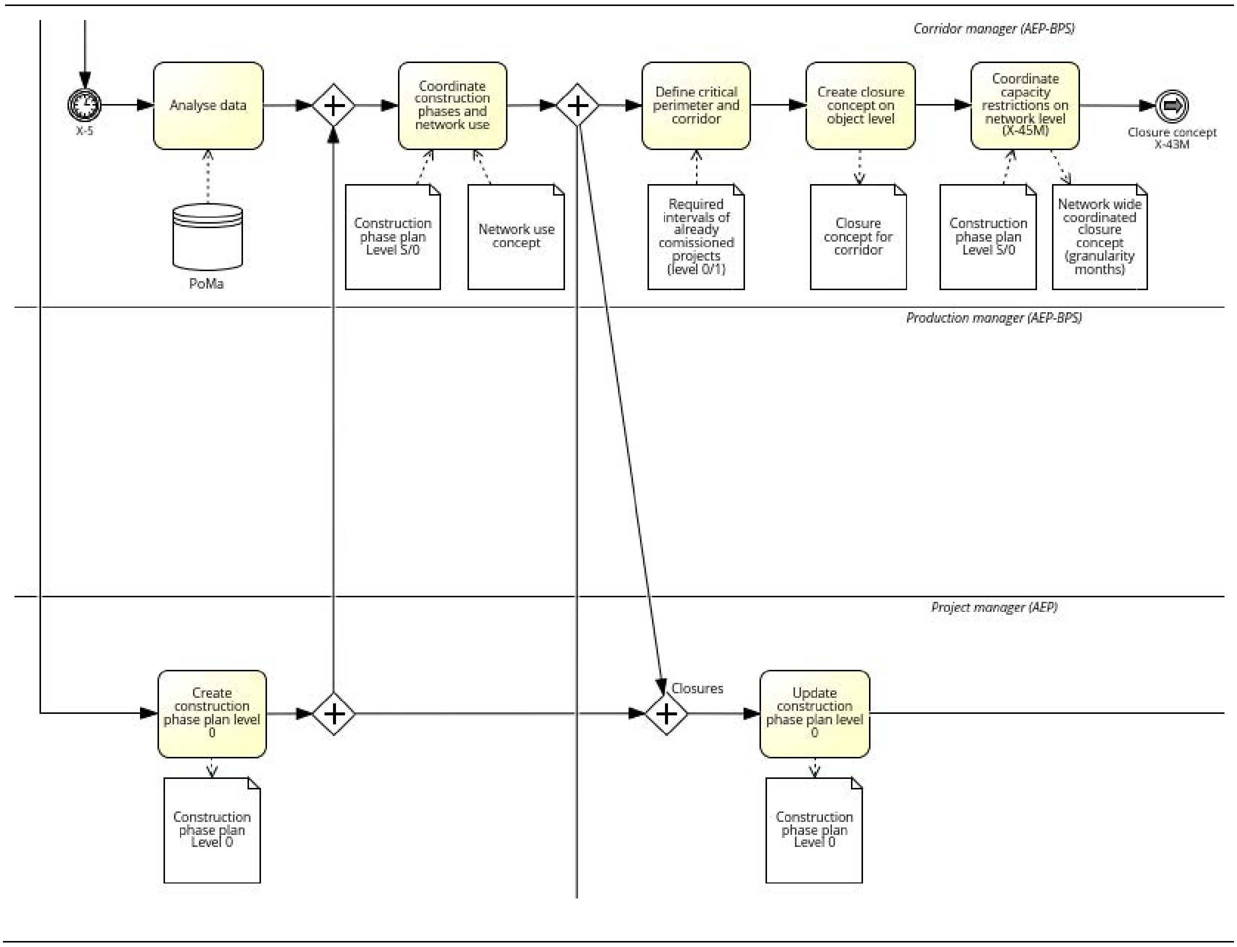

Corridor manager (AEP-BPS)
X-5
Analyse data
PoMa
Coordinate construction phases and network use
Construction phase plan Level 5/0
Network use concept
Define critical perimeter and corridor
Required intervals of already comissioned projects (level 0/1)
Create closure concept on object level
Closure concept for corridor
Coordinate capacity restrictions on network level (X-45M)
Construction phase plan Level 5/0
Network wide coordinated closure concept (granularity months)
Closure concept X-43M
Production manager (AEP-BPS)
Project manager (AEP)
Create construction phase plan level 0
Construction phase plan Level 0
Closures
Update construction phase plan level 0
Construction phase plan Level 0

Figure 107: Processes included in the early production and project planning as part of the intervention planning process [2].

5 Conclusion, limitations, and outlook of this Thesis

As a conclusion of this Thesis, the research gap presented through the objectives addressed in each major sector is aimed to be bridged through the conducted methodology and case study. Overall, it is evident that utilizing parametric elements can significantly speed up the modeling process, thereby making the existence and acceptance of BIM models more relevant. This effect is further enhanced through the employment of a library of standard parametric elements, which also provides a consistent foundation of digital modeling. By incorporating the Dynamo Player as a friendlier user-interface for the scripts, it becomes easier for users to access and harness the full power of the scripts. Finding the optimal way of handling data between different domains, such as the conversion between geodata and building data, is a critical part in implementing BIM effectively. The suggested network view opens up numerous possibilities by providing data coming from a single source and facilitating the coordination among project participants. Lastly, a whole life-cycle integration of the managed assets should be considered. By integrating processes ranging from early decision-making to design and ultimately to construction and maintenance, BIM's full potential can be realized.

The outlined objectives, specifically addressed within the scope of this Thesis, have been approached as follows:

Objective 1. Integrating digital data on an asset level

It has been shown that supplying asset managers with enhanced information and visual representations results in a deeper understanding of the current and future conditions of their assets, ultimately leading to improved decision-making processes. The concept of integrating visual information, in conjunction with the more precise data obtained from conventional sources, extends beyond bridge asset management and holds promise for serving a more holistic purpose across a diverse range of stakeholders.

Objective 2. Providing a unified data source on a network level

Demonstrations on employing an interface that enables various managers to collaborate, access, and share real-time information can significantly enhance communication across various sectors. This becomes particularly crucial during the planning and ordering interventions since it allows for a visual representation of how each asset impacts other areas or domains. This streamlines the clustering of interventions, minimizing potential downtime during operation. The possibility to include different visualization modules, such as the possession time of how long a component is unavailable during the interventions, opens up new possibilities of how data can be seen in the future.

Objective 3. Enhancing the usage of digital models in subsequent phases

The discussion emphasizes that there exist substantial advantages in using a shared model or a common data environment for multiple purposes, thereby enhancing the usefulness of BIM models. Different approaches can be pursued while considering data exchange and potential data loss. Finding an optimal digital workflow that aligns with the established processes of infrastructure managers can offer significant advantages in terms of cost optimization, scheduling efficiency, and resource management for projects.

Limitations

The main limitations of this work are in the aspect of unknown or uncertain data. In the aspect of the unique ID that has been used as a link between the model elements and the values in Excel sheets, a combination of the bridge name and the component has been employed. It is highly probable that this is not the case in the databases containing the assets of the SBB, and thus, adjustments are necessary. Nevertheless, by providing the methodology as a foundation, the same principles can be followed with modifications in this regard.

Moreover, by using a common BIM authoring software like Revit for including the railway assets, bigger data sizes than desired have resulted. This is especially attributed to the substantial number of track components, particularly the sleepers and fastening systems. This fact has added some complexity to both the model and the overall workflow, which should be taken into account when reviewing this methodology.

Furthermore, several adaptations were necessary to convert the available railway track data at the network level from Shapefile into a usable format within the workflow. Depending on the accessible data, an alternative methodology with a more straightforward data exchange can be proposed, thereby reducing the necessary steps to fulfill the goals of this methodology.

Outlook

Future outlook and perspectives for the workflow proposed in this Master's Thesis, as part of the STABILITY project, encompass several aspects of the methodology. Their refinement can potentially lead to even more meaningful outcomes, making a significant contribution to the domain of digitalization and automation in the construction industry.

Optimizing the mentioned data size of the network view can improve the usability of this single source of truth throughout the life span of railway assets.

Intervention cost information has been incorporated at the network level. This data can be further visualized to gain insights into the aspect of resource planning. Similarly, different types of data can be additionally included to create meaningful and valuable visualizations for the asset managers.

Finally, considering the internal strategies of the SBB and the focus on the planned approach for common data environments, which may involve establishing a centralized model with predefined requirements or utilizing multiple interconnected models, the proposed methodology can be implemented accordingly.

References

[1] “Maintaining train schedule (STABILITY) and managing time table reserves via digitalized railway intervention planning,” im.ibi.ethz.ch. https://im.ibi.ethz.ch/en/research/research-and-thesis-projects1/data-_driven-intelligent-predictive-maintenance-of-industrial-as.html (accessed Sep. 29, 2023).

[2] S. Chuo, M. Burkhalter, B.T. Adey, “Current SBB intervention planning process in view of the STABILITY project,” 2022.

[3] “Bridges,” data.sbb.ch. https://data.sbb.ch/explore/dataset/brucken/information (accessed Sep. 29, 2023).

[4] B. Abbasnejad and H. I. Moud, “BIM and basic challenges associated with its definitions, interpretations and expectations,” *International Journal of Engineering Research and Applications (IJERA)* ISSN: 2248-9622 www.ijera.com Vol. **3**, Issue 2, March-April 2013, Jan. 2012, pp.287-294, ISSN: 2248-9622 www.ijera.com

[5] “About AIA document reference material,” Available: https://content.aia.org/sites/default/files/2017-02/2008%20Docs%20Advance%20the%20Use%20of%20BIM%20and%20Integrated%20Projecct%20Delivery.pdf

[6] J. John, “Opportunities and current limitations of cloud-based design automation in the context of Building Information Modelling,” Master Thesis, Technical University of Munich, 2021.

[7] V. Appelt, "Einführung BIM I," [Lecture], BIM in der Verkehrsinfrastruktur IIM – Infrastructure Information Modelling, Apr. 20, 2023.

[8] “Building Information Modelling – BIM | SBB,” company.sbb.ch. https://company.sbb.ch/de/ueber-die-sbb/projekte/nationale-programme/bim.html (accessed Sep. 29, 2023).

[9] “Beauftragt durch das,” Available: https://www.digitalzentrumbau.de/kos/WNetz?art=File.download&id=2555&name=BIM4INFRA2020_AP4_Teil1.pdf

[10] “ARGE BIM4RAIL Endbericht zur wissenschaftlichen Begleitung von 13 Pilotprojekten zur Anwendung von BIM im Schienenwegebau,” 2019. Available: https://bim4infra.de/wp-content/uploads/2021/03/BIM4RAIL_AP1_Endbericht.pdf (accessed Sep. 29, 2023)

[11] “Use of BIM in infrastructure projects by construction professionals globally 2017,” *Statista*. https://www.statista.com/statistics/892338/bim-in-infrastructure-projects-according-to-construction-professionals-globally/

[12] N. Bui et al, “An institutional perspective on BIM implementation – a case study of an intercity railway project in Norway,” *Designing Digitalization* (ISD2018 Proceedings), Mar. 2018, https://aisel.aisnet.org/isd2014/proceedings2018/General/2/.

[13] K. Lysebo, “BIM success for InterCity railway setting the standard for model-based delivery,” Available: https://static.au-uw2-prd.autodesk.com/handout_21354_TR21354_20Lysebo_20AU2016.pdf (accessed Sep. 29, 2023)

[14] A. A. Enshassi, L.A.A. Hamra, S. Alkilani, “Studying the benefits of Building Information Modeling (BIM) in architecture, Engineering and Construction (AEC) Industry in the Gaza Strip,” *Jordan J. Civ. Eng*, Jan. 2018.

[15] A. A. Latiffi, S. Mohd, and U. S. Rakiman, “Potential improvement of building information modeling (BIM) implementation in malaysian construction projects,” in IFIP International Conference on Product Lifecycle Management, 2015, 149–158.

[16] C. B. Farnsworth, S. Beveridge, K. R. Miller, and J. P. Christofferson, “Application, advantages, and methods associated with using BIM in commercial construction,” *International Journal of Construction Education and Research*, vol. **11**, no. 3, pp. 218–236, Mar. 2014, doi: https://doi.org/10.1080/15578771.2013.865683.

[17] M. Bensalah, A. Elouadi, and H. Mharzi, “BIM integration into railway projects - case study,” *Contemporary Engineering Sciences*, vol. **11**, no. 44, pp. 2181–2199, 2018, doi: https://doi.org/10.12988/ces.2018.85217.

[18] A. Sanchez, S. Mohamed, K. Hampson, "Delivering value with BIM: a framework for built environment practitioners," In Proceedings of the CIB World Building Congress 2016: Volume **III**-Building Up Business Operations and Their Logic. Shaping Materials and Technologies, 2016, pp. 272-283.

[19] S. Esser and K. Aicher, “IfcBridge model generation using visual programming,” in *Forum Bauinformatik*, 2019, pp. 101–108.

[20] M. Bensalah, A. Elouadi, and H. Mharzi, “Overview: the opportunity of BIM in railway,” *Smart and Sustainable Built Environment*, vol. **8**, no. 2, pp. 103–116, May 2019, doi: https://doi.org/10.1108/sasbe-11-2017-0060.

[21] A. Ciccone, S. Stasio, D. Asprone, A. Salzano, and M. Nicolella, “Application of openBIM for the management of existing railway infrastructure: case study of the Cancello–Benevento railway line,” *Sustainability*, vol. **14**, no. 4, p. 2283, Feb. 2022, doi: https://doi.org/10.3390/su14042283.

[22] Š. Jaud, H. Hecht, J. Schlenger, and J. Amann, “TUM Open Infra Platform: an open source package for simultaneous viewing and analysis of digital models in the civil engineering domain,” *Journal of Open Source Software*, vol. **7**, no. 72, p. 3061, Apr. 2022, doi: https://doi.org/10.21105/joss.03061.

[23] M. Garramone, E. Tonelli, and M. Scaioni, “A multi-scale BIM/GIS framework for railways asset management,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. **XLVI-5/W12022**, pp. 95–102, Feb. 2022, doi: https://doi.org/10.5194/isprs-archives-xlvi-5-w1-2022-95-2022.

[24] C.-S. Shim, L. Kwang-Myong, W.-S. Son, and J.-W. Moon, “Collaborative design of high-speed railway lines using 3D information models,” Jan. 2008, doi: https://doi.org/10.2749/222137808796106323.

[25] J. Wei et al., “BIM and GIS applications in bridge projects: a critical review,” *Applied Sciences*, vol. **11**, no. 13, p. 6207, Jul. 2021, doi: https://doi.org/10.3390/app11136207.

[26] M. Reinhardt, D. Gebauer, and S. Marx, “Anforderungen und Konzept für eine automatisierte Zustandserfassung von Bauwerken mittels Bilderfassung und -auswertung,” *Bautechnik*, vol. **95**, no. 10, pp. 705–711, Sep. 2018, doi: https://doi.org/10.1002/bate.201800065.

[27] R. S. Panah and M. Kioumarsi, “Application of Building Information Modelling (BIM) in the health monitoring and maintenance process: a systematic review,” *Sensors*, vol. **21**, no. 3, p. 837, Jan. 2021, doi: https://doi.org/10.3390/s21030837.

[28] R. Hartung, H. Naraniecki, K. Klemt-Albert, and S. Marx, “Konzept zur BIM-basierten Instandhaltung von Ingenieurbauwerken mit Monitoringsystemen,” *Bautechnik*, vol. **97**, no. 12, pp. 826–835, Nov. 2020, doi: https://doi.org/10.1002/bate.202000095.

[29] J. Neves, Z. Sampaio, and M. Vilela, “A case study of BIM implementation in rail track rehabilitation,” *Infrastructures*, vol. **4**, no. 1, p. 8, Feb. 2019, doi: https://doi.org/10.3390/infrastructures4010008.

[30] A. Stolz, “Connecting future predictions to BIM,” Master Thesis, Technical University of Munich, 2023.

[31] F. Tehlar and K. Fraberger, “Connecting future predictions on road bridges to BIM,” Master Project, Swiss Federal Institute of Technology, 2023.

[32] G. Roelfstra et al., “Condition evolution in bridge management systems and corrosion-induced deterioration,” *Journal of Bridge Engineering*, vol. **9**, no. 3, pp. 268–277, 2004.

[33] B. Adey, "Changes," [Lecture], Infrastructure Management 1: Process, May. 25, 2022.

[34] A. Borrmann, M. König, C. Koch, and J. Beetz, Eds., *Building Information Modeling*, Cham: Springer International Publishing, 2018. doi: https://doi.org/10.1007/978-3-319-92862-3.

[35] “BIM Fachdatenkatalog,” fdk.app.sbb.ch. https://fdk.app.sbb.ch/en/objects (accessed Sep. 29, 2023).

[36] A. Borrmann, "Parametric Modeling," [Lecture], BIM Fundamentals, Oct. 10, 2022.

[37] A. Borrmann, Y. Ji and J. R. Jubierre, “Multi-scale geometry in civil engineering models: consistency preservation through procedural representations,” in Proc. of the 14th Int. Conf. on Computing in Civil and Building Engineering, Moscow, Russia, 2012.

[38] “An Introduction to Grasshopper – Brendan Harmon,” baharmon.github.io. https://baharmon.github.io/basics

[39] “Level of development specification – previous issue – BIM forum,” bimforum.org. https://bimforum.org/resource/level-of-development-specification-previous-issue/ (accessed Sep. 29, 2023).

[40] “VDI 2552,” VDI. https://www.vdi.de/richtlinien/unsere-richtlinien-highlights/vdi-2552 (accessed Sep. 29, 2023).

[41] S. Oelfe, “Parametrische Konstruktion von Systemparkhäusern mit Hilfe von Grasshopper 3D und Tekla Structures,” Bachelor Thesis, Technical University of Munich, 2018.

[42] “IFC4.3.2.0 Documentation,” ifc43-docs.standards.buildingsmart.org. https://ifc43-docs.standards.buildingsmart.org/ (accessed Sep. 29, 2023).

[43] “Selecting | The Dynamo Primer,” primer.dynamobim.org. https://primer.dynamobim.org/08_Dynamo-for-Revit/8-2_Selecting.html (accessed Sep. 29, 2023).

[44] R. Albrecht, H. Hagen, G. Farin, and H. Noltemeier, *Geometric Modelling*. Springer Science & Business Media, 2012.

[45] S. Kollmannsberger, “The finite cell method: towards engineering applications,” Cumulative habilitation thesis, Technical University of Munich, 2019.

[46] C. Preidel, S. Daum, and A. Borrmann, “Data retrieval from building information models based on visual programming,” *Visualization in Engineering*, vol. **5**, no. 1, Oct. 2017, doi: https://doi.org/10.1186/s40327-017-0055-0.

[47] S. Rulovs, “Implementation of an IFC file creator and modifier using visual programming,” Bachelor Thesis, Technical University of Munich, 2022.

[48] J. Collao, F. Lozano-Galant, J. A. Lozano-Galant, and J. Turmo, “BIM visual programming tools applications in infrastructure projects: a state-of-the-art review,” *Applied Sciences*, vol. **11**, no. 18, p. 8343, Jan. 2021, doi: https://doi.org/10.3390/app11188343.

[49] A. Omara, “Towards parametric design: automation of complex geometrical modelling,” Master Thesis, Technische Universität Dresden, 2019.

[50] “IFC4 Documentation,” standards.buildingsmart.org. https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/

[51] Š. Markič, A. Donaubauer and A. Borrmann, “Enabling geodetic coordinate reference systems in building information modeling for infrastructure,” In Proceeding of the 17th

International Conference on Computing in Civil and Building Engineering, Tampere, Finland (pp. 5-7), Jun. 2018.

[52] “Line (graphical),” data.sbb.ch. https://data.sbb.ch/explore/dataset/linie-mit-polygon/information (accessed Sep. 29, 2023).

[53] “swisstopo - wissen wohin,” Bundesamt für Landestopografie swisstopo. https://www.swisstopo.admin.ch/de/home.html (accessed Sep. 29, 2023).

[54] “Geoprocessing considerations for shapefile output—ArcMap | Documentation,” desktop.arcgis.com. https://desktop.arcgis.com/en/arcmap/latest/manage-data/shapefiles/geoprocessing-considerations-for-shapefile-output.htm#:~:text=Because%20of%20the%20simplicity%20of (accessed Sep. 29, 2023).

[55] “3DCityDB Database – Homepage,” https://www.3dcitydb.org/3dcitydb/

[56] S. Esser and A. Borrmann, “A system architecture ensuring consistency among distributed, heterogeneous information models for civil infrastructure projects,” Proc. Of the 13th European Conference on Product and Process Modeling. Moscow, 2020/2021, May 2021.

[57] Z. Zhang, H. Hamledari, S. Billington, M. Fischer, “4D beyond construction: spatio-temporal and life-cyclic modeling and visualization of infrastructure data,” ITcon 23, 2018, pp 285-304. https://www.itcon.org/paper/2018/14

[58] S. Kaewunruen, J. Sresakoolchai, and Y. Lin, “Digital twins for managing railway maintenance and resilience,” *Open Research Europe*, vol. **1**, p. 91, Nov. 2021, doi: https://doi.org/10.12688/openreseurope.13806.2.

Appendix A – Dynamo codes

The developed Dynamo codes used throughout this Master's Thesis can be found in the depictions below. While Figure A1 illustrates the script used to visualize the condition states and interventions, Figure A2 shows the script that has been utilized to generate the railway track components.

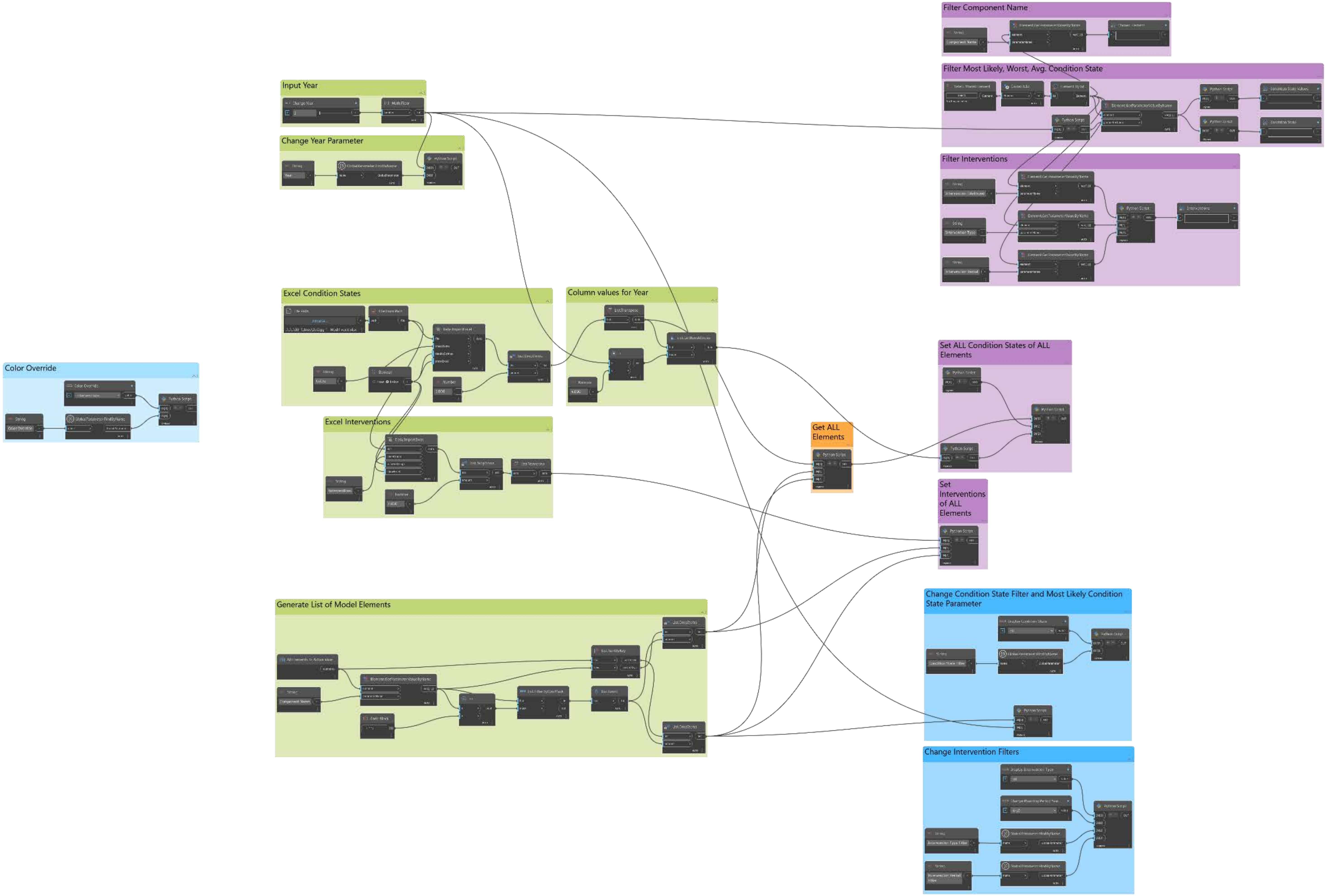

Filter Component Name
Filter Most Likely, Worst, Avg. Condition State
Input Year
Change Year Parameter
Filter Interventions
Excel Condition States
Column values for Year
Set ALL Condition States of ALL Elements
Color Override
Get ALL Elements
Excel Interventions
Set Interventions of ALL Elements
Change Condition State Filter and Most Likely Condition State Parameter
Generate List of Model Elements
Change Intervention Filters

Figure A1: The optimized Dynamo code used to visualize condition states and interventions.

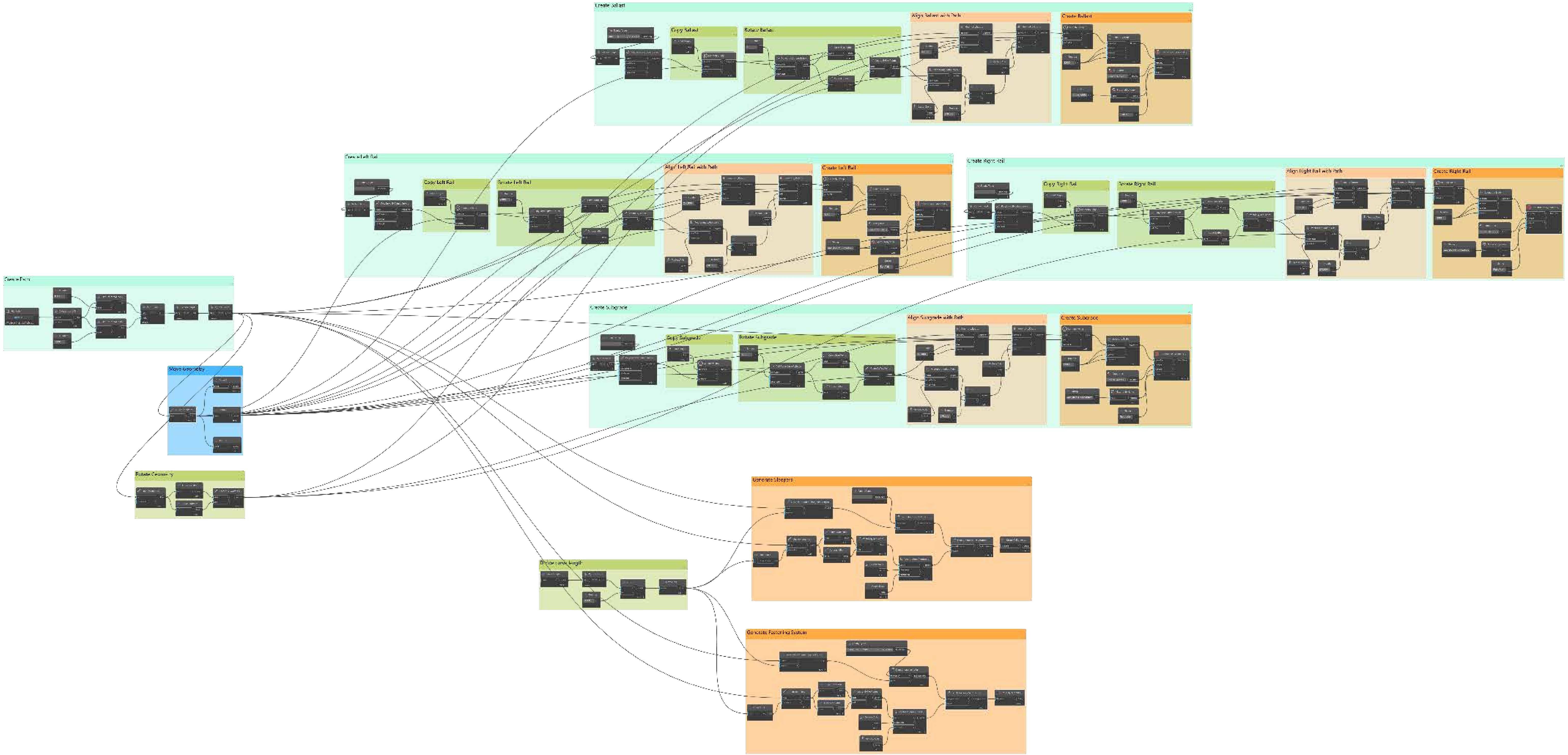


Figure A2: The Dynamo code used to generate the railway track assets.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

CONNECTING FUTURE PREDICTIONS OF RAILWAY ASSETS TO BIM

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
ELSHANI	GENT

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date **Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

ETH Zürich
Institut für Bau- und Infrastrukturmanagement
Stefano-Franscini-Platz 5
8093 Zürich

www.ibi.ethz.ch