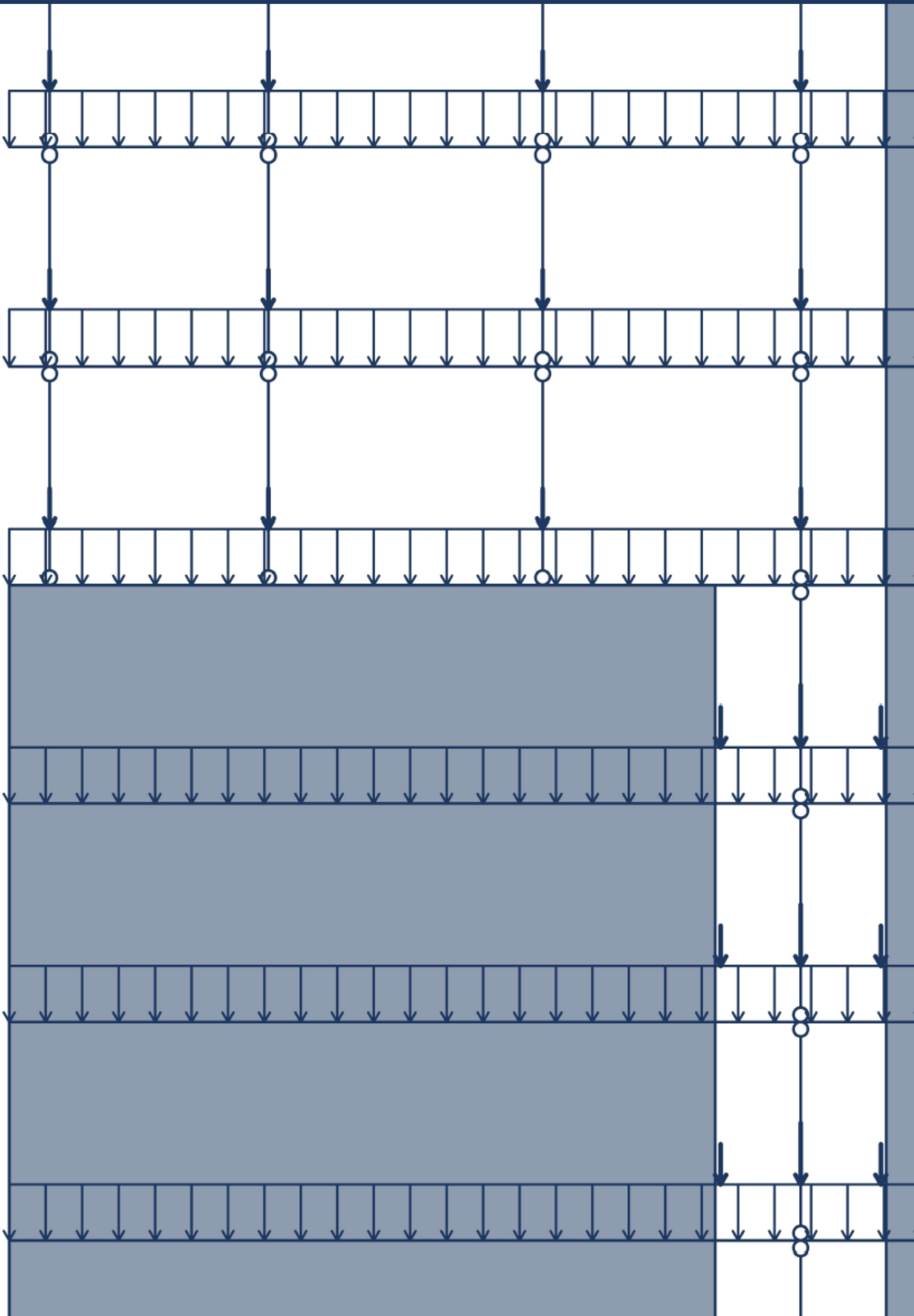
Towards BIM-based automation of the vertical load calculation

R. Willemsen



TU Delft

CAE

ONDERDEEL VAN IV-BOUW

Towards BIM-based automation of the vertical load calculation

by

R.Willemsen

in partial fulfilment of the requirements for the degree of

Master of Science
in Civil Engineering: Building Engineering - Structural Design

at the Delft University of Technology,
to be defended publicly on Friday 28 August, 2020 at 11:00 AM.

Thesis committee:	Dr. Ir. H.R. Schipper,	TU Delft
	Ir. S.T. Kiefte,	CAE Nederland B.V.
	Ir. S. Pasterkamp,	TU Delft
	Dr. Ir. G.A. Van Nederveen,	TU Delft

Preface

This thesis is written to complete my Master Building Engineering at the Delft University of Technology.

Through this way I would like to thank my graduation committee members. First of all I would like to thank Simon Kiefte, my daily supervisor at CAE Nederland, for his guidance and help during the whole process. Furthermore, I want to thank the other members of my committee Roel Schipper, Sander Pasterkamp, and Sander van Nederveen for their feedback during our meetings.

I would also like to thank my colleagues at CAE for their enthusiasm when discussing the topic of my thesis and for helping me with their experience and knowledge.

I also want to thank the people who participated in the interviews and user tests I performed during my research.

Finally, I would like to thank my friends and family for their support and encouragement during my thesis.

R. Willemsen
Delft, August 2020

Summary

The work processes of structural engineers have been changing over the last decades. Digitalisation and automation play an important role in this change. One of the major developments in this context has been the introduction of Building Information Modelling (BIM). In a BIM model the geometry of the actual building is represented. To this geometry additional information regarding the design, construction, and maintenance of the building is attached. The combination of 3D geometry and meaningful data make a BIM model a valuable means for collaboration, both within companies and between different companies. In current engineering practice, BIM software is used for drawing purposes only. This is unfortunate, as a lot of the information present in the model can be used for calculation purposes as well. At the same time, structural calculations still contain manual and time-consuming tasks. In this thesis it is therefore investigated how the information in the BIM model can be used for calculation purposes.

In this thesis the focus is specifically on the vertical load calculation (Dutch: Gewichtsberekening). This is an important calculation in the design of a building, as it is used to determine how the total building weight is distributed to and carried by the foundation. In this process the structural engineer defines and documents how the forces acting on the building are transferred through the load-bearing structure by means of schematisations. Now, this process is often carried out using Excel and structural analysis software, involving a lot of manual work.

The objective of this thesis is to research and develop a method and tool that can be used to automate the manual processes in the vertical load calculation based upon a BIM model. The focus is specifically on multi-storey residential buildings. These are typically mid-rise buildings with a load-bearing structure made out of concrete. To achieve the objective, a framework is developed which the different steps of the vertical load calculation are described using a number of flowcharts. For each of the steps it is indicated if they can be automated. Furthermore, responsibilities are assigned to the different persons involved.

As a basis for the development of the framework, the state-of-the-art in research and practice of the vertical load calculation and its automation has been investigated. Semi-structured interviews were performed to identify how the vertical load calculation is currently organised and to what extend automation is utilised in engineering firms active in the Dutch building industry. It followed that the approach to the vertical load calculation is comparable between companies, the main difference being the way in which buildings are separated into parts that are considered separately: either horizontally or vertically. The current level of automation varies between companies, however all participants noted the digital transformation of the building industry. A literature review has been conducted to obtain an overview of existing BIM-based automation tools and to identify what software can be used to automate the vertical load calculation. The amount of literature on the topic is limited and the focus is mainly on the conceptual design or the construction phase, while most of the work on the vertical load calculation is done during the developed design phase. The literature review also indicated that structural calculations are often automated using object-oriented programming languages to connect BIM and structural analysis software, or using parametric software.

The outcomes from the interviews and the literature review have been analysed to come up with a list of requirements for the development of the method and tool. The requirements relate to the definition of the method, the possibility for user interaction and the usability of the tool. Based on these a concept framework was developed in which the general idea behind the method is explained. Next, the scope of the framework was narrowed to be able to define the process more clearly and to allow for the allocation of responsibilities. The scope was limited to the approach to the vertical load calculation in which the building is vertically divided into separately considered parts. Also, the developed framework specifically focusses on the use of Revit as BIM software. To clearly define possible use cases of the framework lists of the loads, elements, and possible adjustments the structural engineer can apply are included.

The practical applicability of the developed method is tested using Dynamo. A parametric engineering software connected to Revit, using which a tool has been developed as proof of concept. Testing demonstrated that the tool could be successfully applied to simple buildings. A test with a large, more complex BIM model showed the limitations of the prototype tool, as the execution slowed or even failed for certain parts of the process. User tests were successfully performed. Participants especially valued the visualisation of design choices and the possibility to make adjustments to the automatically generated output. Limitations identified were the required knowledge of Revit and Dynamo and the limited functionalities of the current connection to structural analysis software. This are considerations that need to be taken into account for further development of the tool.

It is concluded that the vertical load calculation can be automated based on a BIM model. It is important to clearly specify the different steps of the process and to provide the structural engineer with the means to interfere with the outcomes of the calculation. In this thesis the potential of BIM-based automation of structural calculations is demonstrated. Using the developed tool, schematisations could be quickly generated and directly calculated using the connected structural analysis software. By supporting the structural engineer in documenting design choices early, the introduction of errors and misunderstandings is prevented. In the end this results in time-saving, time that can be spend on the more detailed and complex parts of calculation processes.

List of Figures

List of Tables

Contents

Part 1: Problem definition

1

Introduction

1.1. Digital transformation of the structural engineering practice

The last decades the way the work processes of structural engineers are organised have been changing. Traditional drawing and calculation tools have been exchanged for Computer-Aided Drafting (CAD) and Computer-Aided Engineering (CAE) tools. The next step in the digital transformation of the structural engineering practice is Building Information Modelling (BIM) (Czmoch & Pekala, 2014).

BIM was first introduced in the mid-2000s and currently BIM methods are used in daily engineering practice in America, large parts of Europe as well as in Singapore and South Korea (Borrmann et al., 2018). The adoption of BIM in practice has been slow, this can be explained by the fact that the building sector is traditionally reluctant to change. This can be explained by the large number of companies that are involved in a building project and consequently are all dependent on each other (Agarwal et al., 2016).

Despite these developments, major issues like tedious or time-consuming (manual) tasks as well as communication gaps, still exist (Chi et al., 2015; Czmoch & Pekala, 2014; Hyun et al., 2018; Mangal & Cheng, 2018). Therefore, engineering firms are altering their practices using parametric, algorithms, BIM, design computation techniques, and scripting tools that help to automate parts of their design and improve cooperation (Andia & Spiegelhalter, 2014).

1.2. Design automation

Design automation is not new: Since the early development of computer-aided modelling and simulation, engineers have been automating chains of engineering activities (Dixon, 1995). Using design automation development processes can be made less time-demanding and more organised (Sandberg et al., 2016). Automation helps reduce the introduction of errors, as the manual re-entering of information from different software packages is no longer needed (Borrmann et al., 2018).

Design automation requires a large initial investment of time and money. Therefore, there should only be opted for design automation if processes are repetitive and time demanding (Van der Linden, 2018). Another important limitation one needs to consider before one starts to automate, is that it is impossible to automate every step of a design or calculation process. Automation is limited to the steps of the process that can be represented in the form of clear rules of logic. (Van der Linden, 2018).

1.3. Definition of BIM

Automation already occurs to a certain extent in the current structural engineering practice through the use of BIM technologies. There exist many different definitions of BIM. The National BIM Standard of the United States defines BIM as: *"A digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition''* (NIBS, 2019).

In the UK, BIM is defined by the Construction Project Information Committee as a *"...digital representation of physical and functional characteristics of a facility creating a shared knowledge resource for information about it forming a reliable basis for decisions during its life cycle, from earliest conception to demolition"* (RIBA, 2012, pp. 3).

In the Netherlands, the BIM Loket defines BIM as a digital representation of the functional and physical properties of a structure. It is a virtual version of a building or infrastructure, or the process of achieving that. The core of BIM is a 3D model, the geometry. To this, all the information that is needed to design, construct, and maintain the structure, is attached. In essence, BIM is a way of collaborating between parties, during the whole life cycle of a structure (BIM Loket, 2019b).

To clarify how BIM is viewed in this thesis, in the following, a definition of BIM will be given that will be used throughout this thesis. The author agrees with the BIM Loket: 3D geometry is the core of BIM, because all other information in the model relates to the geometry. It is relevant for all users, and therefore a valuable means for collaboration between different users.

To give the geometry more meaning, additional information is attached to the geometry in the form of parameters. Most BIM software also provide an analytical model to the 3D geometry. This is a wireframe representation of the physical model (Hasan et al., 2019), that can be exported to structural analysis software to be used to perform structural analysis.

The additional information might be discipline-specific and therefore not relevant for all users of the model. The way it is incorporated in the model, distinguishes BIM from a regular 3D CAD model.
In most literature this additional information is referred to as 'semantic information' (Borrmann et al., 2018; Eastman et al., 2011), in other literature the term 'metadata' is used (Bogani et al., 2008; Steel et al., 2012). For the definition of BIM in this thesis, the term 'metadata' is preferred, as its definition literally describes what is meant by it: Metadata is "information that is given to describe or help you use other information" (Cambridge Dictionary, 2020).

From the point of view of this thesis, the analytical model is considered an important part of the BIM model. Therefore, it is defined as a special kind of metadata. On the one hand the analytical model gives information about the 3D geometry, defining where centre lines are situated. While on the other hand, parameters like dimensions, positions, and end-conditions, describe the analytical model itself. In Figure 1.1 an example is given of how this definition applies to a concrete column modelled in Revit.

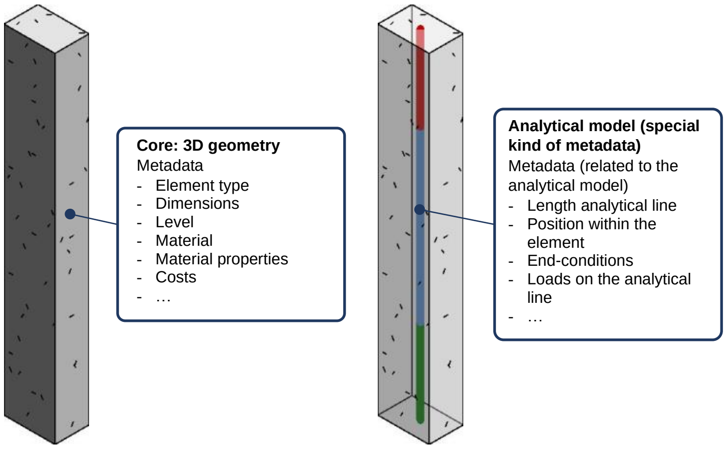

Core: 3D geometry
Metadata
- Element type
- Dimensions
- Level
- Material
- Material properties
- Costs
- …
Analytical model (special kind of metadata)
Metadata (related to the analytical model)
- Length analytical line
- Position within the element
- End-conditions
- Loads on the analytical line
- …

Figure 1.1: Definition of Building Information Modelling used in this thesis

1.4. Extending the current use of BIM

Currently, BIM is mostly used as 3D modelling technique (Arayici et al., 2018). The time saved compared to CAD- and CAE-systems is substantial. The productivity gain is estimated on 15% to 41% (Sacks & Barak, 2008). Therefore, it can be stated that BIM software has proven itself for drawing purposes.

A lot of information needed for structural calculations is already included in the BIM models currently used for drawing purposes only. Therefore, the use of BIM could be extended to calculation purposes as well. This information can then directly be extracted from the model requiring no manual re-entering. Furthermore, the visualisation ability of BIM plays an important role. Visualisation is an important part of structural calculations, since it improves understanding and collaboration, and BIM can help to provide this.

2
Research structure

In the previous chapter the subject of this MSc thesis was introduced. Based on this a research objective and research questions are derived.

2.1. Research objective

Although the current use of BIM is primarily focused on drawing purposes (Arayici et al., 2018), it can be considered a suitable means for automation of structural calculations for the reasons mentioned in Subsection 1.4. Therefore, this MSc thesis aims to investigate if BIM can be used for the automation of structural calculations and how this can best be done.

This means it will be investigated if the information in the BIM model can be used directly for calculation purposes or if additional information and/or processing of the information in the model is needed. Furthermore, it will be investigated how the information can be used and if the outcomes of the different process steps can be incorporated into the BIM model.

The scope of this MSc thesis is limited to the vertical load calculation (Dutch: Gewichtsberekening) of multi-storey residential buildings, which are typically mid-rise concrete buildings. This results in the following research objective:

> *Research and develop a method and tool that can be used to automate the manual calculation processes in structural calculations, specifically in the vertical load calculation of a multi-storey residential building, based upon a 3D BIM model*

2.2. Research questions

The main research question is derived from the research objective stated above:

> *How can manual calculation processes in structural calculations, specifically in the vertical load calculation of multi-storey residential buildings, be automated using a 3D BIM model?*

The following sub-questions are formulated based on this main research question.

1. *How is the vertical load calculation currently organised and which software is used to support the calculation?*
2. *What methods and software tools already exist that strive to automate similar processes?*
3. *What currently available software can be used to automate the vertical load calculation?*
4. *What are requirements for a method and/or software tool that strives to automate the vertical load calculation?*
5. *How should a method that strives to automate the vertical load calculation be composed?*
6. *Can such a method be successfully applied to practical cases?*

2.3. Methodology

The proposed structure of the report is given in Figure 2.1 below. This proposed structure is based upon the design stages distinguished by Dym et al. (2013).

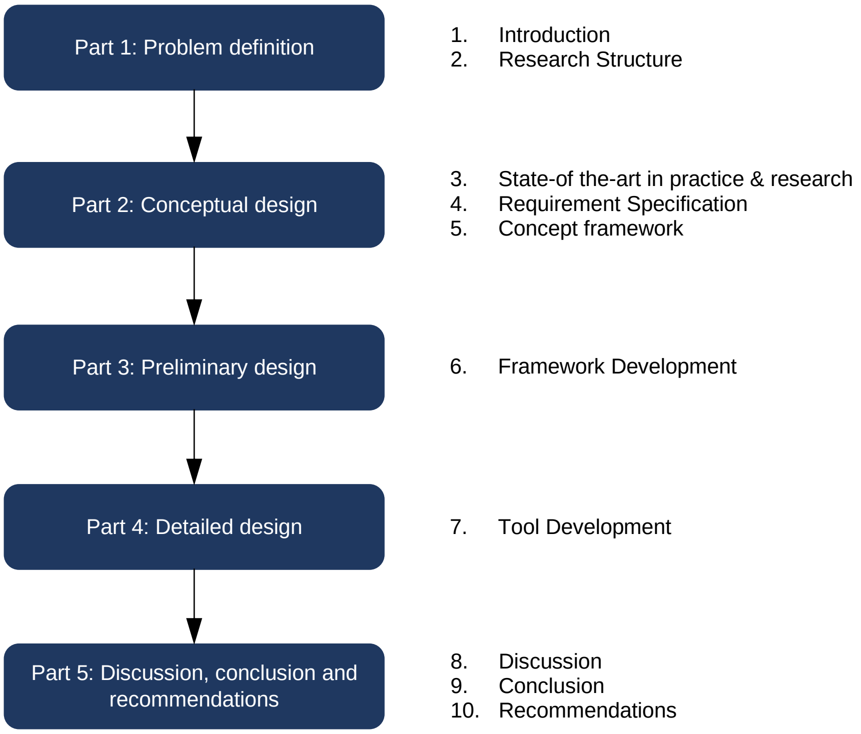

Part 1: Problem definition
1. Introduction
2. Research Structure
Part 2: Conceptual design
3. State-of the-art in practice & research
4. Requirement Specification
5. Concept framework
Part 3: Preliminary design
6. Framework Development
Part 4: Detailed design
7. Tool Development
Part 5: Discussion, conclusion and recommendations
8. Discussion
9. Conclusion
10. Recommendations

Figure 2.1: Research outline

In Part 1 the problem context of the research is defined based on an initial literature review, after which the research objective, research questions, and methodology are formulated.

Part 2 provides the state-of-the-art in both practice and research regarding the automation of structural design processes and specifically of the vertical load calculation. The state-of-the art in practice is defined based on a close observation of the current organisation of the vertical load calculation within CAE Nederland and by performing semi-structured interviews with structural engineers of other Dutch engineering firms. The state-of-the-art in research is defined on the basis of a literature review of currently available tools, software, and methods that automate structural design and calculation processes.

Part 2 also incorporates a requirement specification for the development of a method and tool that strive to automate the vertical load calculation. The formulated requirements are derived from the state-of-the-art. Using these requirements a concept framework is developed, in which the general idea behind the proposed method for the automation of the vertical load calculation is described and explained using flowcharts.

In Part 3 the framework is developed in more detail. To be able to so, the scope of the framework is narrowed. This allows for a more detailed description of how the different process steps should be performed and the assignment of responsibilities to the different roles involved in the vertical load calculation.

In Part 4 it is investigated if the proposed method can be successfully applied to some practical cases. For this proof of concept, the developed framework is (partially) implemented in a prototype tool and tested using a number of test cases. The practical applicability is further tested in user tests, in which two structural engineers performed a vertical load calculation using the prototype tool. Challenges that are faced during the implementation and limitations of the tool that are identified during the tests are discussed.

Part 5 is the final part of the report. The tool and framework are discussed from the perspective of achieving the research objective and regarding their limitations. The answers to the research questions are given in the conclusions, and recommendations for future research and development are proposed.

Part 2: Conceptual Design

3

State-of-the-art in practice and research

This chapter gives the state-of-the-art of the vertical load calculation and the automation of structural calculations. The current organisation of the vertical load calculation at CAE Nederland is described and compared with that of other Dutch structural engineering firms. A literature review was performed to identify methods and automation tools that can serve as a reference for the automation of the vertical load calculation. Those are analysed and categorised, after which an overview of suitable approaches and software used by the reference tools is given.

3.1. Current practice of the vertical load calculation

In this section the current organisation of the vertical load calculation is defined based on a close observation of the current organisation the vertical load calculation at CAE Nederland. The two most important persons involved in the vertical load calculation are the structural engineer and the BIM modeller. The structural engineer is responsible for the design of the load-bearing structure of the building, and should inform the BIM modeller about the dimensions and position of the different elements. The digital drawings and models the BIM modeller produces, can be used for further collaboration within the company and with other external parties involved in the building project.

The ultimate purpose of the vertical load calculation is to determine the loads acting on the foundation. Given the scope, the focus in this thesis will mainly be on pile foundations. In that this goal comes down to determining the number of piles needed to support the building. Therefore, the pile force reactions should be obtained and justified. In order to do this, the way the loads are transferred through the load-bearing structure are defined and recorded in a schematisation of the load-bearing structure. Also, the first dimensioning of structural members (wall thicknesses, column dimensions, etc.) is performed.

In the process of defining the schema, the structural engineer may encounter situations that can lead to problems. Therefore, he or she applies a quick check to verify if the elements that might lead to a problem can take up the loads that are applied on them. Furthermore, the vertical load calculation forms the basis for more detailed calculations in a later stage of the design process, where the loads are deduced from the vertical load calculation.

3.1.1. Calculation processes

A flowchart to illustrate the organisation of the calculation processes in the vertical load calculation is shown in Figure 3.1. The calculations are started on the basis of the design starting points, which consist of a summation of all the principles that are used in the vertical load calculation. This includes information about the location of the project, functions, and the applicable codes and regulations that are used. Based on this information and the initial building geometry, it is determined what loads are acting on the building. The loads are organised in load packages, which are convenient groups of loads that act on the same building element (an area load for a residential floor, for example, can consist of the self-weight of the floor, a topping and partitioning walls). These load packages are also part of the design starting points.

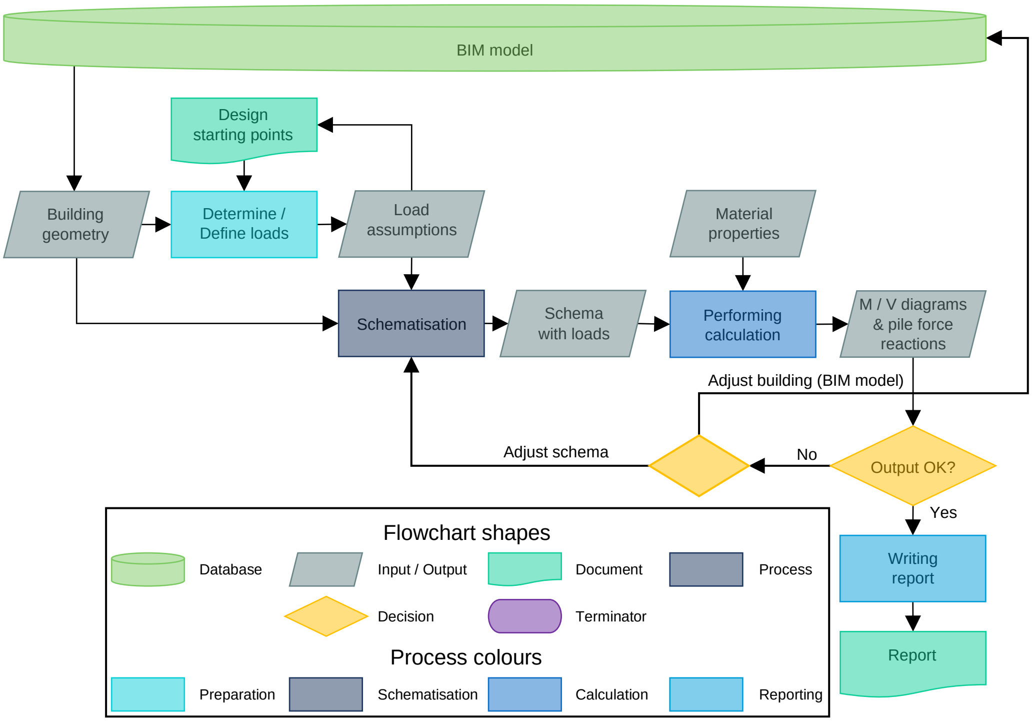


Figure 3.1: Flowchart of the current organisation of the calculation processes in the vertical load calculation

The actual vertical load calculation consist of two major parts, that both contain several sub-processes. These are the schematisation and the actual calculation itself. In the schematisation process it is defined how the building will be schematised to be able to perform the actual calculation. The structural engineer can utilise several types of schematisations. In this MSc thesis those schematisations are divided into four different categories, with increasing complexity, that are discussed below (Figure 3.2):

- 0D schema
- 1D schema
- 2D schema
- 3D schema

0D schema
The complete building can be schematised as a point load to make a quick initial guess of the number of piles required based on the total building weight. This schematisation type is not used in the actual calculation process.

1D schema
This schematisation type is used when the dimensions, spans (directions), and grid sizes are more or less fixed. Based on this the load-bearing walls on each of the grid axes can be schematised as a 1D line element. In this schematisation type the focus is on the level of the foundation, it is only defined where and with what magnitude the different loads arrive on the level of the foundation. Also the way the connection to the foundation piles is schematised is defined (hinged, elastic of fixed). The outcome is a linear division of the piles along the considered grid axis.

2D schema

A 2D schema is more detailed than a 1D schema, which implies that more effort is required to make one. Therefore, this type is only used when a 1D schema is not sufficient to satisfy the purpose of the vertical load calculation.

A complete load-bearing wall can be schematised as a vertical 2D schema consisting of a combination of 2D wall elements and linear 1D elements. Compared to a 1D schema, the way the loads are transferred through the building is now further specified, since there is no longer only looked at how the loads arrive at the level of the foundation. This results in a more detailed specification of pile reaction forces and of the linear division of the piles along the grid axis.

Alternatively, a horizontal 2D schema can be used in which the complete foundation level is schematised as a 2D plate element.

3D schema

In the final phases of the design a complete 3D calculation can be performed, in which the complete building is schematised using a combination of 2D plate elements and 1D linear elements. This schematisation type is not often used in practice, as 2D calculations are more clear and insightful for most cases. In addition, 2D calculations are often already specific enough to fully satisfy the purpose of the vertical load calculation. In Figure 3.2 an example is given of how the different types of schematisation would look like for a multi-storey building.

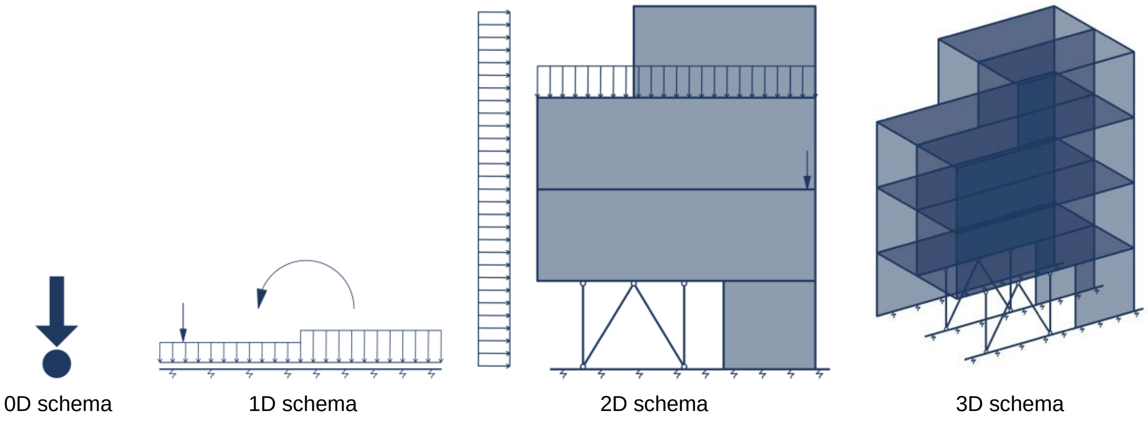


Figure 3.2: Example of the different levels of schematisation used in the vertical load calculation of a multi-storey building

As a final note, it needs to be mentioned that the calculation processes are not linear, but are performed in loops as the design of the building is constantly changing, requiring an alteration in the way the loads are transferred through the building. This phenomenon is also displayed in the flowchart in Figure 3.1.

Software assisting the calculation processes

Currently the vertical load calculation is often assisted by Excel spreadsheets. The spreadsheets are set up in such a way that the user only has to fill in or add the project specific information, such as geometry, dimensions, materials, etc. The spreadsheets contain predefined load packages, which can be combined to come up with the different loads acting on each part of the building. In this manner, the way the loads are transferred through the structure can be determined and those loads can be quantified, resulting in a schematisation with the different loads assigned.

Structural analysis software can then be used to come up with the pile reaction forces based on the defined schema. In case of CAE Nederland this is mainly SCIA Engineer and Technosoft. Other structural analysis software tools that are currently most used in structural engineering practice are Ansys, AxisVM, Diamonds, DIANA, GSA, IDEA, MatrixFrame, QEC, RFEM/RSTAB, Robot Structural Analysis, Sofistik, and Struct4U (Veenendaal, 2018). In Section 3.4 an overview is given which of these software might be useful for the automation of the vertical load calculation.

3.1.2. BIM modelling processes

In engineering practice, the different disciplines create their own BIM model: architectural, structural, installations, etc. The structural engineering firm makes a BIM model of the load-bearing structure. This BIM model is started on the basis of the model of the architect, who has made an initial estimation of what the load-bearing structure should look like. Most often, the grid the architect has proposed is used. To assure the integration of the load-bearing structure in the rest of the building, clash-detection is performed multiple times over the course of a project.

The modelling processes are structured using the concept of "Level of Development" (LOD), which defines the required level of detail of all information in the model, both geometric and other information (Borrmann et al., 2018). To illustrate the concept of "Level of Development", an example is indicated in Figure 3.3. The following levels of development represent the different steps in the modelling process that are relevant for the vertical load calculation:

- LOD200
- LOD300
- LOD350

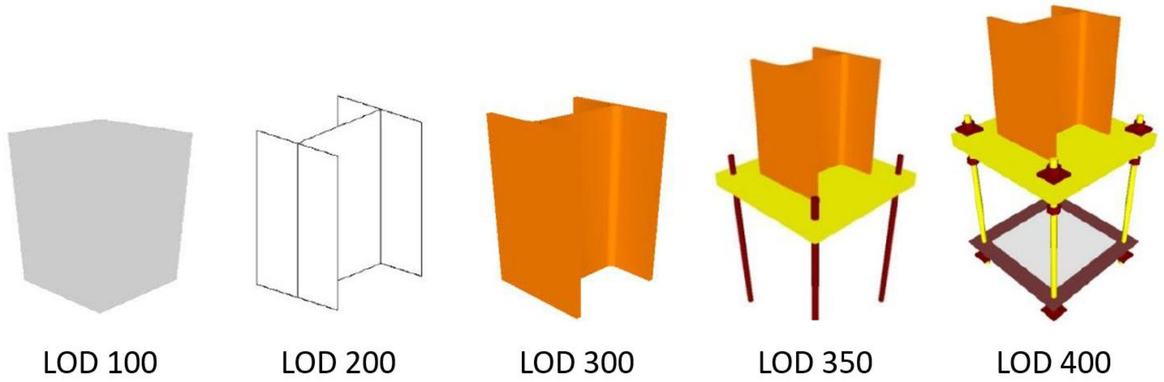


Figure 3.3: Example of the level of development principle applied to a steel column and its connection at the bottom (BIMforum, 2019)

LOD200
A LOD200 BIM model should show the positioning and type of elements used, as well as the amount of space that is occupied by those elements, as dimensions are still approximate in this phase. The different elements are classified using the NL-SfB-classification (BIM Loket, 2019a). If already known, information regarding material properties, like the concrete strength class, is added.

LOD300
In this phase the dimensions of the different elements are made exact, their measurements are being included in the model. The approximate dimensions of openings and cutouts are added. If not included earlier on, the material properties, as well as other relevant information, are added to the model.

LOD350
In this phase the dimensions of openings and cutouts are made exact, by including their measurements in the model.

BIM software

Within CAE Nederland, Revit (Autodesk Inc., 2019d) is used as BIM software. Revit is probably best known for its architectural application, but it also has a structural version: Revit Structure (Autodesk Inc., 2019e). It can be used to draw and analyse structural geometry, as well as share and visualize designs to help cooperation. Other capabilities include modelling of the concrete reinforcement, steel details, detailed documentation of the designed structures, a bi-directional connection with Robot Structural Analysis software, a connection with Dynamo (a parametric modelling tool) and a connection with concrete and steel fabrication tools (Autodesk Inc., 2019a). Revit is considered easy to use due to its well-designed and user-friendly interface and it is also considered as a strong BIM platform since it supports information exchange with a large range of other software tools (Eastman et al., 2011; Qin, 2018). Revit is currently the most used BIM software (RIBA, 2019).

Other BIM software tools that are used in structural engineering practice are Tekla Structures, ArchiCAD, Allplan, Infraworks, and Civil 3D (Veenendaal, 2018). In Section 3.4 an overview is given of the available software that might be useful for the automation of the vertical load calculation.

3.1.3. Comparison of modelling and calculation processes

Modelling and calculation processes are compared by placing them along the relevant phases of the general design process. Since there exist many views on how the general design process should be organised, the RIBA Plan of Work 2013 (RIBA, 2013) is used here. This plan gives the standard model for the building design and construction process in the UK. Eight different stages are defined:

0. Strategic Definition
1. Preparation and Brief
2. Concept Design
3. Developed Design
4. Technical Design
5. Construction
6. Handover and Close Out
7. In Use

The stages relevant for the vertical load calculation are: the preparation and brief, the concept design, the developed design, and the technical design. The comparison of calculation and modelling processes along the general design process is indicated in Figure 3.4.

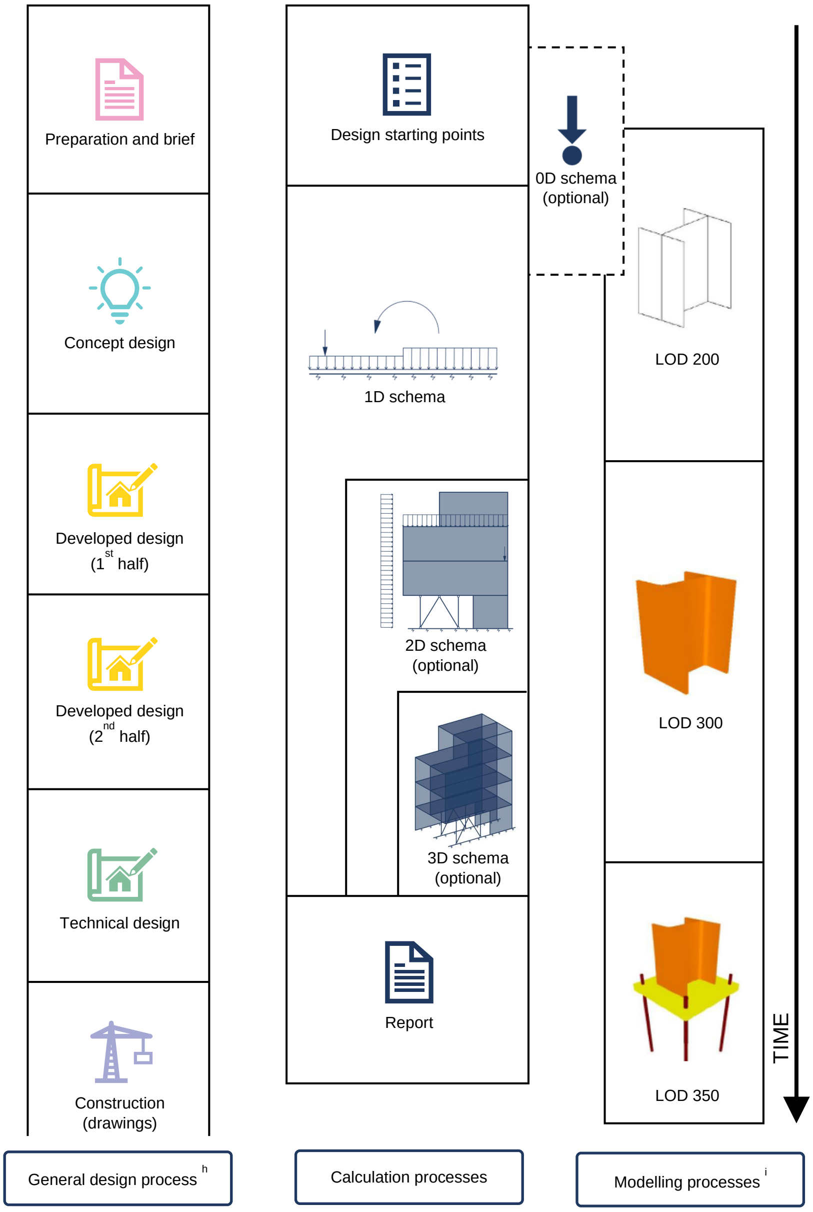


Figure 3.4: Comparison of the different phases that can be identified in the modelling and calculation processes; [a] Based on RIBA (2013); [b] Based on BIMforum (2019)

From the above figure it can be seen that the different processes are not completely simultaneous. This is not a problem since the boundaries between the different phases are not hard boundaries that have to be followed strictly. Generally, as the design progresses through the different design phases, the extent to which the design is specified increases. The modelling processes follow along this increase in a similar manner, as more information is specified in the BIM models, as the LOD increases. The calculation processes do not necessarily follow along the same line. If a 1D schema is sufficient and reliable enough to fulfil the purpose of the vertical load calculation, there is no need to move on to a 2D schema. The same holds in case of a 2D schema with respect to changing to a 3D schema.

At CAE Nederland most of the work, about 60%, is performed during the development phase of a project (Concept - Technical design). In this part, most of the work is done when the LOD300 BIM models are used: 60-70% of the total workload of the development phase (S.Kiefte, personal communication, December 20, 2019).

3.2. Comparison with Dutch engineering practice

3.2.1. Interviews

To compare the current practice at CAE Nederland as described in Section 3.1, with that of other Dutch engineering firms interviews were performed. Four structural engineers of three engineering firms that operate within the Dutch building industry, as well as three structural engineers of CAE Nederland were interviewed in six different interviews. In the interviews a semi-structured approach was applied in which a few questions were used to guide the conversations and to allow the participants to come up with their own input. The questions in the interviews performed with the external structural engineers were focused on the current level of automation within their company. The questions in the interviews with the engineers of CAE Nederland were focused on their view on the current organisation of the vertical load calculation. An overview of all the participants is given in Table 3.1 below.

Table 3.1: Interviewees

No.	Position	Experience	Company	Date
1	Structural engineer (affinity with automation & optimisation)	>5 yrs	Dutch engineering firm with >150 employees, specialised in building design	2-12-2019
2	Structural engineer	>20 yrs	Dutch engineering firm with >250 employees, specialised in building design	9-12-2019
	Structural engineer & computational designer	>5 yrs		
3	Structural engineer (affinity with parametric design)	~3 yrs	International engineering firm with >13800 employees worldwide and 1 office located in the Netherlands (>250 employees), specialised in building design	12-12-2019
4	Structural engineer	~3 yrs	CAE Nederland	20-02-2020
5	Structural engineer	~3 yrs	CAE Nederland	21-02-2020
6	Structural engineer & project manager	>10 yrs	CAE Nederland	09-03-2020

The interviews were analysed using a thematic analysis in which there is aimed to provide a rich description of the entire dataset that is obtained through the interviews (Braun & Clarke, 2006). The names of the themes are derived from the dataset itself (Kissling, 1996), and those themes are illustrated using excerpts from the dataset (Breakwell et al., 2000). The documentation of the interviews is included in Appendix A.

3.2.2. Findings from interviews

Current organisation of the vertical load calculation

From the analysis the interviews with external structural engineers (Appendix A), it followed that all four participants find the current practice as described in Section 3.1 comparable to the way the vertical load calculation is organised within their company. It also followed, that the main difference between companies is the way the building is split into parts that are considered separately in the vertical load calculation. The buildings are either divided horizontally or vertically. In the first case, the vertical load calculation is performed per floor, in the latter case the vertical load calculation is performed per wall. This is clarified in Figure 3.5.

In case the vertical load calculation is performed per load-bearing wall or grid axis, the loads acting on all floors are directly placed upon the schematisation of the wall. The loads on the foundation are obtained directly. Furthermore, it allows to clearly see how the loads from the wind are distributed to the foundation.

In case the vertical load calculation is performed per floor, the distribution of the loads is defined per floor. The calculation is started at the roof level, after which there is worked towards the level of the foundation. With each floor going down, the loads from that floor are added to those from the floors above. In that way, it can be clearly seen what loads act on the load-bearing structure per floor.

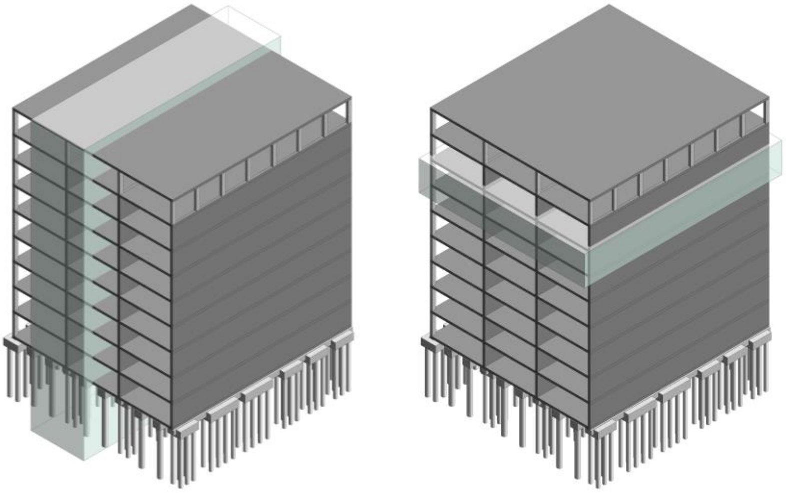


Figure 3.5: Considered building part in a vertical load calculation performed per wall and per floor respectively

The first approach is best suitable for buildings that mainly contain load-bearing walls and/or with one main load-bearing direction. The second approach is best suitable for buildings that mainly contain columns and/or two-way-bearing floors.

Digital transformation of the building industry

All four structural engineers that participated in the interviews noted the digital transformation of the building industry. Their opinions differ on the timing and the extent of this transformation. One participant believes that completely manual calculations will always remain, while the others believe all calculations will eventually be automated.

Current level of automation

The current level of automation varies between the companies that were interviewed. Within one company, they deliberately chose not to automate their calculations as of yet, as the projects they are typically working on can be designated as ‘one-off projects’. Within another company manual calculations are still more often used, because they are easier to perform and they provide more insight, especially for the more simple and smaller projects they are working on. On the other hand, within the third company there is a special department that automates parts of their calculations.

The four structural engineers that participated in the interviews do not know a company that is automating the vertical load calculation on the basis of a BIM model. They named other structural engineering firms that automate (parts of) their calculation processes, as well as companies that are hired to do the automation. Furthermore, it was mentioned that vendors of structural analysis software are working on improving the automatic generation of calculation models from a BIM model.

3.3. Reference tools

In this section an overview is given of currently existing tools that automate structural design processes using BIM. This overview is non-exhaustive, as only tools that were found in literature are included. From the interviews with external structural engineers it followed that companies are working on automating their structural calculations (see Subsection 3.2 and Appendix A), which is why more tools exist than described in literature.

The literature on BIM based tools for the building industry is diverse. Extensive literature can be found on the topics of safety design, construction cost estimation, site layout design, quantity take-off, construction scheduling (Liu et al., 2018), construction waste estimating (Liu et al., 2015), building energy analysis (Farzaneh et al., 2018), code compliance checking (Preidel & Borrmann, 2015), and construction preparation (Aram, 2015). Compared to the abovementioned topics the amount of literature on structural automation tools is limited. Below an overview is given of reference tools found in literature.

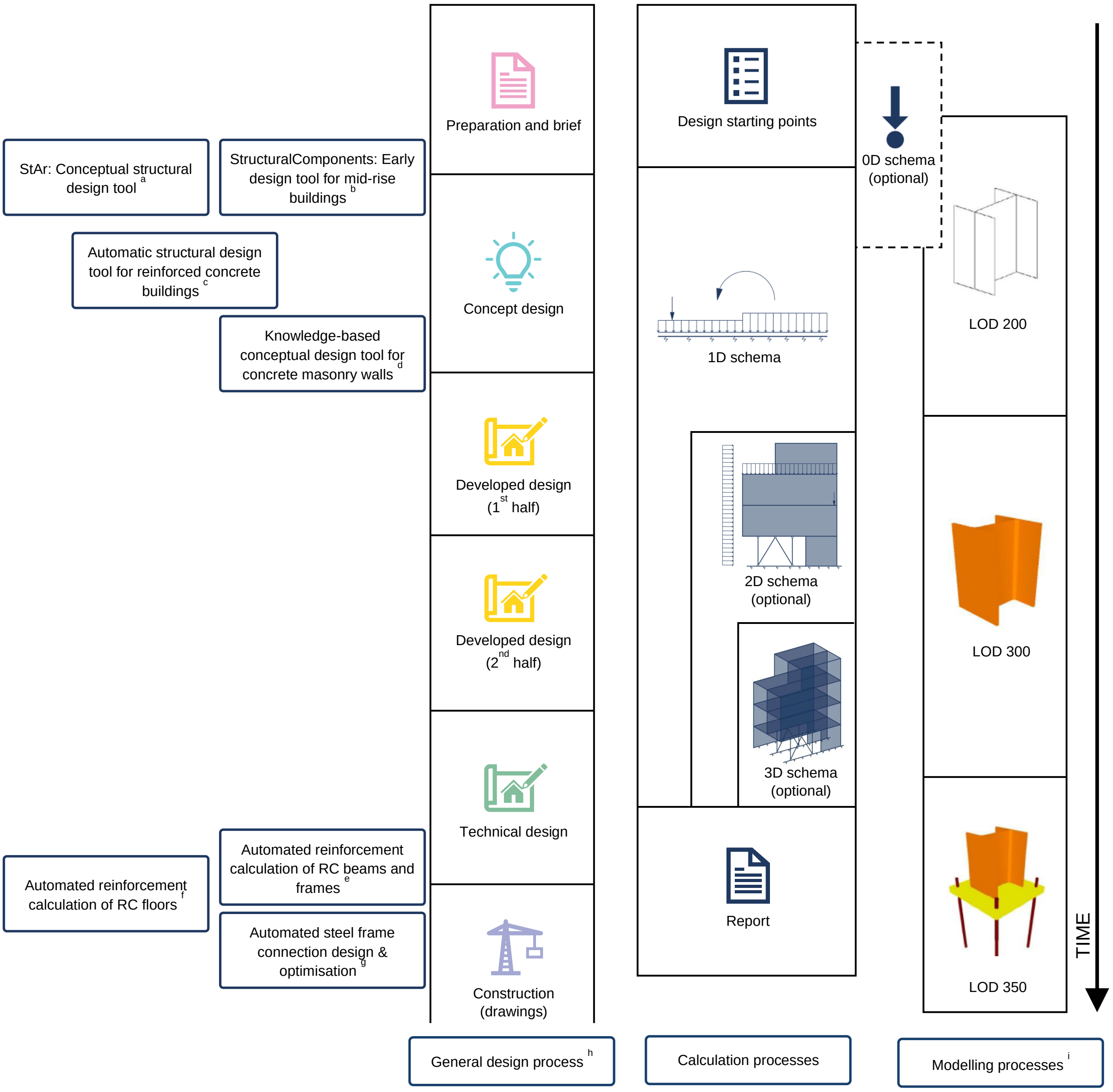

StAr: Conceptual structural design tool a
StructuralComponents: Early design tool for mid-rise buildings b
Automatic structural design tool for reinforced concrete buildings c
Knowledge-based conceptual design tool for concrete masonry walls d
Automated reinforcement calculation of RC floors f
Automated reinforcement calculation of RC beams and frames e
Automated steel frame connection design & optimisation g
Preparation and brief
Concept design
Developed design (1st half)
Developed design (2nd half)
Technical design
Construction (drawings)
Design starting points
1D schema
2D schema (optional)
3D schema (optional)
Report
0D schema (optional)
LOD 200
LOD 300
LOD 350
TIME
General design process h
Calculation processes
Modelling processes i

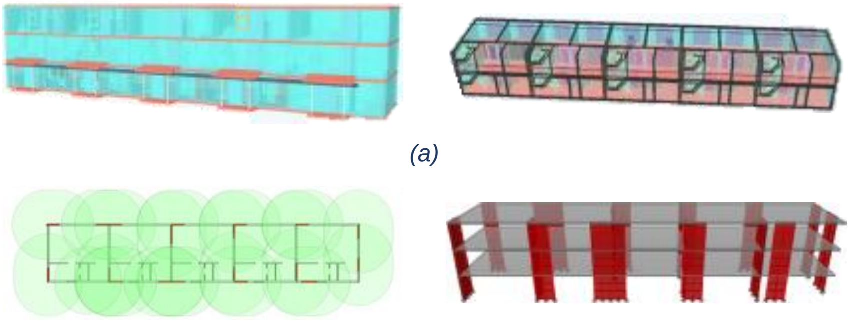
Figure 3.6: Position of the different reference tools found in literature within the general design process compared to the vertical load calculation; [a] Mora et al. (2006), Mora et al. (2008); [b] Bovenberg (2015), Dierker Viik (2019), Hohrath (2018), Niño Romero (2019); [c] Tafraout (2019); [d] Cavieres et al. (2011); [e] Mangal & Cheng (2018); [f] Eleftheriadis et al. (2018); [g] Nguyen & Vu (2020); [h] Based on RIBA (2013); [i] Based on BIMforum (2019)

The reference tools are classified according to their position within the general design process, to determine to what extend those tools can serve as a reference for the automation of the vertical load calculation. From Figure 3.6 it follows that the reference tools can be classified into two groups: conceptual design and technical design tools. Below there will be elaborated on those tools shortly.

3.3.1. Conceptual design tools

Conceptual design is highly iterative and intuitive, because information is missing or still subject to change. The major challenge for conceptual design tools is therefore to bridge the representation difference between BIM and conceptual design (Cavieres et al., 2011). This is interesting for the automation of the vertical load calculation, where the representation difference between BIM model and schematisation also needs to be bridged, as will be discussed in more detail in Subsection 4.1.2. Below the conceptual design tools as depicted in the overview in Figure 3.6 are shortly described.

Both Mora et al. (2008) and Tafraout et al. (2019) developed a framework and tool that should help the structural engineer with the interpretation of architectural models and assist in the structural design. Mora et al. (2008) developed StAr, which is a prototype software of the framework they proposed, developed in Visual C++. Their tool is not fully automated on purpose, as they state that user interactivity is important for a structural engineer, since it assures that the engineer is always in control of the process. The tool assists the structural engineer in the finding of load paths in architectural models, using several different algorithms that can be activated by choice. Those algorithms verify, for example, if certain elements can be classified as load-bearing without conflicting with architectural constraints (Mora et al., 2008).
Tafraout et al. (2019) developed a similar framework and tool, that generates a set of conceptual structural designs on the basis of an initial, architectural BIM model and a set of structural requirements. The approach is based on a genetic algorithm (GA) that starts with an initial population of solutions that fulfil to the requirements. After which a series of evolutions is performed in which the initial population evolves using crossovers and mutations. During this process the solutions that best comply with the requirements are kept. An example of the use of this tool is given in Figure 3.7. The generation of such a structural design takes 50-100 iterations (Tafraout et al., 2019).



(a)

(b)

Figure 3.7: (a) Architectural BIM model which is used as input for the automation tool; (b) Automatically created structural model of the reinforced concrete wall system (Tafraout et al., 2019)

Cavieres et al. (2011) developed a knowledge-based tool for the automation of conceptual design of masonry walls implemented in parametric modelling software GenerativeComponents (Bentley). Their aim was to embed construction and structural design knowledge in generative rules and feedback rule-checking functions. In contradiction to the approach proposed by Mora et al. (2008), the technical knowledge is implemented within the tool completely, limiting the structural engineer’s influence.

Another structural design tool that is focused on automation of structural conceptual design is StructuralComponents. This ongoing project is capable of quickly testing multiple structural design alternatives. The tool aims at providing easy usability, easy changeability of design logic, insight in design choices and improved collaboration (Bovenberg, 2015; Viik, 2019; Hohrath, 2018; Romero, 2019), which corresponds with the view of Mora et al.(2008). There are also differences: StAr is developed for more traditional BIM software like Revit, while StructuralComponents uses its own version of a BIM model that utilises larger, more simple elements than those in traditional BIM software. These building blocks can be easily combined and adapted, making them more suitable for conceptual design than the lower-level elements of traditional BIM software (Bovenberg, 2015; Hohrath, 2018).
At the moment there are up to seven different versions of StructuralComponents. The two most recent versions (version 6 and 7), were incorporated in a Python script which is implemented in Grasshopper to generate parametric geometries (the BIM model within StructuralComponents) (Viik, 2019; Romero, 2019).

3.3.2. Technical design tools

During the technical design phase the design is developed in such detail that production information can be prepared (Designing Buildings Ltd., 2019).The focus is therefore mainly on the single elements or the connections between elements. This can also be seen in the technical automation tools that were reviewed. While in the vertical load calculation the focus is more on the complete building, it is still interesting to investigate the approaches that were taken in those software tools, as each of the single elements in the BIM model is of importance for the whole.

Mangal & Cheng (2018) present a BIM-based framework for the automated optimisation of reinforced concrete beams and reinforced concrete building frames. A similar framework for floors is developed by Eleftheriadis et al. (2018). Both approaches follow more or less the approach. First, all relevant information about the elements that will be reinforced is extracted from the BIM model. Then, the section forces are determined, based on which the required amount of reinforcement can be calculated. The final step is the optimisation of the reinforcement, in which an optimal configuration of the reinforcement bars is sought, also taking practical considerations into account (Eleftheriadis et al., 2018; Mangal & Cheng, 2018). A genetic algorithm (GA) is used to find this optimal configuration. Both approaches were implemented using a BIM model made in Revit, linked to Robot Structural Analysis software by a program written in C# (Eleftheriadis et al., 2017; Mangal & Cheng, 2018). In Figure 3.8 an image is included to give an example of how the genetic algorithm is applied.

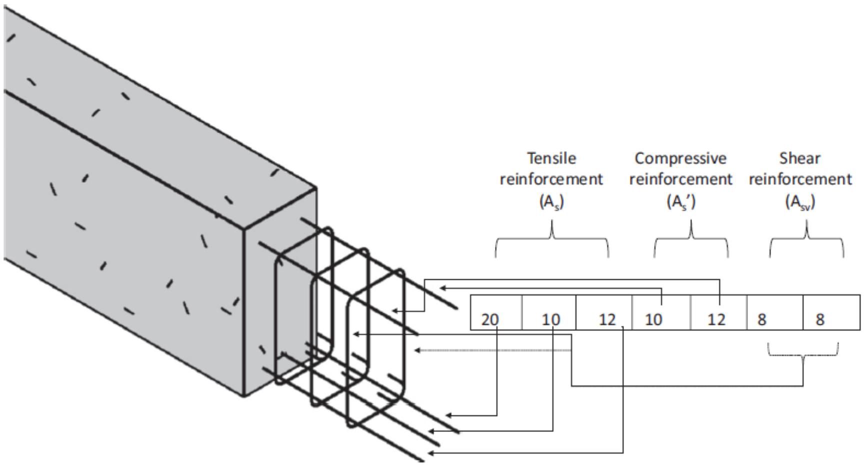

Tensile reinforcement (A_s)
Compressive reinforcement (A_s')
Shear reinforcement (A_{sv})
20
10
12
10
12
8
8

Figure 3.8: Example of gene encoding for the application of the genetic algorithm to a reinforced concrete beam (Mangal & Cheng, 2018)

Nguyen & Vu (2020) propose a framework for the automatic optimisation of steel connections based on the information in a BIM model. After which the optimised connections are automatically generated within the BIM model. To do this, all information relevant for the calculation and optimisation is first retrieved from the BIM model. A differential evolution (DE) algorithm is then used to derive the optimal connection configuration. A differential evolution algorithm differs from other evolutionary algorithms since it uses a linear combination of a randomly selected group of solutions instead of a combination of crossovers and mutations (Storn & Price, 1997). The approach was implemented using Tekla Structure as BIM software. A .NET toolkit utilising Tekla Open API was developed to implement the differential evolution algorithm. The results are implemented back into the BIM model (Nguyen & Vu, 2020).

3.4. Approaches & Software

The reference tools discussed in the previous section make use of a number of approaches and software packages to automate structural design processes. These approaches can be classified as follows:

- Knowledge based engineering
- File-based exchange (proprietary & public exchange formats; direct links)
- Generative design
- Evolutionary algorithms (genetic algorithms; differential evolutionary algorithms)
- Parametric engineering

In this section there will be elaborated on the approaches that are considered suitable for the automation of the vertical load calculation. Therefore, generative design and evolutionary algorithms are excluded, as they are more useful for the generation of design alternatives or the optimisation of specific building elements. First, an overview is given of the most used software for BIM, structural analysis and parametric design. After which the principles of each of the suitable approaches are discussed.

3.4.1. Software overview

In Table 3.2-3.4 a non-exhaustive overview is given of the software that can be used to automate the vertical load calculation.

Table 3.2: Software overview of BIM, structural analysis and parametric software

	Import	Export	Plugins	URL of website
Allplan (Nemetschek)	DXF, DWG, IFC, SketchUp (*.skp), Rhinoceros (*.3dm), SCIA Engineer (*.asf)	DXF, DWG, IFC, SketchUp (*.skp), Rhinoceros (*.3dm), Excel (*.xlsx)	SCIA Engineer	https://www.allplan.com/en/
ArchiCAD (Graphisoft)	DWG, DXF, IFC, SketchUp (*.skp), Rhinoceros (*.3dm), Excel (*.xlsx)	DWG, DXF, IFC, gbXML, SketchUp (*.skp), Rhinoceros (*.3dm), Excel (*.xlsx)	Grasshopper	https://www.graphisoft.com/archicad/
Revit (Autodesk)	DWG, DXF, gbXML, IFC, RVT, SketchUp (*.skp), Rhinoceros (*.3dm)	DWG, DXF, gbXML, IFC	Dynamo, Grasshopper, SCIA Engineer (CADS)	https://www.autodesk.nl/products/revit/overview
Tekla Structures (Trimble)	DWG, DXF, IFC, SketchUp (*.skp)	DWG, DXF, IFC, SketchUp (*.skp)	Grasshopper	https://www.tekla.com/products/tekla-structures

Table 3.3: Software overview of structural analysis software

	Import	Export	Plugins	URL of website
AxisVM	DXF, IFC	DXF, IFC, Tekla Structures (*.asc, *.stp), Allplan (*.asf)	Grasshopper / Rhinoceros (*.gh, *.3dm), Dynamo / Revit (*dyn, *rvt)	https://axisvm.eu/index.html
MatrixFrame	DXF, IFC	PDF, Excel (*.xls)	Dynamo / Revit, Tekla Structures	https://www.matrix-software.com/nl/structural-engineers/matrix-frame
RFEM / RSTAB	DXF, IFC, Tekla Structures (*.stp), SCIA Engineer (*.xml), Excel (*.xlsx)	IFC, Excel (*.xlsx), Allplan (*.asf)	Revit, Tekla Structures	https://www.dlubal.com/en
Robot Structural Analysis (Autodesk)	DWG, DXF	DWG, DXF	Dynamo/Revit	https://www.autodesk.com/products/robot-structural-analysis/overview
SCIA Engineer (Nemetschek)	DWG, DXF, IFC	IFC	Allplan, Grasshopper / Rhinoceros, Dynamo / Revit, Tekla Structures	https://www.scia.net/nl/software/scia-engineer
Sofistik			Grasshopper / Rhinoceros, Dynamo / Revit	https://www.sofistik.de/

Table 3.4: Overview of parametric software

	Import	Export	Plugins	URL of website
Grasshopper (Rhinoceros)			Multiple, see: https://www.food4rhino.com/	https://www.grasshopper3d.com/
Dynamo (Autodesk)			Multiple, see: https://dynamopackages.com/	https://dynamobim.org/

3.4.2. Knowledge based engineering

Knowledge based engineering is an approach to design automation that utilises a description of rules to automate repetitive tasks (Zawadzki, 2018). This approach tries to achieve this goal by capturing technical design knowledge in the form of logical rules or algorithms, which are maintained and re-used to automate design and engineering processes (Verhagen et al., 2012).

BIM can be an useful means for knowledge based engineering, since its metadata can be used for the implementation of technical knowledge within the BIM-based automation tool.

Knowledge based tools are limited by their scope and the richness of their knowledge base (Wang et al., 2002), therefore it should be clear beforehand what information should be incorporated. Another challenge for the implementation of knowledge based engineering within the automation tool is the problem of how to capture and embed technical design knowledge into usable rules of logic (Cavieres et al., 2011).

Related to the principle of knowledge base engineering is the principle of machine learning. While knowledge based engineering still requires a manual inventory of the knowledge that needs to be implemented, machine learning is a method that obtains this knowledge automatically. Machine learning is defined as a computational method that utilises past information to be able to learn and make accurate predictions. This past information is a dataset that should be large enough and of high enough quality in order to be successful (Talwalkar, 2018). For the automation of an engineering process like the vertical load calculation, knowledge based engineering is still preferred over machine learning, since the machine learning technique works as a black box (de Maat et al., 2010).

3.4.3. File-based exchange

There are three types of file-based exchange: a direct link between two types of software, a proprietary exchange format and a public product data model exchange format. Direct links and proprietary exchange formats are developed by the companies involved, while the public data models are publicly managed.

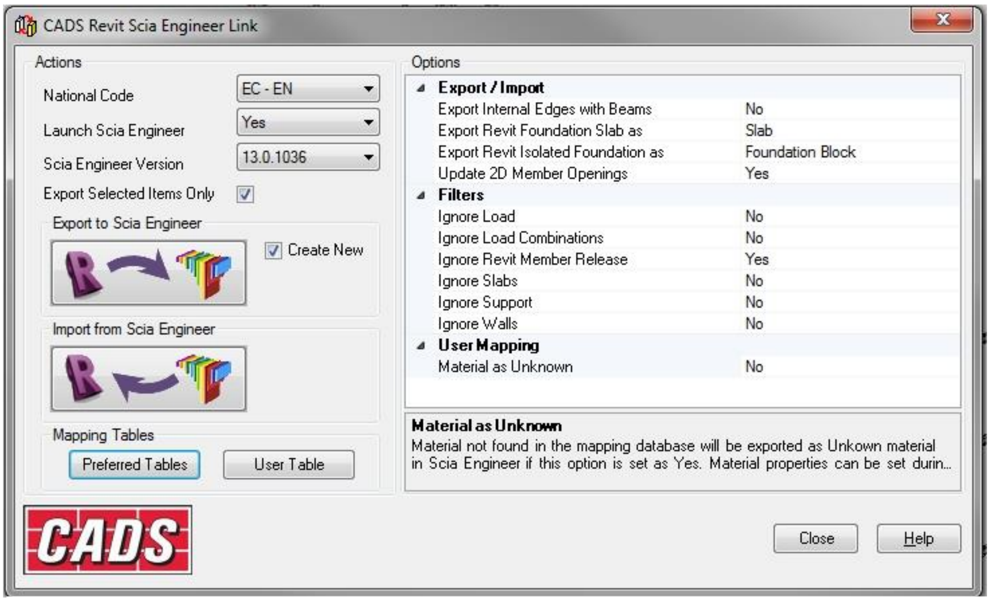
Application Programming Interfaces (API)

A direct link makes use of the Application Programming Interfaces (API) of the software that is connected. The API of a software product can be used to extract data from that application and import that same data into another software package using that software's API. Furthermore, the API can be used to extend the functionalities of the software (Eastman et al., 2011).

The most used BIM software, Revit (RIBA, 2019), offers users the ability to interactively design and manipulate Revit using algorithms and computational logic through the Revit API (Yang et al., 2018), which supports the use of C#, VB.NET, and C++/CLI programming languages (Autodesk Inc., 2019c). Furthermore, Revit Python Shell and pyRevit can be used to connect to the Revit API. These tools use Iron Python, which is an implementation of Python written in C# allowing Python to run on windows systems (Autodesk Inc., 2019g).

Other BIM software, such as Tekla Structures, also provides the possibility to develop your own tools using the Tekla Open API, which uses the C# programming language (Trimble, 2019b).
To develop tools that connect to the API, knowledge is required of those object-oriented programming languages. Python is generally considered the best choice for engineering applications, since it is more easy to learn and use by non-programmers because of its clear and readable structure (Donaldson, 2003; Fangohr, 2004).

Several vendors of structural analysis software provide direct links with BIM software in the form of plugins (see Table 3.2). These plugins make use of the BIM software's API to translate data to structural analysis software. An example of such a plugin is the CADS plugin of SCIA Engineer that allows for a bi-directional connection between SCIA Engineer and BIM software like Revit and Tekla Structures (SCIA, 2018). Such a plugin requires the user to manually define which elements to translate between the two software packages and which materials to use, as can be seen from the interface of CADS that is shown in Figure 3.9.



CADS Revit Scia Engineer Link

Actions

National Code	EC - EN
Launch Scia Engineer	Yes
Scia Engineer Version	13.0.1036
Export Selected Items Only	☑

Export to Scia Engineer

☑ Create New

Import from Scia Engineer

Mapping Tables

Options

Export / Import	
Export Internal Edges with Beams	No
Export Revit Foundation Slab as	Slab
Export Revit Isolated Foundation as	Foundation Block
Update 2D Member Openings	Yes
Filters	
Ignore Load	No
Ignore Load Combinations	No
Ignore Revit Member Release	Yes
Ignore Slabs	No
Ignore Support	No
Ignore Walls	No
User Mapping	
Material as Unknown	No

Material as Unknown
Material not found in the mapping database will be exported as Unkown material in Scia Engineer if this option is set as Yes. Material properties can be set durin...

Figure 3.9: User interface for the link between Revit and SCIA Engineer

Direct links, whether provided by vendors of structural analysis software or created by the users of structural analysis and BIM software, provide excellent data exchange between the connected software packages. However, the connection is only successful for the intended use cases and for the specific versions of the software for which they were designed (Eastman et al., 2011).

Proprietary exchange formats
In Table 3.2-3.4 above, an overview is given of the mostly used exchange formats. The problem is that those formats are focused on specific data exchange scenarios resulting in the before mentioned 'Information Islands' (Borrmann et al., 2018). Within the building industry, there is a desire to use multiple different software tools to provide functionality that goes beyond the functionality that can be offered by software provided by a single vendor. Furthermore, when different organizations are collaborating within a project, it is not preferable to force the members of the project team to use the same software platform (Eastman et al., 2011).

Public data exchange formats
The most well-known public data model is IFC (Industrial Foundation Classes) (Eastman et al., 2011), other popular public data models are CityGML and gbXML (Borrmann et al., 2018).
Data exchange using public data exchange formats often results in loss of information and/or misrepresentations. BuildingSMART (buildingSMART, 2020a), the developer of the IFC data model, also developed standards that should limit the occurrence of those problems: the Information Delivery Manual (IDM) and the Model View Definitions (MVD) (buildingSMART, 2020b; buildingSMART, 2020c). These standards specify which information is delivered, between which individuals and at which moment in time. The IDM defines requirements related to the content of an IFC data model, to assure the quality of the data that will be exchanged. The MVD defines the mapping and technical application of those requirements to the IFC data model for a specific data exchange scenario (Borrmann et al., 2018).

In an effort to improve interoperability between BIM software and structural analysis software, several studies propose a method to extend on the idea of IDM- and MVD-standards. In this method, the interoperability is organised around a central data model that prevents the loss of information or misrepresentations that occur in the data exchange process. Lai & Deng (2018) state that these losses are mainly caused by the fact that the same information is named differently in different software packages and consequently different software packages export the same information using different IFC entities. They propose an IFC file as a central data model. In this central IFC file algorithms are implemented, that overcome the differences in representation syntax of the different tools that need to be connected. In this way no changes need to made in the data structure of commercial software tools (Hu et al., 2016; Lai & Deng, 2018).

Information in IFC-files can be extracted or extended using object-oriented programming languages like Python, Java, C++, and C# (Borrmann et al., 2018). As was the case for the direct links, this means that knowledge is required of those programming languages to be able to create or adapt the way the information in the IFC files is processed. Open libraries or tools that help to extract information from and insert new information in IFC files are IfcOpenShell, IfcPlusPlus, xBIM toolkit, IFC Engine, STEPcode, and IFC File Analyzer (Borrmann et al., 2018; Zhang & El-Gohary, 2018).

3.4.4. Parametric engineering

Parametric design can be defined as: "A process based on algorithmic thinking that enables the expression of parameters and rules that, together, define, encode and clarify the relationship between design intent and design response" (Jabi, 2013, p. 201). The main advantage of parametric design follows from this definition: by changing the values of parameters, the impact of that action on the design can be determined. This emphasizes why parametric design is a suitable method for design automation, as it helps to give insight in the logic behind the automation tool. A disadvantage of parametric design is that it is difficult to change the design logic, as a change in a relationship between two parameters, has consequences for all elements that are influenced by those parameters. Which could require the complete re-modelling of the part of the design logic influenced by the parameters whose relationship has to be changed (Rolvink et al., 2014).

The most used parametric design software are Grasshopper (Davidson, 2019a) and Dynamo (Autodesk Inc., 2019b). Both Grasshopper and Dynamo are moreover examples of Visual Programming Languages (VPL), which means they represent a modular system of rules and signs using visual elements instead of textual code. Therefore, they do not require any programming or scripting knowledge to use them (Preidel & Borrmann, 2015).

Both software have a built-in collection of those visual elements, but also allow for customisation. Grasshopper can be customized using the VB.net or C# component (Davidson, 2019b) and using the GH Python Remote component (Digital Structures, 2019). Dynamo contains a special Python-node (Autodesk Inc., 2019g), which allows for the customization as it connects to the Revit API. In addition, custom node creation also allows for customization (Autodesk Inc., 2019f).

3.5. Discussion

In this chapter the state-of-the-art of the vertical load calculation and the automation of structural calculations is investigated. It was found that the current organisation of the vertical load calculation at CAE Nederland is comparable to those at other Dutch engineering firms as followed from the interviews with external engineers. The main difference between companies is in the way the building is divided into parts that are considered separately in the vertical load calculation. All seven participants noted the advantages of automating (parts of) the vertical load calculation and believe the digital transformation of the building industry is the way to go.

Reference tools that automate structural calculation were identified using a literature review. The reference tools were classified according to their position within the general design process. It was found that most tools focus on the conceptual and technical design phase, while most of the work in a vertical load calculation is performed during the developed design phase. From the interviews it was also found that Dutch engineering companies are working on automating their structural calculations. The participants of the interviews did not know of an example of the automation of the vertical load calculation as proposed in this thesis.

The approaches and software used for the development of the reference tools were analysed and investigated. The following approaches were found to be suitable for the automation of the vertical load calculation:

- Knowledge based engineering
- File-based exchange (proprietary & public exchange formats; direct links)
- Parametric engineering

These approaches rely on the use of object-oriented programming languages, like Python and C#, to transfer data between BIM and structural analysis software. Alternatively, parametric software connected to BIM software, like Dynamo and Grasshopper, are used for the development of these automation tools.

In the next chapter a list of requirements will be formulated on the basis of the analysis of the reference tools and the outcomes of the interviews.

4

Requirement specification

In this chapter a list of requirements is specified that will be used for the formulation of the framework. First, an overview is given of the challenges and limitations of the use of BIM for the automation of the vertical load calculation. Then an overview is given of the features that should be implemented in a tool that automates the vertical load calculation. Those overviews are based on an analysis of the reference tools (Section 3.3) and the outcomes of the thematic analysis of the interviews (Appendix A). The different limitations and features are categorised in collections of themes. Finally, this results in a list of requirements.

4.1. Challenges and limitations of the use of BIM

4.1.1. Automation and software tools in general

Black box

A risk of automation is that it becomes unclear to the user of a tool how outcomes are obtained. In such a case we speak of a 'black box', and it should be prevented that a tool turns into a 'black box' (Viik, 2019). Four of the participants of the interviews acknowledge this, as they state that a structural engineer should always consider if the outcome of any software tool is correct and according to what can be expected.

Rules of logic

Another question that was raised during the interviews is if it is possible to automate everything? This question connects to the largest limitation of automation that was identified in Subsection 1.2: Processes can only be automated if they follow clear rules of logic. For structural design processes like the vertical load calculation it generally holds that they cannot be automated completely, as they involve design choices that cannot be represented by rules of logic.

Limited interoperability between different software tools

According to Borrmann et al. (2018) interoperability can be defined as: ''the loss-free exchange of data between software products by different vendors'' (p. 83). Currently, a lot of information is lost in the translation from one software package to the other. This is the case, because in the current use of BIM software the focus is on its use as 3D modelling technique, rather than on interoperability (Arayici et al., 2018) There is a lack of unified data sources that convert data between different types of software package and/or models (Chi et al., 2015). According to Lai & Deng (2018) the best possible way of data interoperability currently is through the direct conversion based on proprietary data formats. This however, results in so-called 'Information Islands': Data can only be conversed between tools of the same vendor (Borrmann et al., 2018).

The proposed solution to the interoperability issue are public file formats, of which the IFC-format (Industrial Foundation Classes) is the most popular one. However, public file formats have their own problems: misrepresentations and information losses. Those are likely to arise, if the mapping of parameters from the internal schemas of software to the IFC format, or the other way around, are wrong (Lee, 2011; Ma et al., 2006; Ren et al., 2018).

As the building industry is characterized by a wide variety of software tools and vendors, the tool should aim at overcoming those issues and be compatible with multiple software packages (Viik, 2019; Tafraout et al., 2019).

4.1.2. Representation difference between BIM and schematisation

In a BIM model, building elements cannot be represented using approximate dimensions, only using exact, strict values. Schematisations are more approximate and make use of simple lines or planes.

Information irrelevant for the calculation

A BIM model contains a lot of information that is irrelevant for a calculation. Even in the case of a BIM model in which only the structural elements are modelled. Thermal break elements, for example, are an essential part of the model, but are not relevant for the calculation.

Simplification for the vertical load calculation

For the vertical load calculation the exact details of the different elements in a BIM model are not interesting. Therefore, a lot of simplification takes place in a vertical load calculation as pointed out by five of the participants of the interviews. It is more practical and insightful if not every detail is taken into account in the calculation. Besides, local effects are not of interest for the determination of the loads acting on the level of the foundation.

Structural behaviour

As two of the participants of the interviews point out is the way the elements in a BIM model are modelled is not the way they are schematised in the calculation. Planar elements, like a wall for example, are modelled as one element, in which openings for doors and windows are then created using special ‘opening-elements’. The corresponding analytical model is still one element.

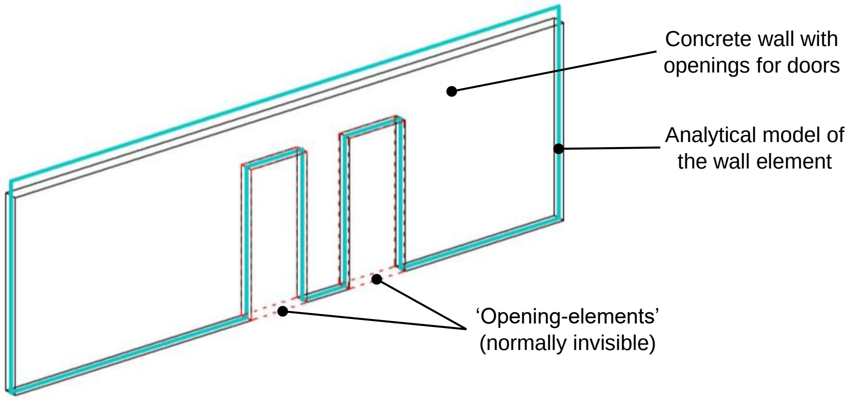

Concrete wall with openings for doors
Analytical model of the wall element
‘Opening-elements’ (normally invisible)

Figure 4.1: Analytical model of a wall element with two door openings. The 'opening-elements', using which the openings are created are normally invisible, for illustration purposes they are indicated using red dashed lines

For a calculation it can be desirable to split this element into multiple elements to define a lower E-modulus for the ‘beam’-element above the opening or to create a hinged connection between the three parts. In that way the schematisation is closer to the actual structural behaviour, which is obviously desired.

Conservative versus realistic calculating

Calculations are typically performed in a conservative manner. Loads are not determined exactly, but are approximate, rounded up, and often based on experience. This helps to quickly perform the calculation, while at the same time, additional structural safety and robustness is added to the design. Therefore, three of the structural engineers suggested that it might be interesting to investigate how this could be implemented in the automation tool. On the other hand, as is suggested by another structural engineer, a computer is capable of performing calculations quickly and accurately, thereby denying the need to add the additional structural safety and robustness to the design. However, the exact way of modelling in a BIM model, is not necessarily according to the realistic behaviour of the structure. We should therefore be careful with the direct use of the BIM model. As is also pointed out by another participant, who states that the danger of using 3D models is that it quickly becomes very complex, which should of course be prevented.

4.1.3. Correctness of the information in the BIM model

For the automation of the vertical load calculation it is of importance that all information relevant for the calculation is correctly implemented in the model.

Correct generation of the analytical model

When a BIM model is created, the focus is on the correct representation of the 3D objects, to indicate how the building should be constructed and not on the correct generation of the analytical lines. Currently created analytical models cannot be directly used for structural analysis, manual adapting of the analytical model is still required, as is identified by two of the participants of the interviews. An example of a deficiency that commonly occurs in the current analytical models is included in Figure 4.2.

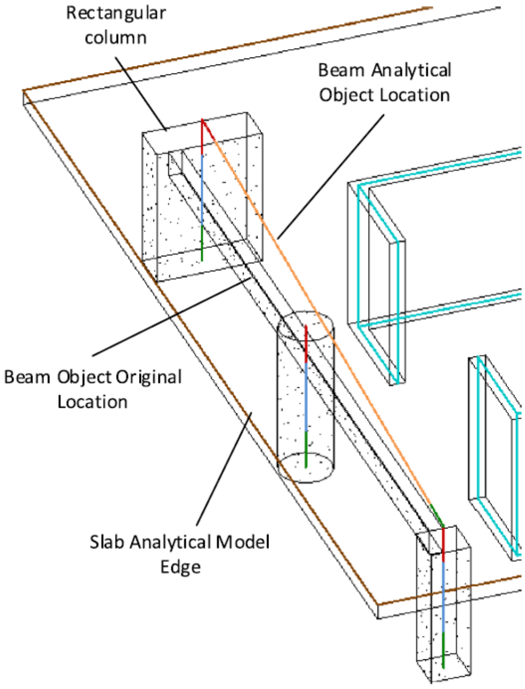

Rectangular column
Beam Analytical Object Location
Beam Object Original Location
Slab Analytical Model Edge

Figure 4.2: Deficiency in the analytical model of a beam (Hasan et al., 2019)

Missing information

All information that is needed for the calculation should be implemented in the model. However, as two of the participants of the interviews point out, it might occur that some information is missing. The tool should be able to handle such a situation and point out to the user what information is missing.

Division of responsibilities

To prevent that information is not included or incorrect information is added to the model, it should be clearly defined who is responsible for the implementation of what information in the model and at which moments in the process.

Scope

Four of the participants of the interviews mention the importance of a good scope to clearly define for what use cases the framework and tool are meant. The tool should be able to handle all input within this scope.

4.2. Features for a vertical load calculation tool

4.2.1. Characteristic features of the vertical load calculation

Repetition

As is explicitly mentioned by four of the participants of the interview, the vertical load calculation is characterised by its repetitive and iterative nature. The actual process of the vertical load calculation is iterative, as the definition of a schema is an iterative process. Furthermore, because in case a change is made to the design, the schematisation has to be adjusted accordingly. In addition, there is a certain amount of repetition within the project, as there exist multiple repetitions within buildings themselves: walls or plans that are more or less similar.

Documentation design choices

An important part of any calculation that also applies to the vertical load calculation is the documentation of design choices. It helps other engineers understand the reasoning behind the calculations. Therefore, the design choices of the structural engineers should be implemented in the BIM model.

Multiple possible schematisations

As was mentioned by two structural engineers in the interview there does not exist such a thing as a perfect schematisation. There are multiple correct schematisation possibilities.

4.2.2. Desired features for the automation of the vertical load calculation

User interactivity

Since the structural engineer is responsible for the outcome of the calculations, he or she want to be in control over the process. Two structural engineers that participated in the interviews acknowledge this, as they mention they prefer to have control over the important choices in the design process. This was also found by Mora et al. (2008), who accounted for this in the automation tool they developed by providing the structural engineer with the choice between multiple pre-defined algorithms that could be used to perform certain operations automatically.

Organisational structure

One of the main advantages of automation, as mentioned in Subsection 1.2, is that it helps to improve the organisation of a process. One of the participants of the interviews thinks the calculation is currently already set up in an efficient way. Two other participants however, think the organisation can still be improved. Therefore, a good description of the process using a flowchart can help to improve the organisational structure of the vertical load calculation. Furthermore, it would also help to explain the procedure to unexperienced structural engineers that are not familiar with the process.

Providing output in multiple ways

As the vertical calculation is also used for other more detailed calculations, like the calculation of wall reinforcement for example, it would be a useful feature for the automation tool to provide output in multiple ways. This is acknowledged by two of the structural engineers in the interviews.

4.3. List of requirements

The following list of requirements can be specified based on the limitations and functionalities of BIM-based automation of the vertical load calculation in Sections 4.1 and 4.2. The requirements are classified into three groups. Both groups and requirements are sorted according to their priorities (highest priority first).

Table 4.1: List of requirements

No.	Requirement
1	*Definition of the method*
1.1	A clear scope must be defined to indicate what use cases the method applies to
1.2	Each of the steps must be clearly defined, this involves a specification of input, output and performed operations for all process steps
1.3	Responsibilities must be assigned to each of the operations to make sure all relevant information is included in the model when needed
2	*User interaction*
2.1	User interactivity must be provided for (not fully automated)
2.2	Design choices must be documented & incorporated
2.3	It must be possible to adjust the analytical model if needed
3	*Usability*
3.1	The tool must be able to handle all input that fits the scope
3.2	The tool must be able to handle missing information
3.3	It must be possible to take repetition into account
3.4	The tool must be compatible with multiple software packages
3.5	It must be possible to provide output in multiple ways

4.4. Discussion

In this chapter the reference tools and outcomes of the interviews were investigated. This resulted in a list of limitations and challenges of the use of BIM for the automation of the vertical load calculation, as well as a list of desired features. In order to overcome the challenges and limitations and to be able to implement the necessary features a list of requirements was made. The requirements in this list are categorised into three groups:

- Definition of the method
- User interaction
- Usability

The requirements related to the definition of the method have the highest priority. These requirements must assure a clear understanding of the different steps and responsibilities defined in the proposed method. The user interaction requirements must assure the structural engineer remains in control of the process: it must be possible to make design choices throughout the process The usability requirements must assure the tool is working correctly, while also being easy-to-use and insightful.

In the next chapters the framework is developed on the basis of the requirements formulated in this chapter.

5

Concept framework

In this chapter a concept framework for the automation of the vertical load calculation is developed. This concept framework explains the general idea behind the proposed method. First, the chosen approach is described. After that, the concept framework is set up. This is done using a number of flowcharts to assure a clear description of the different steps in the vertical load calculation.

5.1. Approach

5.1.1. Based on the current organisation

The framework will be based on the current organisation of the vertical load calculation, as the current organisation follows a clear structure that allows for elaboration where needed to be able to overcome the representation difference between BIM and schematisation. Basing the framework on the current organisation, will also allow for easier implementation in practice, as the structural engineers will recognise the different steps of the process. However, there are multiple approaches to performing a vertical load calculation, making it difficult to formulate a framework that is generally applicable. The general idea behind the different approaches can considered to be similar to the way the vertical load calculation is set up at CAE Nederland (Section 3.1), as followed from the interviews with external structural engineers (Appendix A). In Subsection 3.2.2 it is explained that the main difference between companies is the way the building is divided into parts that are considered separately: vertically or horizontally. The concept framework as described in this chapter is therefore developed considering both of these approaches.

5.1.2. Automated parts

An important requirement for the usability of an automation tool that followed from both literature and interviews is to allow for user interactivity (requirements 2.1-2.3). This is the main reason why it is decided to divide the automation is several parts. Furthermore, it helps to provide insight to the user and prevent the automation tool from turning into a black box. For each of the automated parts, the user can decide if the output the tool provides needs to be adjusted, or if the tool can proceed to the next automated part. This principle is illustrated in Figure 5.1 below.

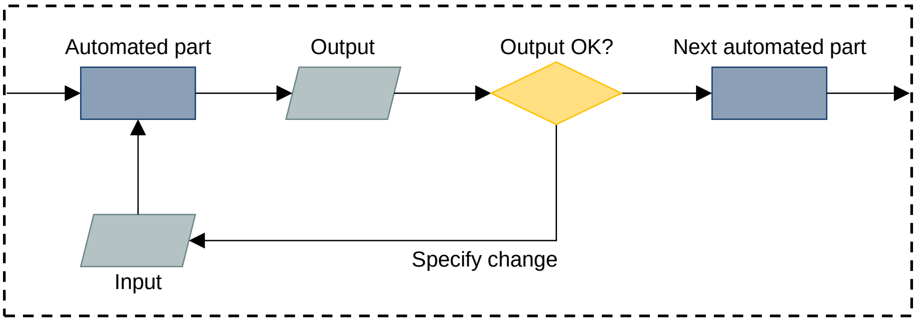


Figure 5.1: Principle of automated parts

5.2. High-level design

The current organisation of the vertical load calculation as captured in the flowchart in Figure 3.1 forms the basis for the framework for the automation of the vertical load calculation. In Figure 5.2 the corresponding flowchart is shown. From this figure it can be seen that compared to the current situation, more information is retrieved from and implemented back into the BIM model (as depicted by the red lines).

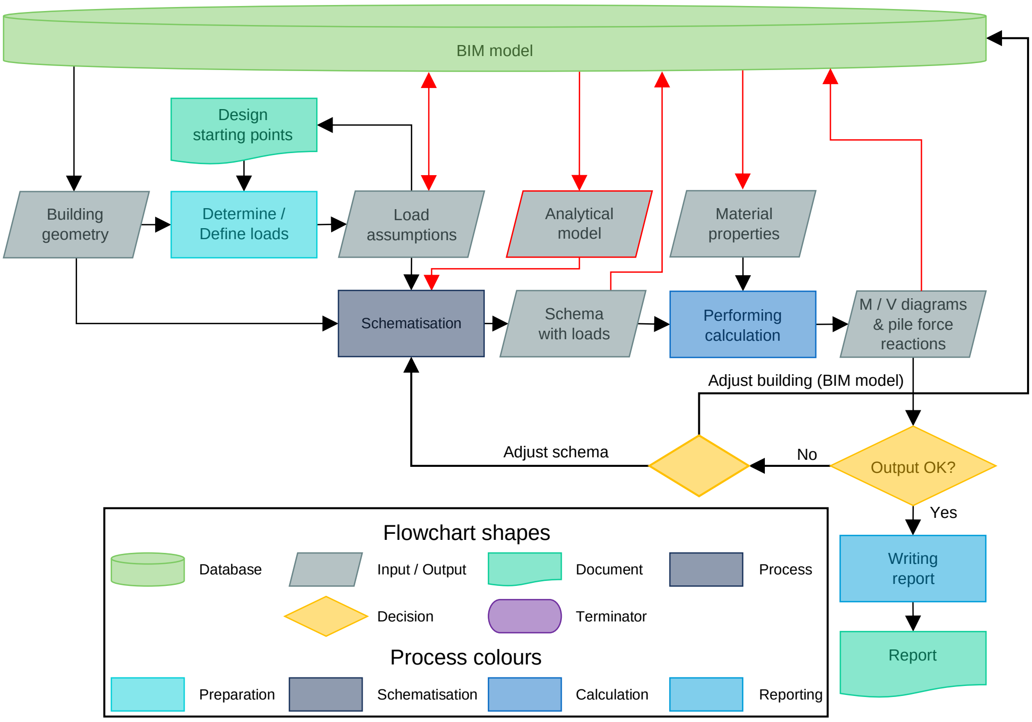


Figure 5.2: Flowchart of the assisted process (the additional links with the BIM model are indicated in red)

5.2.1. BIM model

In the flowchart in Figure 5.2 the BIM model is displayed as a database, because it will function as such in the framework. The BIM model considered in the framework, is the model made by the structural engineering firm, meaning it only contains structural elements. The architectural BIM model is not considered in the framework. However, both obviously depend on each other, as has been explained in Subsection 3.1.2. Over the course of the calculation process, information relevant for the vertical load calculation will be added to this model. In this subsection it is specified what information is expected to be present in the model at the start of the process depicted in Figure 5.2.

The geometry of the structural elements needs to be present to be able to define the relations between the different elements in the model, as well as to determine the self-weight of those elements. Therefore, also the type of material is needed. In order to be able to perform the calculation, the strength class is required. An example of the information that needs to be present at the start of the vertical load calculation is included Figure 5.3. All other relevant information will be implemented in the BIM model over the course of the vertical load calculation.

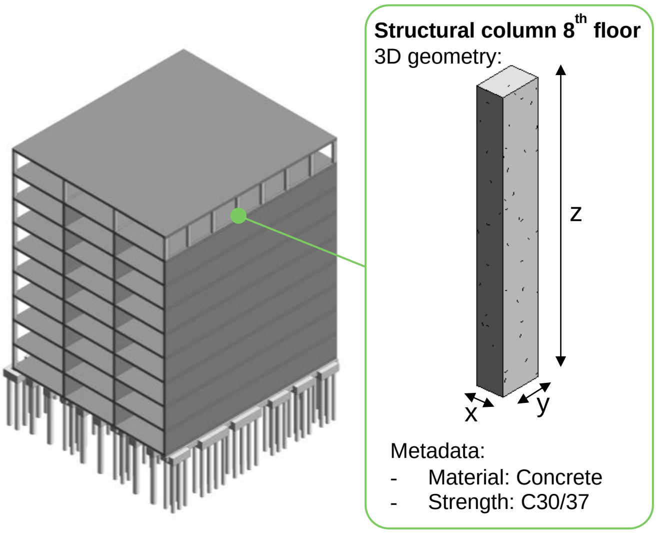


Figure 5.3: Example of an element in the BIM model with the required input assigned

5.2.2. Determine / Define loads

As explained in Subsection 3.1.1 the first step of the vertical load calculation is to identify what loads are acting on the building, leading to the definition of load packages that will be used in the vertical load calculation. The aim of this step is to assure that all loads in the load packages are implemented in the BIM model in such a way that they can be used in the schematisation process. This is also shown in the flowchart in Figure 5.4.

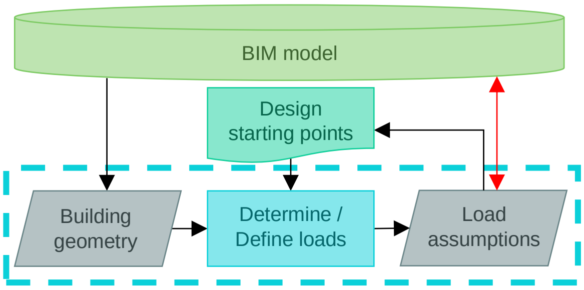


Figure 5.4: Isolated part of the flowchart displaying the 'Determine / Define loads'-process

The self-weight of all structural elements in the BIM model can be derived from the geometry of those elements. All the other permanent and variable loads that act on those structural elements, like variable loads on floors, self-weight of façades, and wind loads, have to be implemented into the BIM model by the structural engineer or the BIM modeller.

By implementing those loads using separate elements that are drawn on top of the structural element they act upon, as indicated in Figure 5.5, it can be visualised what loads are assumed to be acting on what part of the building. Furthermore, the geometry and metadata of those elements can be used to derive the loads acting on the schema, in the *'Loads Definition'*-stage of the schematisation process.

The metadata of these load elements should at least contain information about:

- The type of load:
  - Permanent / Variable;
  - Area / Line / Point;
- The magnitude;
- The orientation;
- And the units.

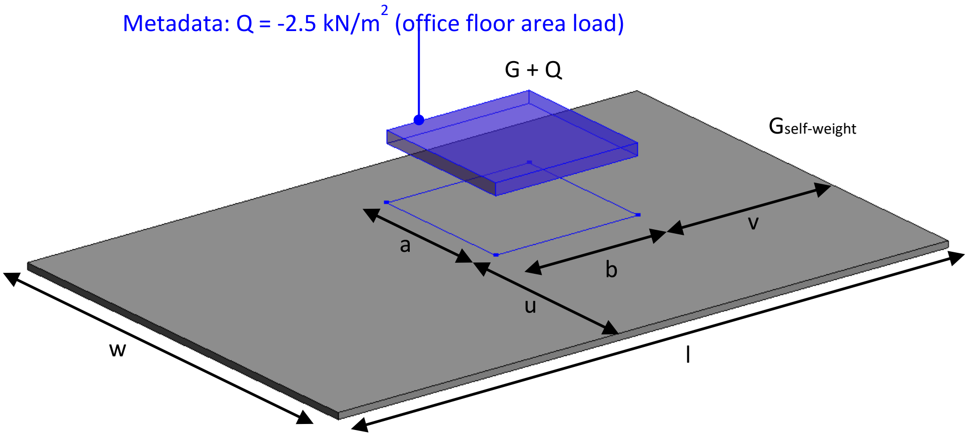


Figure 5.5: Implementing the loads acting on the structural elements into the BIM model

5.2.3. Schematisation

The schematisation process of the vertical load calculation can be subdivided into multiple sub-processes, as can be seen from Figure 5.6, in which the current organisation of the schematisation process is shown. To be able to partially automate this process using a BIM model this flowchart is adjusted to the flowchart shown in Figure 5.7. The additional links between the different steps of the schematisation process and the BIM model are indicated in red. Furthermore, the '*Adjusting Analytical Model*'-step is added to the process to be able to use the analytical model of the BIM model in the vertical load calculation (thereby fulfilling requirement 2.2, see Section 4.3). In the following each of the steps will be discussed in more detail.

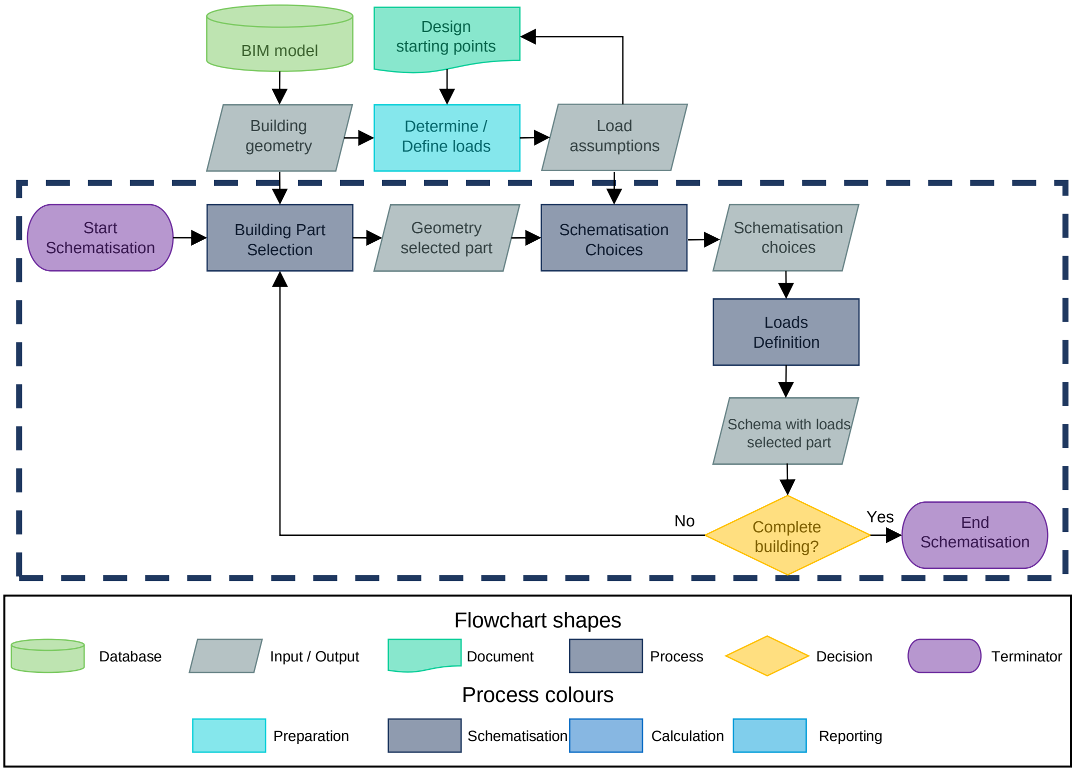


Figure 5.6: Isolated part of the flowchart of the current organisation of the schematisation process

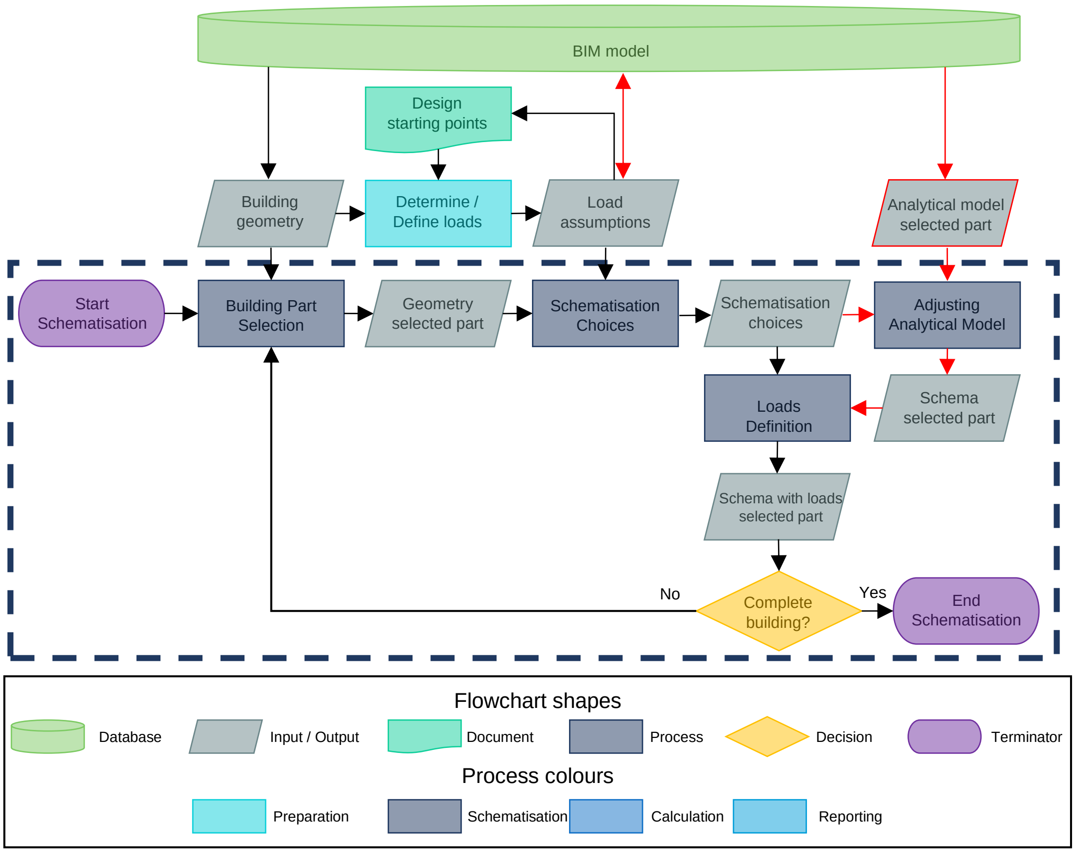


Figure 5.7: Isolated part of the flowchart of the assisted schematisation process

Building Part Selection

In a vertical load calculation the building is most often divided into separate parts, which are schematised separately, as explained in Subsection 3.1.1. The aim of this step is to choose a building part, after which the elements in this part are isolated from the rest of the BIM model. By doing that the geometry and metadata of those elements can be used for the derivation of the schema of the chosen part. This is also clarified in the flowchart given in Figure 5.8.

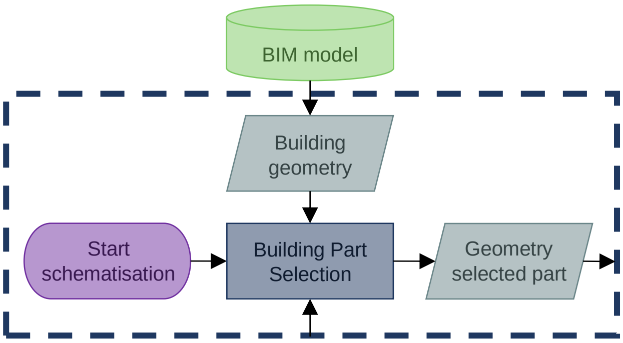


Figure 5.8: Isolated part of the flowchart displaying the 'Building Part Selection'-process

To be able to do this, the structural engineer or the BIM modeller should indicate in the BIM model which different parts are considered. Those parts can either be focused on the load-bearing walls or on the floors of the building, depending on the chosen approach of the vertical load calculation. In Figure 3.5 an example is given of how a building part can be visualised for both approaches.

Schematisation Choices

The aim of this step is to specify and document the structural engineers design choices for the considered part as chosen in the previous step. Based on the geometry and the assumed loads the structural engineer determines how this considered part of the building will be schematised (Figure 5.9). Depending on the different type of elements in the considered part, as well as the loads acting upon them, it is determined how loads will be transferred to the foundation. Figure 5.10 gives an example of the different type of elements that can be found in a part of the building for both of the approaches to performing a vertical load calculation.

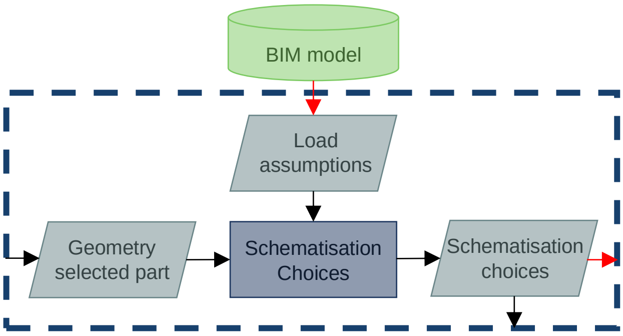


Figure 5.9: Isolated part of the flowchart displaying the 'Schematisation Choices'-process

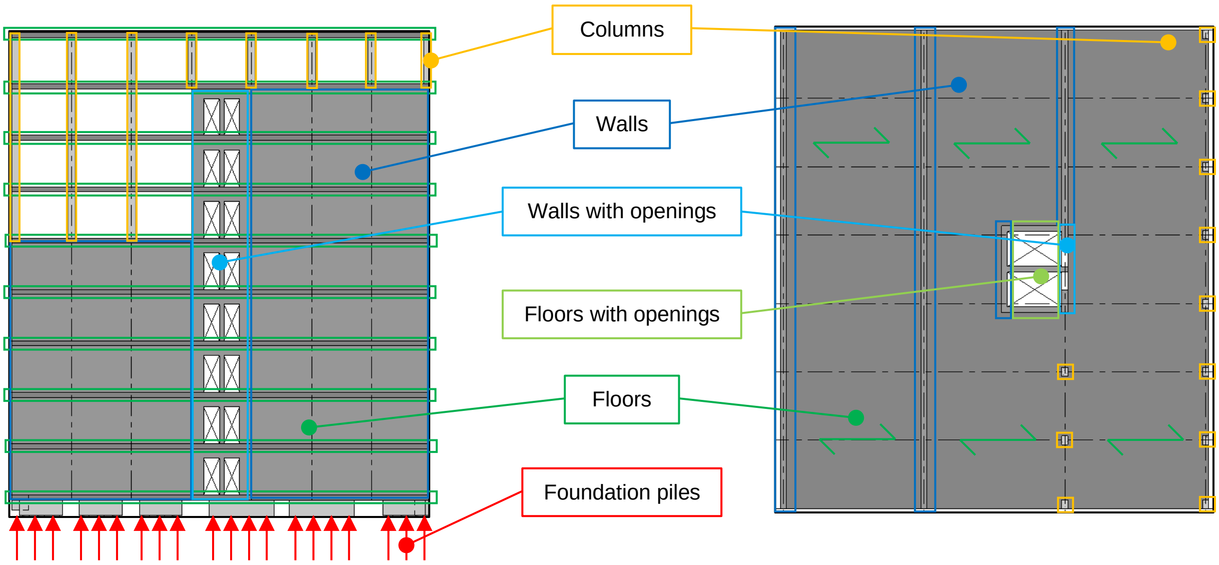


Figure 5.10: Investigation of the different types of elements in a building part

The structural engineer specifies how the different elements are used in the schema. Elements can be identified as:

- '*Schema'*-elements
- '*Support'*-elements
- '*AsLoad'*-elements
- '*Neglected'*-elements

The '*Schema'*-elements are used to derive the schema from their analytical model. The *'Support'*-elements are used to derive the location and stiffness of the supports. The *'AsLoad'*-elements act upon the *'Schema*'-elements and distribute all loads acting on them to the *'Schema'*-elements. The *'Neglected*'-elements are irrelevant for the vertical load calculation and are therefore neglected.

The output of this step is a specification of design choices, that can be visualised by colouring the elements in the building part according to the way they are considered in the derivation of the schema. Additionally, the structural engineer can add messages to the elements in the BIM model about checks or remarks about elements that need special attention in the remainder of the vertical load calculation (requirement 2.2).

Adjusting Analytical Model

The aim of this step is to assure the correct generation of the schema based on the analytical model of the elements that are specified as *'Schema'*-elements in the previous step. This results in a schema that can be used in the calculation. This is shown in the flowchart in Figure 5.11.

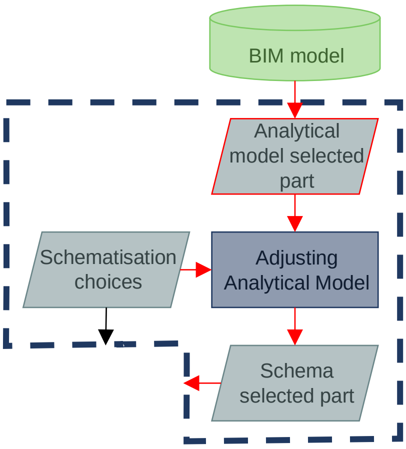


Figure 5.11: Isolated part of the flowchart showing the input and output of the 'Adjusting Analytical Model'-process

As mentioned in Subsection 4.1.3, the generation of the analytical model in BIM is not always correct. Moreover, there is an important representation difference between BIM model and schematisation, meaning that even if the analytical model would be generated correctly, that model could still differ from the schematisation. Therefore, the structural engineer will be provided with the choice between pre-defined adjustment scenarios that can be used to adjust the analytical model according to the schematisation choices made in the previous step (thereby satisfying requirement 2.2 & 2.3, see Section 4.3.). The position and dimensions of the analytical model elements can be adjusted. Furthermore, the configuration and the nature of these elements can be adjusted to make sure they better represent the real behaviour of the building. Furthermore, the type of connections between the elements, as well as the metadata of the elements can be adjusted.

Loads Definition

The aim of the last step of the schematisation process is to define the loads acting on the schema, based on the schematisation choices and the adjusted schema that are derived in the previous step of the schematisation process. In Figure 5.12 the flowchart of the step of the process is shown.

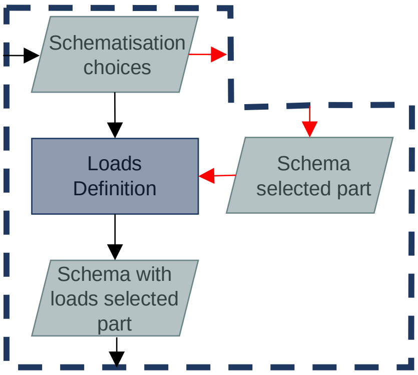


Figure 5.12: Isolated part of the flowchart displaying the 'Loads Definitions'-process

Based on the geometry and metadata of the *'Schema'*- and *'AsLoad'*-elements as defined in the *'Schematisation Choices'*-process, it can be determined what floors (or other *AsLoad'*-elements) distribute what part of the loads acting on them to the schema. This results in a schema with loads that can be used in the calculation. To assure user interactivity (requirement 2.1-2.3), the structural engineer can change the way the loads are derived by choosing between a number of pre-defined adjustment scenarios. This includes adjustments to the way openings are taken into account as well as the size of the floor areas considered to calculate the loads acting on the schema.

5.2.4. Calculation

The main focus of the framework is on the derivation of the schema. However, the calculation is an important step in the vertical load calculation, as it serves to satisfy the purpose of the vertical load calculation by finding the pile reaction forces which can be used to find the number of piles needed. In Figure 5.13 the flowchart of the calculation process is shown.

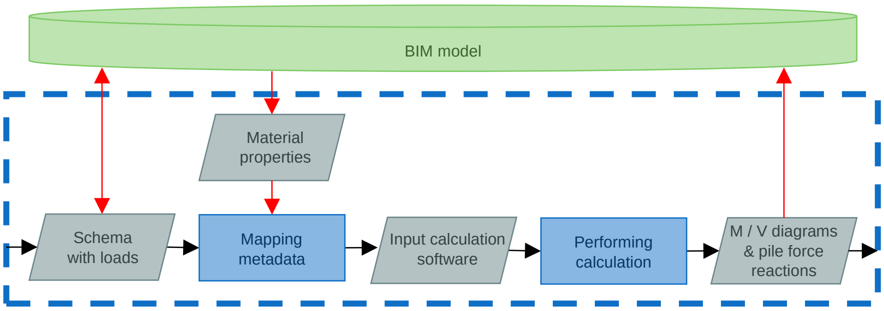


Figure 5.13: Isolated part of the flowchart displaying the 'Calculation'-process

In order to perform the calculation, the data in the BIM model is adjusted in such a way that it can be inputted into the structural analysis software that will be used in the calculation. This is done in the process-step *'Mapping metadata'*, by which it is assured that mismatching of BIM-parameters and the parameters of the chosen structural analysis software is prevented. In this framework, no specific structural analysis software is chosen, the above described approach can be applied to any structural analysis software that is compatible with BIM software.

5.3. Discussion

In this chapter the general idea behind the method that strives to (partially) automate the vertical load calculation is explained. The developed framework is based on the current organisation of the vertical load calculation. The framework consists of automated parts that are alternated with decision points that allow the structural engineer to interact with the automation tool, thereby fulfilling the requirements related to user interaction as defined in Section 4.3.

In the *'Building Part Selection'*-step it can be defined how the building will be divided into several parts that are considered separately in the vertical load calculation. The structural engineer can specify how the elements in the BIM model should be used in the derivation of the schema in the *'Schematisation Choices'*-step. The *'Adjust Analytical Model'*-process allows the structural engineer to make adjustments to the analytical model that is created by the BIM model. The structural engineer can overrule the automation tool in the *'Loads Definition'*-process if the way the loads are calculated is not as desired. To be able to find the pile reaction forces, the developed schema with loads is exported to structural analysis software in the *'Calculation'*-process.

The developed flowcharts give a description of the different steps in the assisted process. Since the aim for the concept framework was to give the general idea behind the proposed method, the information in this chapter is not very detailed. Therefore, the scope of the framework will be narrowed in the next chapter. By doing that, the framework can be developed in more detail, defining each of the process steps and uses cases of the framework more specific (requirement 1.1-1.3).

Part 3: Preliminary Design definition

6

Framework development

In the previous chapter the general idea of the framework is outlined. To allow for a more precise description of the different steps and the responsibilities of each of the roles involved, the scope of the framework is narrowed in this chapter. This narrowed scope is given first, after which a list of load and element types that are included into the framework is given. Finally, each of the framework steps are accommodated with a chart indicating how these steps of the framework have to be performed and who is responsible for the different tasks involved.

6.1. Scope of the framework

Using this scope one can determine if the framework can be applied to a certain use case (requirement 1.1). The framework will focus on a vertical load calculation performed per wall as this is the approach that is used by CAE Nederland. For the same reason, the framework will target the use of Revit as BIM software. Although this limits the use cases of the framework, it allows to be more precise in its description. By choosing Revit as BIM software the major part of the use cases will still be considered, as Revit is currently the most used BIM software (RIBA, 2019).

In the framework the 2D schema (see Figure 3.2) is chosen as the starting point of the derivation of the schema as done in the *'Schematisation'*-process. At the end of this process the structural engineer will be provided with the possibility to reduce the schema to the simpler 1D schema. All load types that can be inserted in the BIM model and are incorporated in the framework are given in Table 6.1. For the implementation of loads in the BIM model, the *'Analyze'*-tab of Revit has to be used (Autodesk Inc., 2020a). For each of the different loads included in Table 6.1, it is indicated which relevant metadata should be specified. Currently, only loads acting in the z-direction that orient to the global coordinate system of the BIM model ('*Orient to'* is set to *'Project'*) are considered. Table 6.2 sums up all the element types that are incorporated in the framework. This list is composed of all elements that commonly occur in mid-rise residential buildings with a load-bearing structure typically made out of concrete (to fit the scope of this MSc thesis).

Table 6.1: List of load types included in the framework

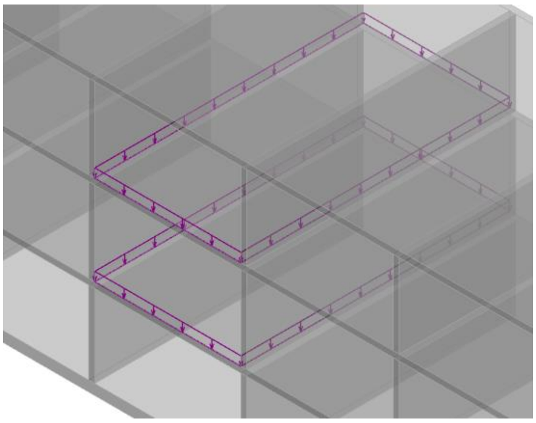
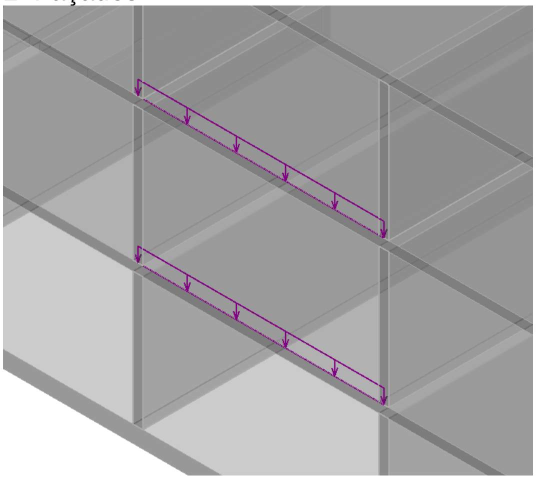
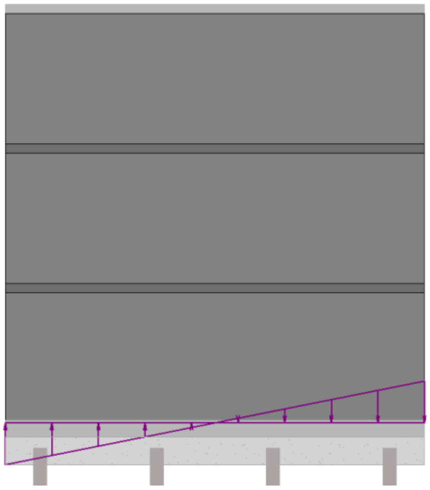
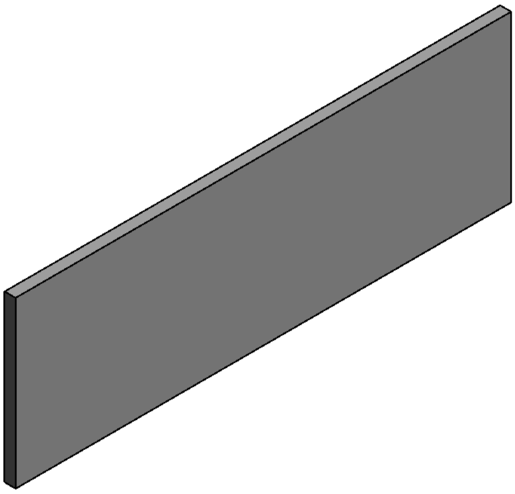
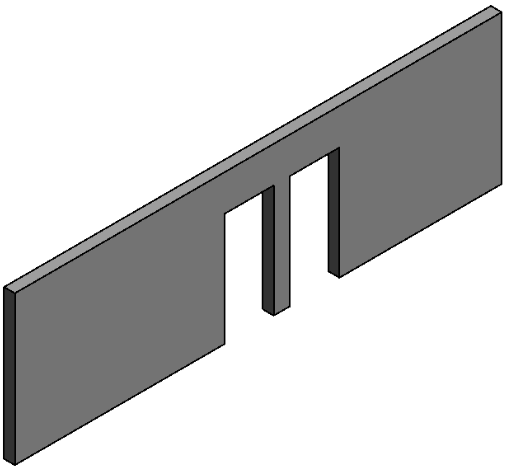
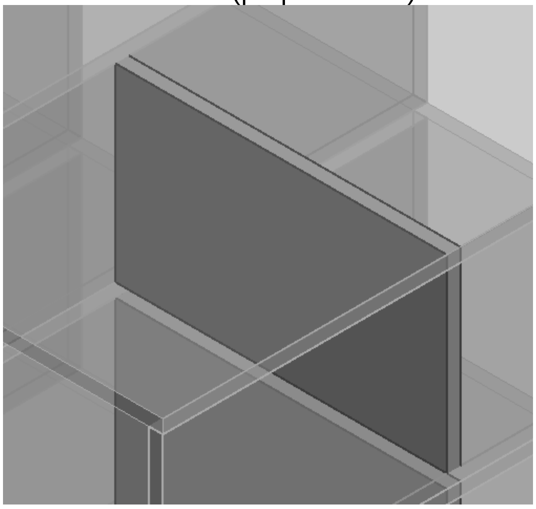
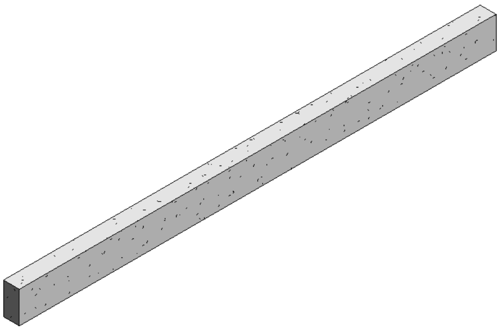
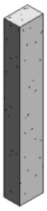
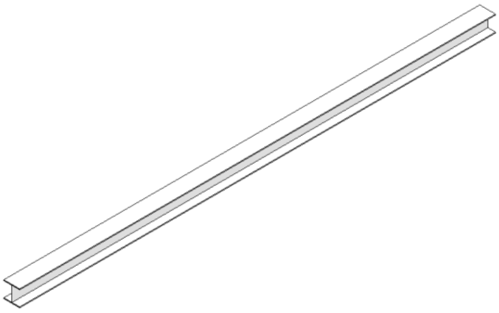
Loads	**Details**
1. Area loads 	The total area loads (excluding self-weight) can be implemented in the BIM model using the sketch-based area loads on each level - *Uniform load*: True - *Orient to:* Project - *Fz 1*: Total load in z-direction [kN/m^2] (upward is positive) - *Description*: The permanent and variable part of the load can be indicated: G_*valueG*_Q_*valueQ* - *Comments*: Additional comments about the nature of the area load Different types of area loads can be distinguished in the BIM model by colouring them differently
2. Façades 	The self-weight of façades can be implemented in the BIM model using sketch-based line loads on each floor level or added up at foundation level [kN/m] - *Uniform load*: True - *Orient to:* Project - *Fz 1*: Total load in z-direction [kN/m] (upward is positive) - *Description*: The amount of floors that are represented by the line load can be indicated: *n_floors_value* - *Comments*: Additional comments about the nature of the line load Different types of line loads can be distinguished in the BIM model by colouring them differently
3. Resulting stresses from wind loads 	The resultants from the wind loads as follow from the stability calculation can be implemented in the BIM model using sketch-based line loads on each floor level or added up at the foundation level [kN/m] - *Uniform Load:* False - *Fz 1*: Total load in z-direction [kN/m] at the first end-point of the line load (upward is positive) - *Fz 2:* Total load in z-direction [kN/m] at the second end-point of the line load (upward is positive) - *Description*: Part of the building represented by the wind load *n_floors_wall_width* - *Comments*: Additional comments about the nature of the line load Different types of line loads can be distinguished in the BIM model by colouring them differently

Table 6.2: List of element types that are included in the framework

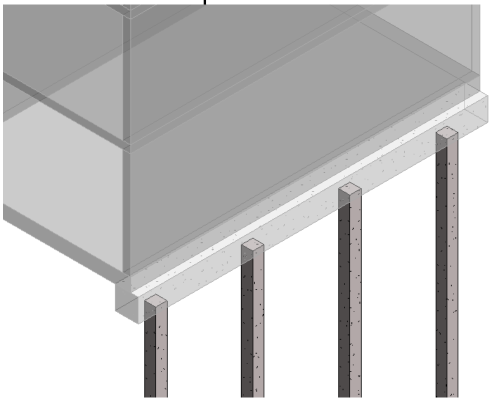
Elements	Details
1.1. Concrete wall 	Structural wall (planar analytical model)
1.2. Concrete wall (small width) 	Structural wall (width ≤ 400 mm) (planar analytical model) - According to clause 5.3.1(7) in Eurocode 2, a wall may be calculated as a column if the longest side of the cross-section is smaller than four times the smallest side, and if the height of the wall is more than three times larger than the longest side of the cross-section (Nederlands Normalisatie-Instituut, 2011).
1.3. Concrete wall (with openings) 	Structural wall with openings (modelled using ‘opening’-elements) (planar analytical model)

1.4. Concrete wall (perpendicular) 	Structural wall perpendicular to the main load-bearing wall (planar analytical model)
2.1. Concrete beam 	Structural framing (linear analytical model)
2.2. Concrete column 	Structural columns (linear analytical model)
2.3. Steel beam 	Structural framing (linear analytical model)

2.4. Steel column

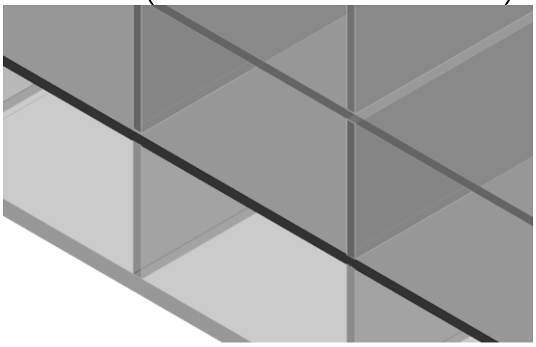
Structural columns (linear analytical model)

2.5. Foundation piles



Structural foundations

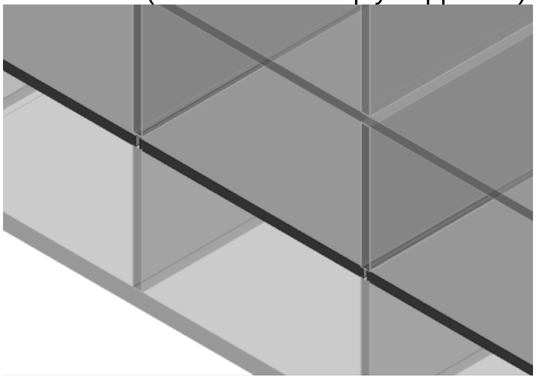
3.1A. Floor (modelled as continuous floor)



One way spanning structural floors modelled as continuous slabs

Based on the type of the floor can be calculated as continuous (cast in-situ, etc.) or as simply supported (hollow core, etc.)

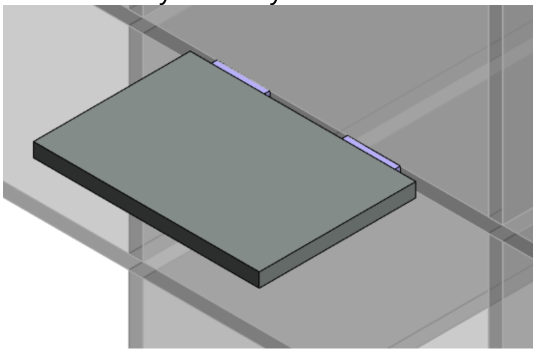
3.1B. Floor (modelled as simply supported)



One way spanning structural floors modelled as simply supported slabs

Always calculated as simply supported slabs

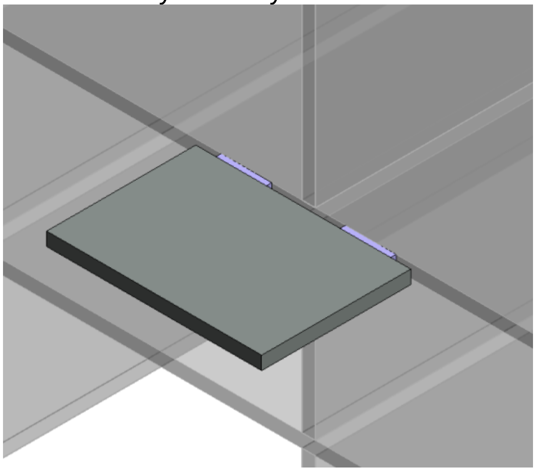
3.2A. Balcony / Gallery connected to the floor at the middle of the span



Cantilevering balcony or gallery modelled as a structural floor connected to the middle of the floor using thermal break elements

When classified as ‘*AsLoad*’-element calculated as local bending moment plus point load

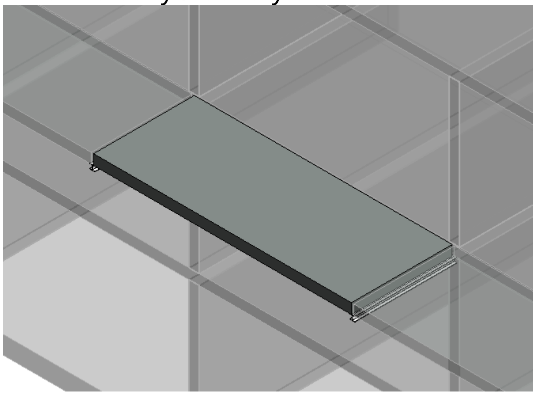
3.2B. Balcony / Gallery connected to the floor at load-bearing wall



Cantilevering balcony or gallery modelled as a structural floor connected to the floor at the load-bearing wall using thermal break elements

When classified as ‘*AsLoad*’-element calculated as local bending moment plus point load

3.2C. Balcony / Gallery connected to the load-bearing walls by structural beams



Cantilevering balcony or gallery modelled as a structural floor supported structural framing elements that are connected to the main load-bearing structure

When classified as ‘*AsLoad*’-element calculated as local bending moment plus point load

6.2. Determine / Define loads

The general idea behind the '*Determine / Define Loads*'-process is explained in Subsection 5.2.2 and Figure 5.4. In Figure 6.1 below, it is shown who is responsible for what operation in the process of determining and defining the loads acting on the building. The structural engineer determines the loads acting on the building, and subsequently instructs the BIM modeller on which elements the different loads act. The BIM modeller can insert those loads on the specified elements using the sketch-based load elements in Revit (Autodesk Inc., 2020a).

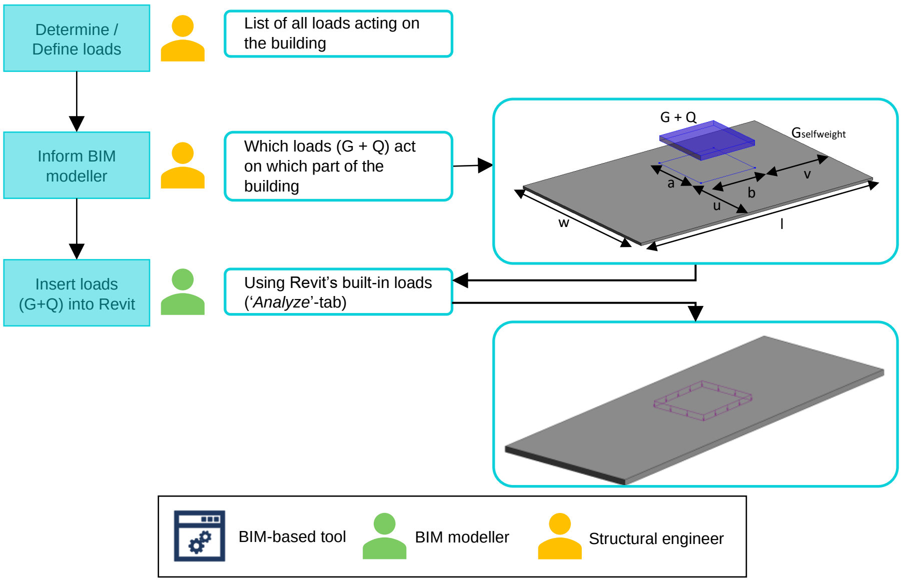

Determine / Define loads
List of all loads acting on the building
Inform BIM modeller
Which loads (G + Q) act on which part of the building
G + Q
Gselfweight
a
b
u
v
w
l
Insert loads (G+Q) into Revit
Using Revit's built-in loads ('Analyze'-tab)
BIM-based tool
BIM modeller
Structural engineer

Figure 6.1: Flowchart indicating the consecutive operations and corresponding responsibilities of the 'Determine / Define loads' process

Using this principle it can be visualised in the BIM model which area loads act on what floor area, how the façades are supported and where stability loads are taken up by the load-bearing structure (see Table 6.1).

6.3. Schematisation

6.3.1. Building part selection

The general idea behind the ‘*Building Part* Selection’-process is explained in 5.2.3 and Figure 5.8. The operations and responsibilities involved are shown in Figure 6.2.

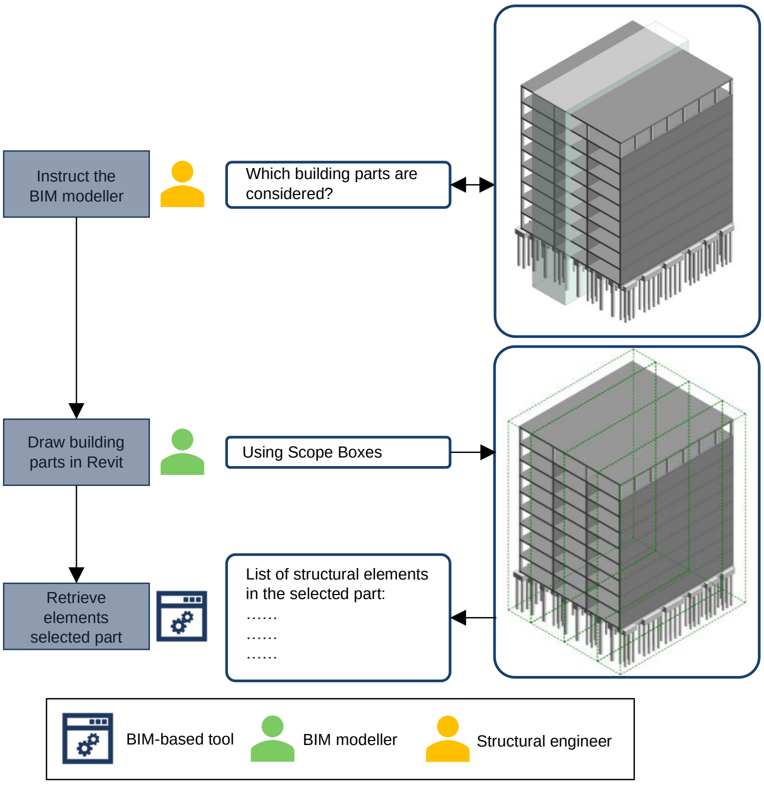
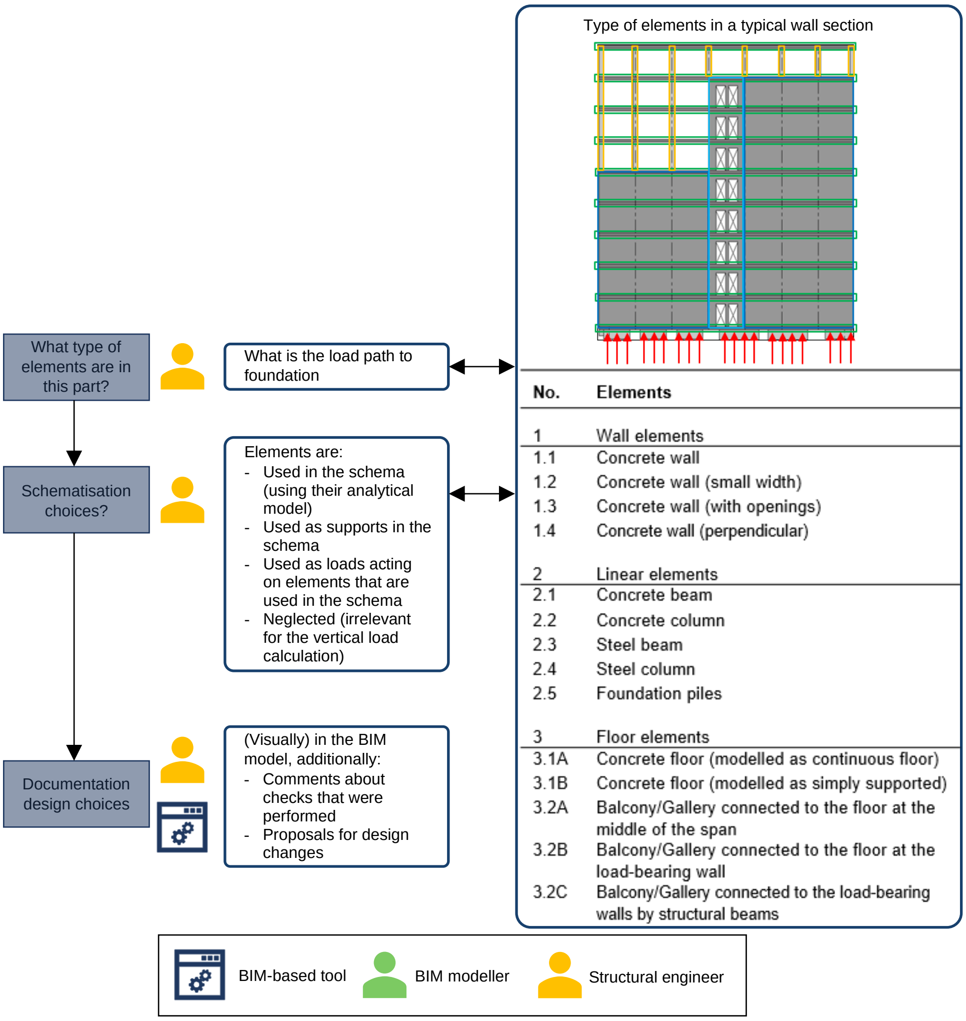

Instruct the BIM modeller
Which building parts are considered?
Draw building parts in Revit
Using Scope Boxes
Retrieve elements selected part
List of structural elements in the selected part:
……
……
……
BIM-based tool
BIM modeller
Structural engineer

Figure 6.2: Flowchart indicating the consecutive operations and corresponding responsibilities of the 'Building Part Selection'-process

The structural engineers has to determine what parts of the building are considered and instruct the BIM modeller about this. In case of a vertical load calculation performed per wall, this means defining which walls will be calculated. The BIM modeller can than draw scope boxes around those walls, which the tool can use to isolate the structural elements in the selected building part (by intersecting all structural elements in the BIM model with each of the scope box-elements), this results in a list of elements per building part.

6.3.2. Schematisation choices

The general idea behind the '*Schematisation Choices*'-process is explained in 5.2.3 and Figure 5.9. In Figure 6.3 the consecutive operations that have to be performed by the structural engineer to implement the design choices in the BIM model are displayed.


Type of elements in a typical wall section
What type of elements are in this part?
What is the load path to foundation
Schematisation choices?
Elements are:
- Used in the schema (using their analytical model)
- Used as supports in the schema
- Used as loads acting on elements that are used in the schema
- Neglected (irrelevant for the vertical load calculation)
Documentation design choices
(Visually) in the BIM model, additionally:
- Comments about checks that were performed
- Proposals for design changes
BIM-based tool
BIM modeller
Structural engineer

No.	Elements
1	Wall elements
1.1	Concrete wall
1.2	Concrete wall (small width)
1.3	Concrete wall (with openings)
1.4	Concrete wall (perpendicular)
2	Linear elements
2.1	Concrete beam
2.2	Concrete column
2.3	Steel beam
2.4	Steel column
2.5	Foundation piles
3	Floor elements
3.1A	Concrete floor (modelled as continuous floor)
3.1B	Concrete floor (modelled as simply supported)
3.2A	Balcony/Gallery connected to the floor at the middle of the span
3.2B	Balcony/Gallery connected to the floor at the load-bearing wall
3.2C	Balcony/Gallery connected to the load-bearing walls by structural beams

Figure 6.3: Flowchart indicating the consecutive operations and corresponding responsibilities of the 'Schematisation Choices'-process

The structural engineer has to determine which elements are considered as: '*Schema'*-, '*Support'*-, '*AsLoad*'- or '*Neglected'*-element in the remainder of the schematisation process. This is indicated in the BIM model by using different colours for each of the abovementioned categories. Additionally, the structural engineer can add comments to the metadata of single elements to document small checks that were performed or proposals for design changes.

6.3.3. Adjusting the analytical model

In Subsection 5.2.3 and Figure 5.11 the general idea behind the *'Adjusting Analytical Model'*-process is explained. The consecutive operations and responsibilities that are involved in this process are elaborated in Figure 6.4 below (requirement 1.2-1.3).

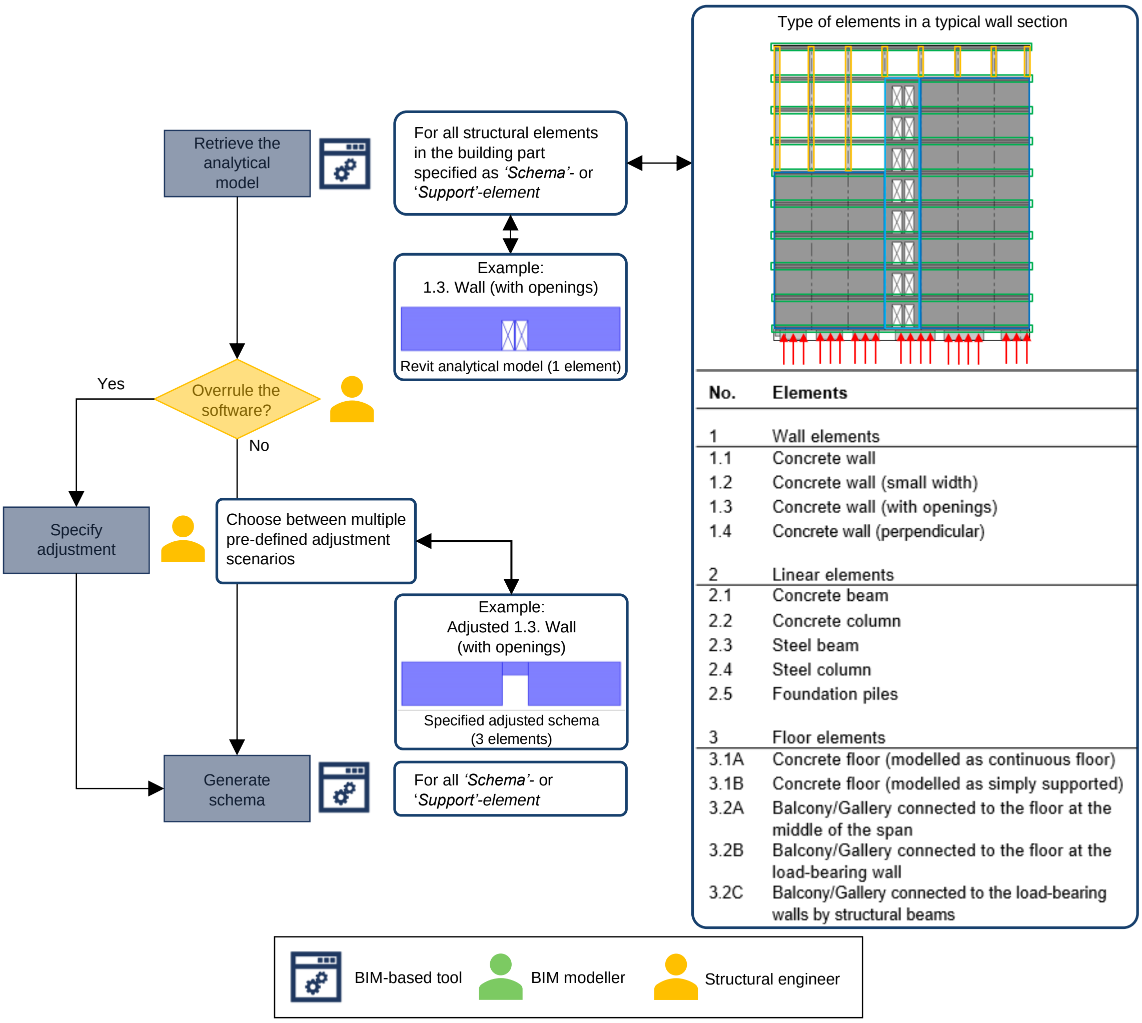
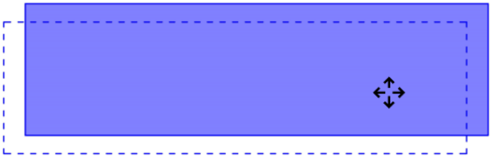

Retrieve the analytical model
For all structural elements in the building part specified as 'Schema'- or 'Support'-element
Example:
1.3. Wall (with openings)
Revit analytical model (1 element)
Yes
Overrule the software?
No
Specify adjustment
Choose between multiple pre-defined adjustment scenarios
Example:
Adjusted 1.3. Wall (with openings)
Specified adjusted schema (3 elements)
Generate schema
For all 'Schema'- or 'Support'-element
Type of elements in a typical wall section
No. Elements
1 Wall elements
1.1 Concrete wall
1.2 Concrete wall (small width)
1.3 Concrete wall (with openings)
1.4 Concrete wall (perpendicular)
2 Linear elements
2.1 Concrete beam
2.2 Concrete column
2.3 Steel beam
2.4 Steel column
2.5 Foundation piles
3 Floor elements
3.1A Concrete floor (modelled as continuous floor)
3.1B Concrete floor (modelled as simply supported)
3.2A Balcony/Gallery connected to the floor at the middle of the span
3.2B Balcony/Gallery connected to the floor at the load-bearing wall
3.2C Balcony/Gallery connected to the load-bearing walls by structural beams
BIM-based tool
BIM modeller
Structural engineer

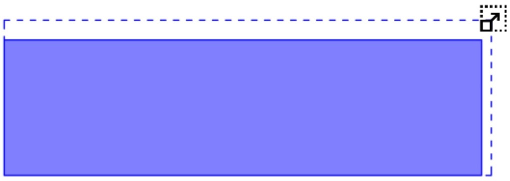
Figure 6.4: Flowchart indicating the consecutive operations and corresponding responsibilities of the 'Adjusting Analytical Model'-process

For each of the elements that are specified as *'Schema'*- or *'Support'*- element in the *'Schematisation choices'*-process the structural engineer can specify adjustments that are then applied to the schema by the BIM-based tool. All adjustment scenarios that are included in the framework are given in Table 6.3 below.

Table 6.3: Adjustment scenarios for the adjustments to the analytical model

Adjustment scenario	**Details**
Adjust wall position 	The position of the analytical model of the wall elements can be adjusted: - Applicable to wall elements 1.1-1.4

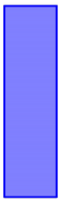
Adjust wall dimensions



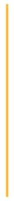
The dimensions of the analytical model of the model of the wall elements can be adjusted:

- Applicable to wall element 1.1-1.3

Adjust configuration wall element with small width



(0) Keep standard configuration (planar)

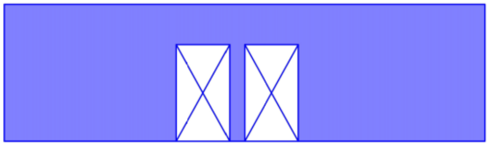


(1) Change to column (linear)

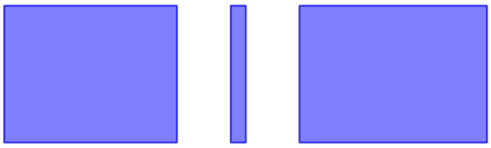
The type of analytical model of the small wall elements can be adjusted according to the scenarios indicated on the left.

- Applicable to wall element 1.2
- In case scenario (0) is chosen the loads from the element are calculated as a distributed load, in case scenario (1) is chosen the loads from the element are calculated as a point load

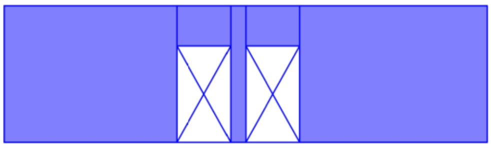
Adjust configuration wall element with openings



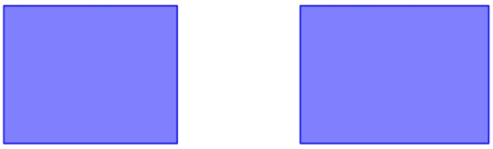
(0) Keep standard configuration



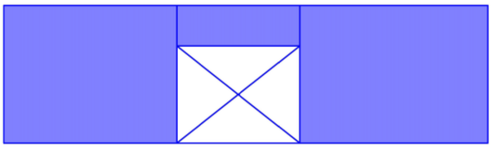
(1) Replace single wall element by separate wall segments



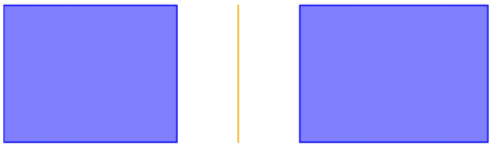
(2) Split single wall element in multiple segments



(3) Replace openings with one opening



(4) Replace openings with one opening and a lintel

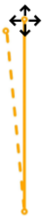


(5) Replace wall segment(s) with columns

The analytical model of the wall elements with openings can be adjusted according to the scenarios specified on the left.

- Applicable to wall element 1.3
- Scenario (3)-(5) can only be chosen in case the amount of openings greater than or equal to two

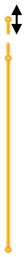
Adjust position end-points linear elements



The positions of the end-points of the analytical model of linear elements can be adjusted

- Applicable to linear element 2.1-2.4

Adjust length linear elements



The length of the analytical model of the linear elements can be adjusted

- Applicable to linear element 2.1-2.4

Adjust position supports

The position of the supports can be adjusted.

- Applicable to linear element 2.5

6.3.4. Loads definition

The general idea behind the '*Loads definition*'-process is given in Subsection 5.2.3 and Figure 5.12. In Figure 6.5 the consecutive operations and responsibilities of this process are indicated.

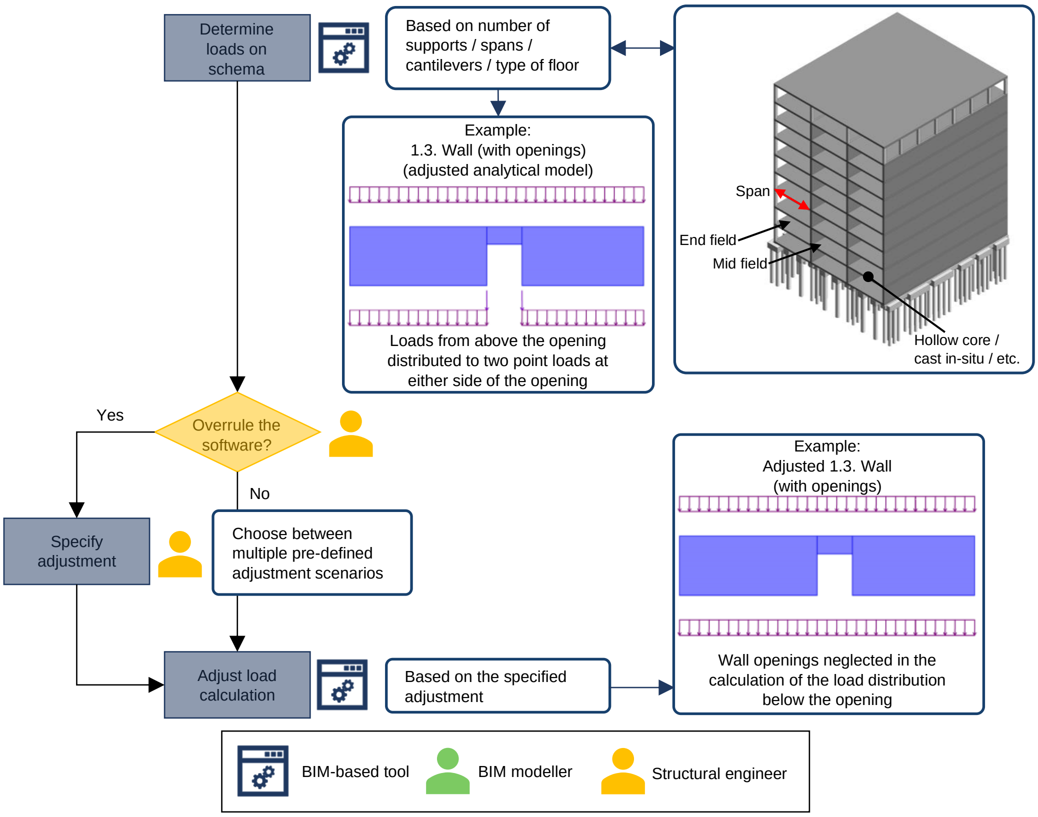
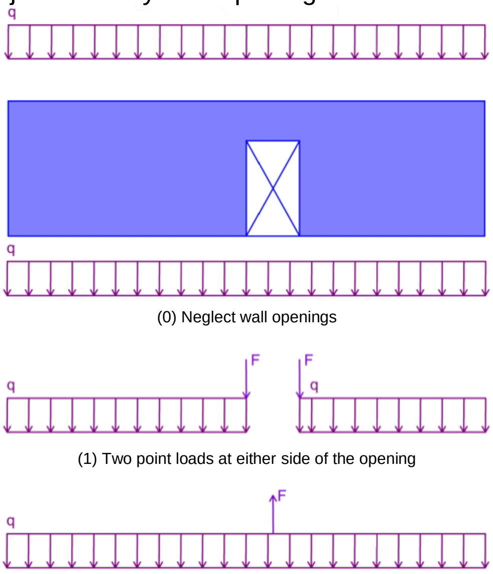
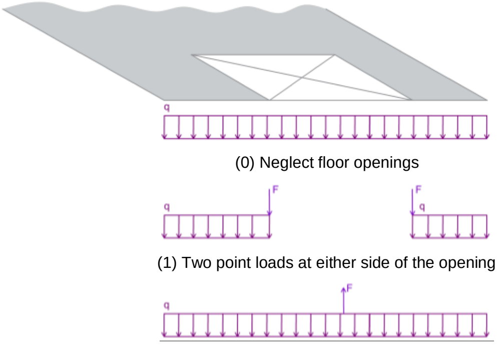
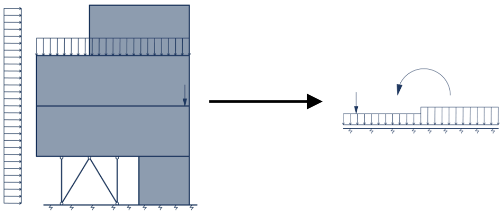

Determine loads on schema
Based on number of supports / spans / cantilevers / type of floor
Example:
1.3. Wall (with openings)
(adjusted analytical model)
Loads from above the opening distributed to two point loads at either side of the opening
Span
End field
Mid field
Hollow core / cast in-situ / etc.
Yes
Overrule the software?
No
Specify adjustment
Choose between multiple pre-defined adjustment scenarios
Example:
Adjusted 1.3. Wall
(with openings)
Wall openings neglected in the calculation of the load distribution below the opening
Adjust load calculation
Based on the specified adjustment
BIM-based tool
BIM modeller
Structural engineer

Figure 6.5: Flowchart indicating the consecutive operations and corresponding responsibilities of the 'Loads Definition'-process

The way the *'AsLoad'*-elements are incorporated in the framework is incorporated in Table 6.2. The possible adjustments that can be made to the way the loads are defined are indicated in Table 6.4 below.

Table 6.4: Adjustment scenarios for the loads definition

Adjustment scenario	Details
Adjust the way wall openings are considered  (0) Neglect wall openings (1) Two point loads at either side of the opening (2) Upward point load at the centre of the opening	- Applicable to wall element 1.3
Adjust the way floor openings are considered  (0) Neglect floor openings (1) Two point loads at either side of the opening (2) Upward point load at the centre of the opening	- Applicable to floor elements 3.1A & 3.1B in case they contain an opening
Adjust the way loads are distributed from floors to the *'Schema'*-elements	- Applicable to floor elements 3.1A & 3.1B in case they contain an opening - Standard values are: 0.4/0.6 for end-fields of continuous floors & 0.5/0.5 for all other floor fields (both continuous & simple)
Adjust the consequence class used	Standard CC2 is used
Adjust the level of the schematisation  2D schema 1D schema	Reduce the 2D schema of the wall to a 1D schema

6.4. Calculation

The general idea behind the calculation process is explained in Subsection 5.2.4 and Figure 5.13. The consecutive operations and responsibilities of the calculation process are indicated in Figure 6.6.

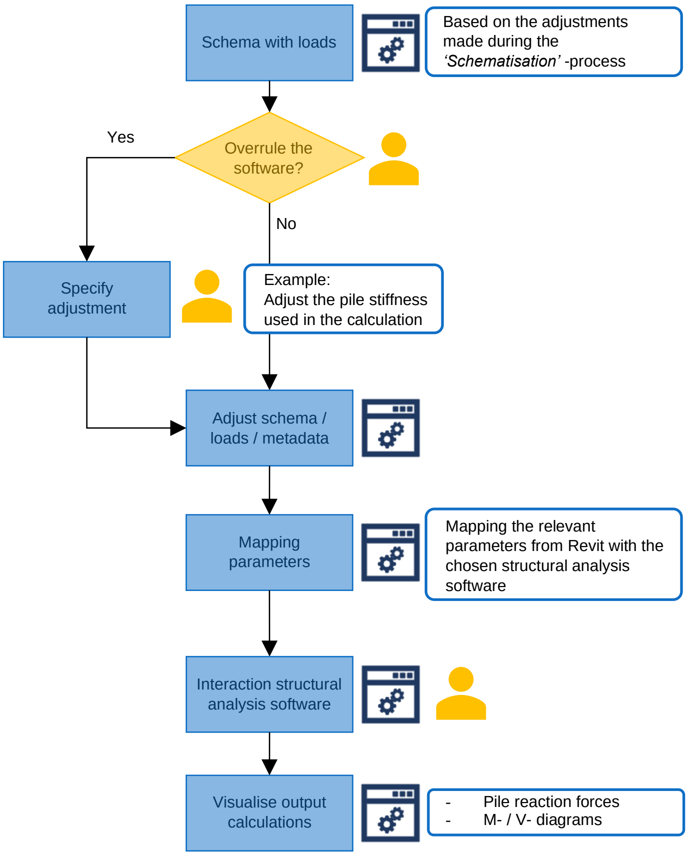

Schema with loads
Based on the adjustments made during the *'Schematisation'* -process
Yes
Overrule the software?
No
Specify adjustment
Example: Adjust the pile stiffness used in the calculation
Adjust schema / loads / metadata
Mapping parameters
Mapping the relevant parameters from Revit with the chosen structural analysis software
Interaction structural analysis software
Visualise output calculations
- Pile reaction forces
- M- / V- diagrams

Figure 6.6: Flowchart indicating the consecutive operations and corresponding responsibilities of the 'Calculation'-process

The adjustments that can be made to the schema with loads that will be used to perform the calculation is incorporated in Table 6.5.

Table 6.5: Adjustments scenarios for the calculation

Adjustment scenario	**Details**
Adjust material properties	For all *'Schema'*-elements the strength class, E-modulus, etc. can be adjusted. This adjustment scenario applies to element 1.1-1.3 & 2.1-2.4
Adjust connection type	For all *'Schema'*-elements the connections between the different elements can be adjusted: free, flexible & rigid. This adjustment scenario applies to element 1.1-1.3 & 2.1-2.4
Adjust stiffness support	For all *'Support'*-elements the stiffness can be adjusted. This adjustment scenario applies to element 2.5

6.5. Discussion

In this chapter the framework is developed in more detail. It is defined how Revit can be used to apply the general idea of the framework as developed in Chapter 5, to the approach to the vertical load calculation in which the building is divided vertically. First, the scope of the framework is defined by specifying a list of loads and a list of elements that are considered in the framework (Table 6.1 & 6.2). After that, the consequent operations of each of the process steps are specified using flowcharts. Furthermore, responsibilities are assigned to both tool and persons involved in the process. In doing so, requirement 1.1-1.3 are fulfilled. For the *'Adjust Analytical Model'*-, *'Loads Definition'*- and *'Calculation'*-processes adjustment scenarios are defined to allow the user to make adjustments to the outcomes proposed by the BIM-based tool (requirement 2.3).

To proof if the proposed method can be applied to a practical case a prototype tool will be developed in the next chapter to perform a proof of concept of the framework described in this chapter.

Part 4: Detailed Design

7

Tool development

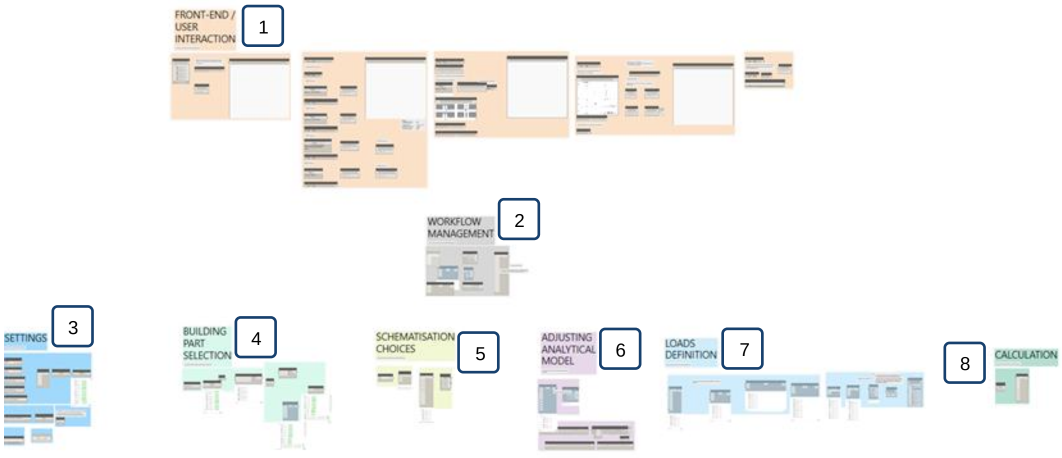
In this chapter the development of a prototype tool using Dynamo is described. This tool is used to perform a proof of concept of the method describing the automation of the vertical load calculation as developed in the previous chapter. To this purpose a number of test cases were developed and user tests were performed. Furthermore, the challenges that were faced during the development of the tool and limitations of the proposed method and tool were identified.

7.1. Developing the tool in Dynamo

As described in Section 3.4, Dynamo is a parametric design software connected to Revit. Moreover, Dynamo is a visual programming language in which a design logic can be created using components or nodes that represent certain rules and parameters. Dynamo contains a Python node that can be used for user customisation and to perform complex operations that cannot be performed using the 'standard' nodes.

7.1.1. Overview Dynamo graph

In Figure 7.1 below an overview is given of the way the Dynamo graph is built up.



(1) Front-End / User interaction
(2) Workflow Management
(3) Settings
(4) Building Part Selection

(5) Schematisation Choices
(6) Adjusting Analytical Model
(7) Loads Definition
(8) Calculation

Figure 7.1: Overview of the Dynamo graph

It contains a combination of standard Dynamo and Python nodes that are used to implement the framework. Python functions used in these nodes is organised in an external library, that contains modules for each of the steps of the framework incorporated in the tool. To improve the understandability of the graph, color-coding is used to indicate the different process steps and to make a distinction between the front-end and the back-end of the tool. Below each of the differently coloured groups is explained shortly.

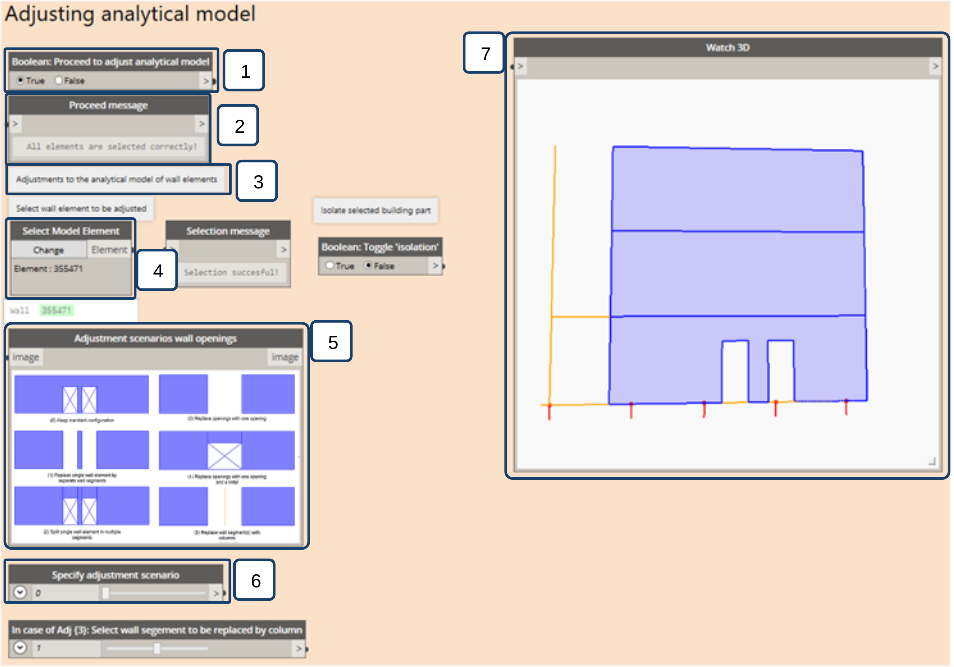
User interaction

The orange groups (1) represent the front-end of the tool. The nodes in these groups allow the user to interact with the tool. Each of the orange groups (1) indicate a different step in the process of the vertical load calculation. They resemble the steps proposed in the framework developed in Chapter 5 and clarified in the flowcharts in Figure 5.2 and Figure 5.7. The following process steps are implemented in the Dynamo tool:

- Building Part Selection
- Schematisation Choices
- Adjusting Analytical Model
- Loads Definition
- Calculation

Each of the orange groups has a similar arrangement: using the nodes on the left the user can specify his or her decisions for the considered part of the process, on the right a 3D preview of the geometry is provided, in which the effect of the choices made using the nodes on the left are visualised. The *'User interaction'*-group for the *'Adjusting Analytical Model'*-step is included in Figure 7.2.

From Figure 7.2 it can be seen that different types of nodes are used to allow for user interaction. Input nodes (1, 4, 6) can be used to indicate design choices and/or to adjust the output of the tool. Text (3) and images (5) are used to help the user make decisions. Watch nodes are used to provide the user with in-between results and visualisations, they also warn the user if a certain operation cannot be performed. In Appendix B a manual that is used in the user tests is included, this manual includes all nodes that are used for user interaction.


Adjusting analytical model
Boolean: Proceed to adjust analytical model
● True ○ False
1
Proceed message
2
Adjustments to the analytical model of wall elements
3
Select wall element to be adjusted
Select Model Element
Change
Element
4
Selection message
Isolate selected building part
Boolean: Toggle 'isolation'
○ True ● False
Adjustment scenarios wall openings
image
image
5
Specify adjustment scenario
6
In case of Adj (3): Select wall segement to be replaced by column
Watch 3D
7

All elements are selected correctly!

Element : 355471

Wall 355471

Selection succesful!

Specify adjustment scenario: 0

In case of Adj (3): Select wall segement to be replaced by column: 1

(1) Boolean node (user input)
(2) Text node
(3) Select model element node (user input)
(4) Image node
(5) Image node
(6) Number slider (user input)
(7) 3D watch node (tool output)

Figure 7.2: User interaction group for the 'Adjusting Analytical Model'-step

Graph execution management

The grey groups (2) assure the correct execution of the Dynamo graph as well as the generation of relevant 3D geometry to visualise design choices for each step of the process. In Dynamo it is not possible to partially run a graph. To be able to divide the Dynamo logic into the several parts corresponding to the different steps of the process, a '*Boolean'*-node is included at the start of each step. Using these nodes the incremental execution of the graph is managed, when the *'Boolean node'* of a certain step in the process is set to *'True'*, all nodes belonging to that step in the process are run and their output is used to generate in-between results and geometry to the user. When the *'Boolean node'* is set to *'False*, the nodes belonging to that process step are still run, however no results nor geometry are generated based on their output. The *'Boolean node'* at the start of each process step can only be set to *'True'* if the node of the previous step in the process is also set to *'True'*. This is clarified in the flowchart given in Figure 7.3, in which the logic behind the execution of the Dynamo graph is provided.

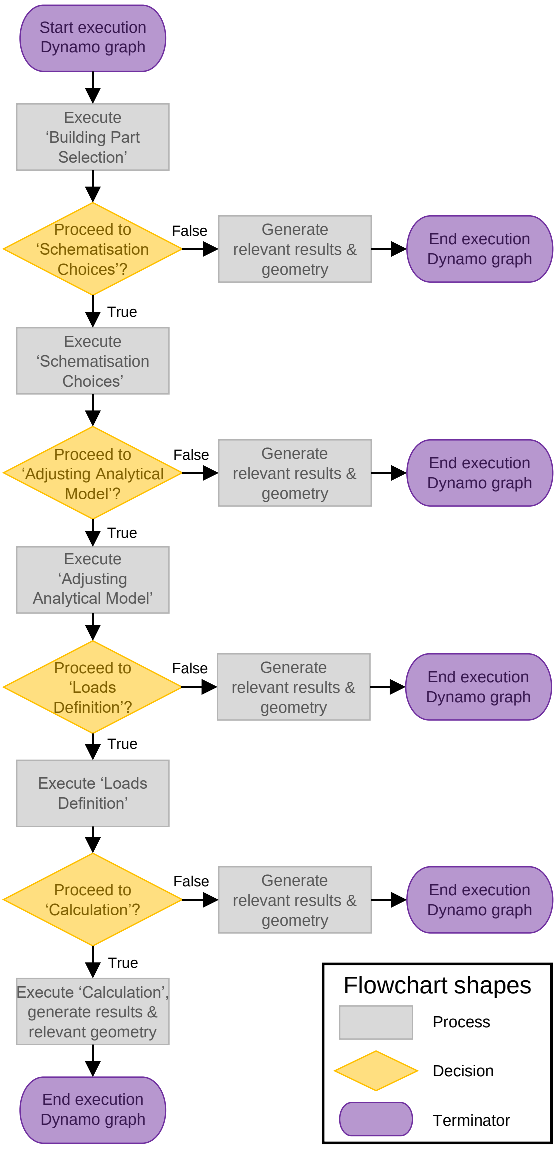

Start execution Dynamo graph
Execute 'Building Part Selection'
Proceed to 'Schematisation Choices'?
False
Generate relevant results & geometry
End execution Dynamo graph
True
Execute 'Schematisation Choices'
Proceed to 'Adjusting Analytical Model'?
False
Generate relevant results & geometry
End execution Dynamo graph
True
Execute 'Adjusting Analytical Model'
Proceed to 'Loads Definition'?
False
Generate relevant results & geometry
End execution Dynamo graph
True
Execute 'Loads Definition'
Proceed to 'Calculation'?
False
Generate relevant results & geometry
End execution Dynamo graph
True
Execute 'Calculation', generate results & relevant geometry
End execution Dynamo graph
Flowchart shapes
Process
Decision
Terminator

Figure 7.3: Logic behind the incremental execution of the Dynamo graph

Settings

The dark blue groups (3) contain the settings of the tool. This includes all aspects of the Dynamo tool that can be changed, but may not be changed by the end-user. For example the colours used in the visualisations of design choices or the file location of images which clarify the adjustment scenarios.

Building Part Selection

In the light green groups (4) the *'Building part selection'*-process is incorporated: all elements which intersect with the scope box that is selected by the user in the corresponding orange *'User interaction'*-group are filtered using these nodes.

Schematisation Choices

The light green groups (5) contain the nodes that process the design choices the user makes in the corresponding orange *'User interaction'*-group. In that group the user can select model elements from Revit and specify how the different elements in the considered building part should be used in the following steps. To this end four different element types were defined in Subsection 5.2.3. In the *'Schematisation Choices'*-groups, it is verified if the choices made by the user are possible, when this is not the case the user is warned. For example a wall cannot be classified as a *'Support'*-element, as that category is reserved for foundation piles only. Furthermore, warnings are raised if elements are not assigned to a category or if elements are assigned to multiple categories which is not possible.

Adjusting analytical model

In the purple groups (6) the adjustments that have to be made to the analytical model of the *'Schema'*-elements, as specified by the users in the corresponding orange *'User interaction'*-group, are processed.

To keep track of the adjustments made to the different *'Schema'*-elements, and to do the same for the elements classified into the other categories, Python classes were defined. In object oriented programming a class defines a type of something, describing parameters and functionalities belong to that specific type (Borrmann et al., 2018). Objects are instances of a class. For each of the four different categories a class is defined that describes what attributes belong to an object of that class, meaning a *'SchemaElement'*-object contains an attribute which describes the adjustments that are made to its analytical model, whereas a *'SupportElement'*-object contains an attribute which specifies the type of support. All four classes inherit the attributes of one parent class *'SC_Element'*, which specifies the attributes that belong to all four classes, like the Revit element and category they belong to. In Figure 7.4 a UML (Unified Modeling Language) diagram is included in which the relations between all Python classes that were defined are indicated.

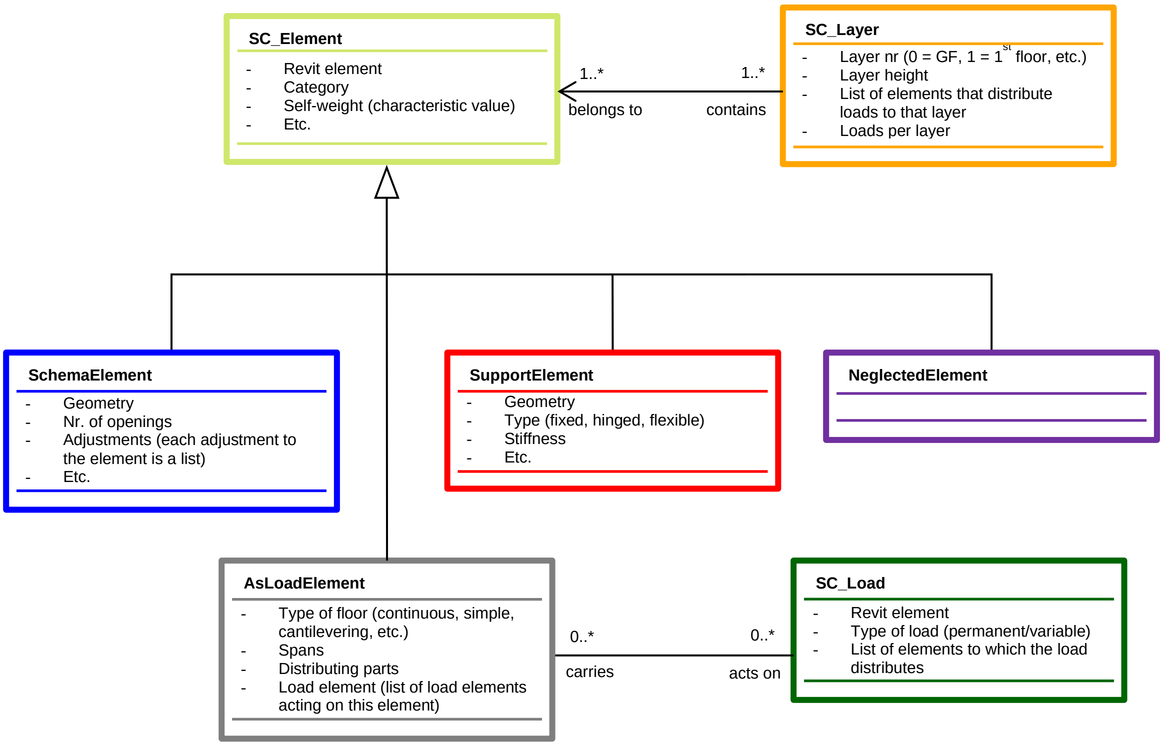

SC_Element
- Revit element
- Category
- Self-weight (characteristic value)
- Etc.
1..*
belongs to
1..*
contains
SC_Layer
- Layer nr (0 = GF, 1 = 1st floor, etc.)
- Layer height
- List of elements that distribute loads to that layer
- Loads per layer
SchemaElement
- Geometry
- Nr. of openings
- Adjustments (each adjustment to the element is a list)
- Etc.
SupportElement
- Geometry
- Type (fixed, hinged, flexible)
- Stiffness
- Etc.
NeglectedElement
AsLoadElement
- Type of floor (continuous, simple, cantilevering, etc.)
- Spans
- Distributing parts
- Load element (list of load elements acting on this element)
0..*
carries
0..*
acts on
SC_Load
- Revit element
- Type of load (permanent/variable)
- List of elements to which the load distributes

Figure 7.4: UML diagram indicating the relations between the different Python classes

In the nodes in the purple groups (6) the analytical model geometry of the *'Schema'*-elements is retrieved and adjusted according to the adjustments chosen by the user. The positions of the supports are derived in a similar manner, based on which a 2D schema is generated.

Loads definition

In the light blue groups (7) the loads acting on the schema are calculated for each of the building layers of the schema, based on the information in the BIM model (geometry of the elements, as well as the loads implemented by the BIM modeller). For each of the building layers an instance of the *'SC_layer'*-class is created, while for each of the loads implemented in the BIM model an instance of the *'SC_Load'*-class is created. For the proof of concept, only the loads acting on elements that are defined as *'AsLoadElement'* are considered, as can be seen from the relations between these classes in Figure 7.4. Based on the instructions of the user made in the in the corresponding orange *'User interaction'*-group, the loads per layer are calculated and drawn in the schema. In addition, the user can decide to reduce the 2D schema to a 1D schema.

Calculation

In the green groups (8) the schema with loads, as defined in the previous step, is exported to an XML-file that can be used to perform the calculation in SCIA-engineer (SCIA, 2020b). For the proof of concept, this step is implemented for the 1D schema only and data-exchange between Dynamo and SCIA Engineer is one-directional. In that way the functionality of the complete framework could be proofed. In future developments this step can be extended to allow for the implementation of 2D schema's and bi-directional data exchange.

7.1.2. Challenges faced during the implementation of the framework

Retrieving the analytical model through the Revit API

To properly implement the *'Adjusting analytical model'*-process, the analytical model corresponding to the geometry element has to be retrieved. In order to do this properly, the Revit API has to be accessed. There exists comprehensive documentation about the Revit API (Talarico, 2020), however some effort is still required to fully understand its workings and to properly implement it in the prototype tool.

Geometry tessellation

Drawing geometry in Dynamo requires a lot of memory, more than the actual generating of the geometry. If large numbers of elements have to be drawn in Dynamo, this can result in slowing the execution of the graph or eventually crashing of Dynamo. Drawn geometry is referred to as tessellated geometry. Throughout the graph, multiple operations involve the creation of geometry. To prevent the abovementioned issues, those operations are mainly performed within Python nodes to make sure the geometry created during those operations is not tessellated. Only, the geometry that is generated for visualisation purposes is actually drawn (Aggarwal, 2019).

Geometry working range

The geometry working range specifies the range of numbers Dynamo uses for the accurate rendering and calculation of geometry (Pratap, 2017). When the numbers that are used to perform geometry operations are outside of the geometry working range, there is no guarantee these operations are successful. In the development of the tool it was found that some of the elements of the test case building required a different geometry working range settings than others. The large concrete continuous floors required a larger geometry working range than the steel column and beam (IPE/HEA), meaning that independent of the geometry working range settings chosen geometry operations could not be performed successfully. To overcome this problem all geometry in the model was scaled down, so the same geometry working range could be used for the different elements in the model. A disadvantage of this solution is that it increases the risk of introduction of errors, as every time a value derived from a geometry operation is used in a calculation the value must be adjusted according to the chosen scale factor

Text or numbers in geometry preview
An important limitation of the use of Dynamo for the visualisation of design choices is that it is not possible to display text or numbers in the 3D watch nodes, as can be seen in Figure 7.2. This means dimensions, load names and numbers, or other clarifications cannot be displayed in Dynamo. To overcome this issue, a functionality has been included in the prototype tool through which the derived schematisations and loads can be shown in a Revit view.

7.2. Proof of concept

In this section the development of test cases that are used to test the framework are explained, as well as the outcomes of the user tests that were performed. Furthermore, a guideline is provided for further development of the prototype tool.

7.2.1. Test cases

Multiple test cases are used to develop and test the prototype tool. Starting point is a simple terraced house. The complexity of this first test case is increased in a step-by-step process, adding different element types from Table 6.2 in every following test case. This resulted in four different test cases which are included in Figure 7.5-7.8. The loads that were implemented in BIM to be able to perform the tests are included in Appendix B. These four test cases are used to develop the tool and proof that the framework can be successfully applied for all combinations of elements considered.

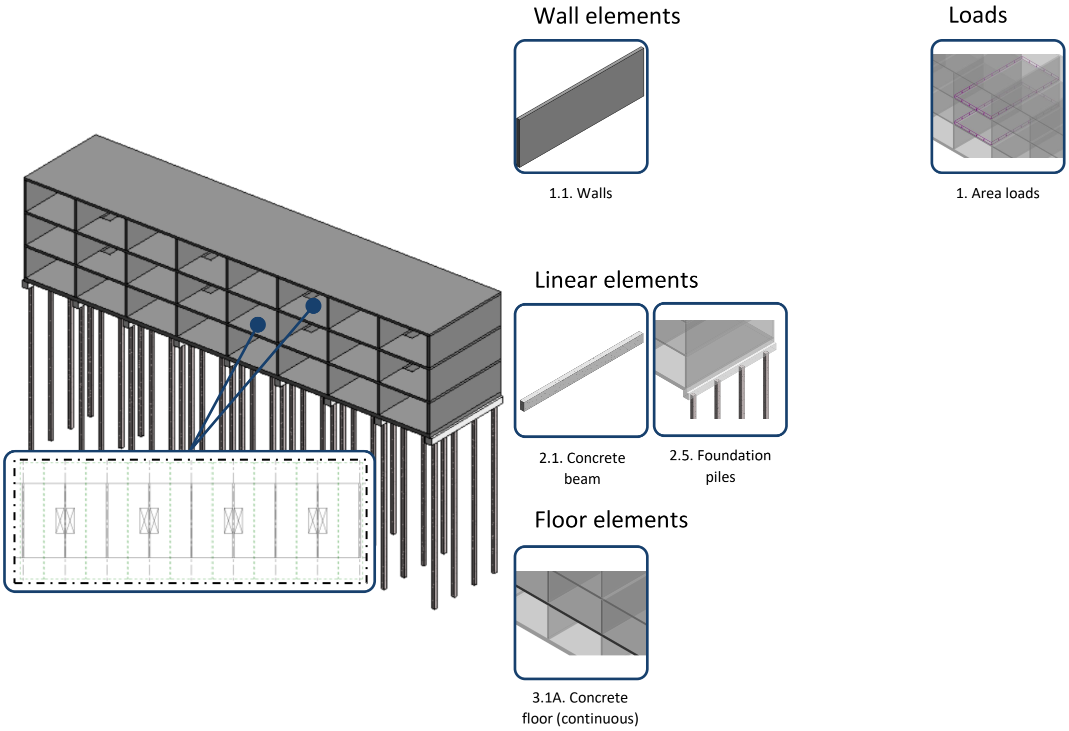


Figure 7.5: Test case A: Simple terraced house with floor openings at the first and second floor

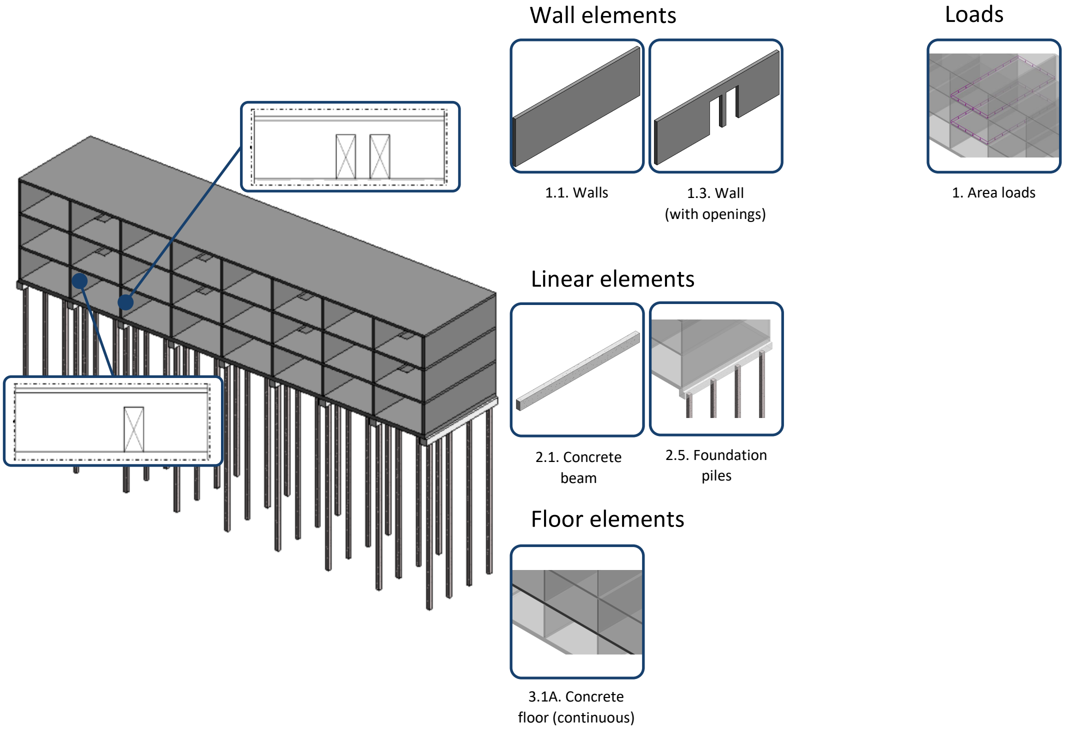


Figure 7.7: Test case B: Wall openings added to two of the walls at the ground floor

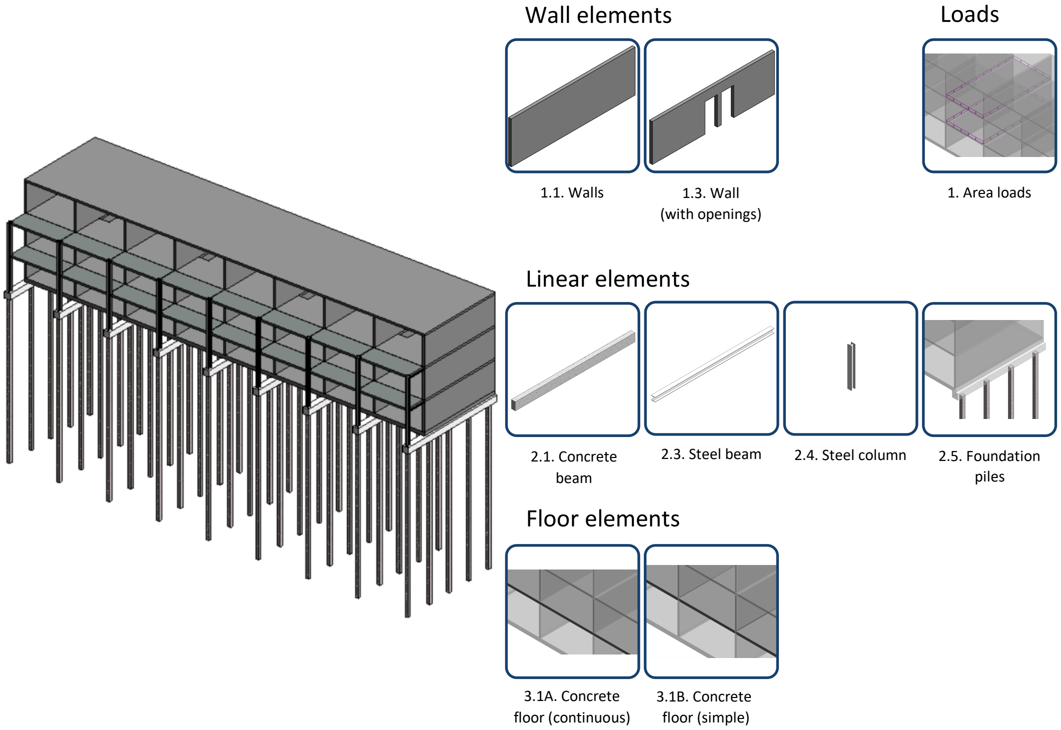


Figure 7.6: Test case C: Galleries supported by a steel structure added to the front of the building

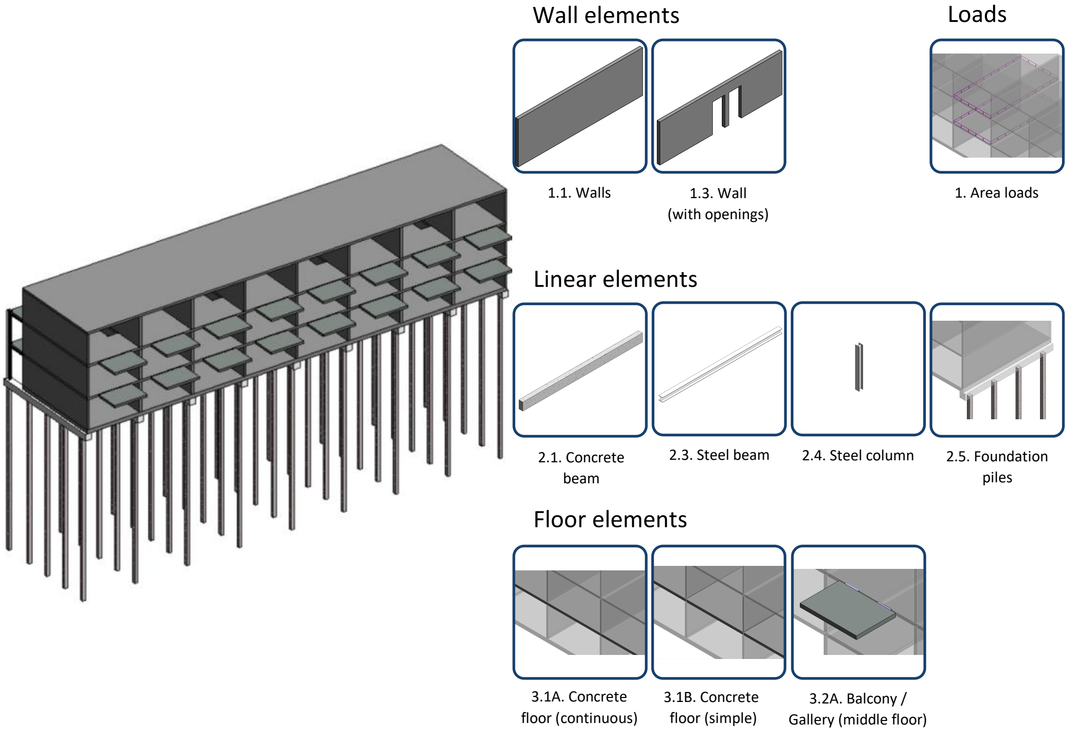


Figure 7.8: Test case D: Cantilevering balconies added to the back of the building

As a final test case, a BIM model of a larger and more complex building is used. This test case is included in Figure 7.9. The first four test cases were primarily used to demonstrate the usefulness of the framework. The aim of this final test case is to identify the limits of the prototype tool. Below the observations made during the testing of the test cases are summed up.

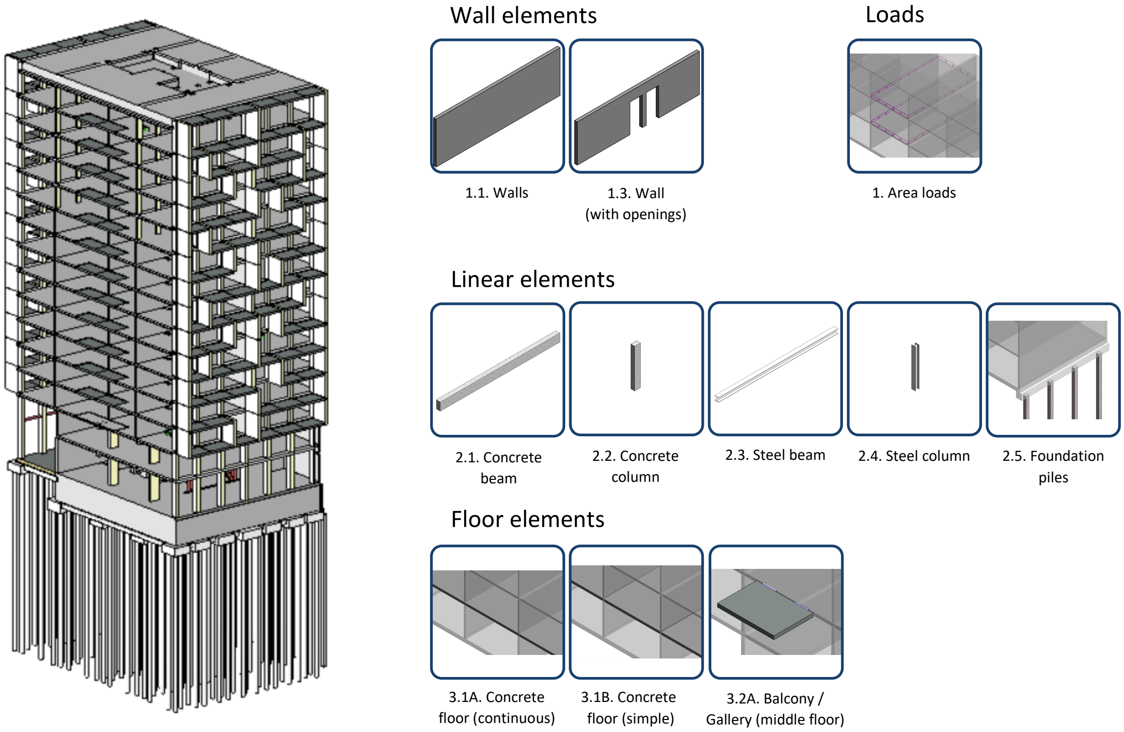


Figure 7.9: Test case E: Large, more complex BIM model

Observations

Test case A-D demonstrated that the following elements, as categorised in Table 6.2, were successfully implemented in the prototype tool:

- Wall elements 1.1 & 1.3
- Linear elements 2.1-2.5
- Floor elements 3.1A, 3.1B & 3.2A

The elements not yet included in the prototype tool, are similar to the ones presented in the list above and can be implemented in a similar manner, only requiring small alterations. Test case A-D showed that the prototype tool can be successfully used to perform a vertical load calculation for a simple building.

Test case E identified some limitations of the prototype software. The execution of the Dynamo graph is slower for larger building models. The *'Building Part Selection'*-process could be successfully performed. During the *'Schematisation Choices'*-process however, it was identified that the *Model groups* (groups of model elements per building layer) in the BIM model hindered the selection of individual elements on different layers. An attempt to select a single element from the model, resulted in the selection of the complete model group instead. For testing purposes, the model groups were ungrouped to be able to continue to the next step of the process. This is not a solution that can be applied in practice, as it is risky and inconvenient to ungroup and regroup all *Model groups* every time elements have to be selected.

In the *'Adjust Analytical Model'*-step the analytical model of all *'Schema'*-elements was retrieved successfully. Analytical model geometry was not completely as desired and needed to be adjusted. As not all adjustment scenarios are currently implemented in the prototype software, part of these alterations were performed manually.

During the execution of the *'Loads Definition'*-process Dynamo failed to perform certain operations, most likely due to the complex floor shape. To be able to continue testing, the outcomes that could not be determined automatically, were inserted manually. To prevent this in the future, more decision points should be implemented in the tool, to allow the user to specify the input in case Dynamo fails to perform a certain operation.

A 1D schema of the building could be successfully generated and exported to SCIA Engineer, and calculated afterwards.

7.2.2. User tests

To further test the practical applicability of the framework, the developed prototype tool was given to two structural engineers of CAE Nederland. It was investigated if they could perform a vertical load calculation for a simple building (test case C). At the start of the tests, the principle of the framework was shortly described and they were provided with a one-page manual of the tool (included in Appendix B). For the user tests the same loads as used in tests performed on the test cases were implemented in the BIM model (see Appendix B). The most important observations of the user tests are discussed below.

Observations

The user tests demonstrated that the developed framework was successfully incorporated in the tool. Both participants recognised the approach behind the framework in the way the tool was set up: Automated parts followed by decision points through which the user can incorporate design choices. Both participants valued the way users are informed about the choices they can make and how the effects of their choices are visualised in the 3D watch nodes. Furthermore, the user tests showed that the way nodes are organised in each of the orange *'User Interaction'*-groups can be improved. It was not always clear to the participants what nodes belonged together. Next to this, they would also prefer to be able to export characteristic load values to SCIA Engineer, as currently only design values are exported. Lastly, the user tests showed that some basic knowledge of both Revit and Dynamo is required to be able to use the tool.

7.2.3. Road map

This subsection describes the steps needed to take, to further develop the tool. For each of the steps of the framework included in the prototype tool a timeline of its complete development process is given. This time line is represented by a bar along which the features belonging to that process step are indicated. The extent to which this bar is filled up shows which features are currently implemented in the prototype tool. The timelines are included in Figure 7.10-7.14.

Some additional remarks about the different functionalities of the prototype software are needed. Currently, the element selection is fully implemented, as can be seen from Figure 7.11. However, to overcome the problems with *Model groups* as identified using test case E, alternative ways of selecting need to be investigated. Furthermore, more decision points should be added to the tool, to allow the user to specify the values that should be used in the following steps, if Dynamo fails to perform a certain operation.

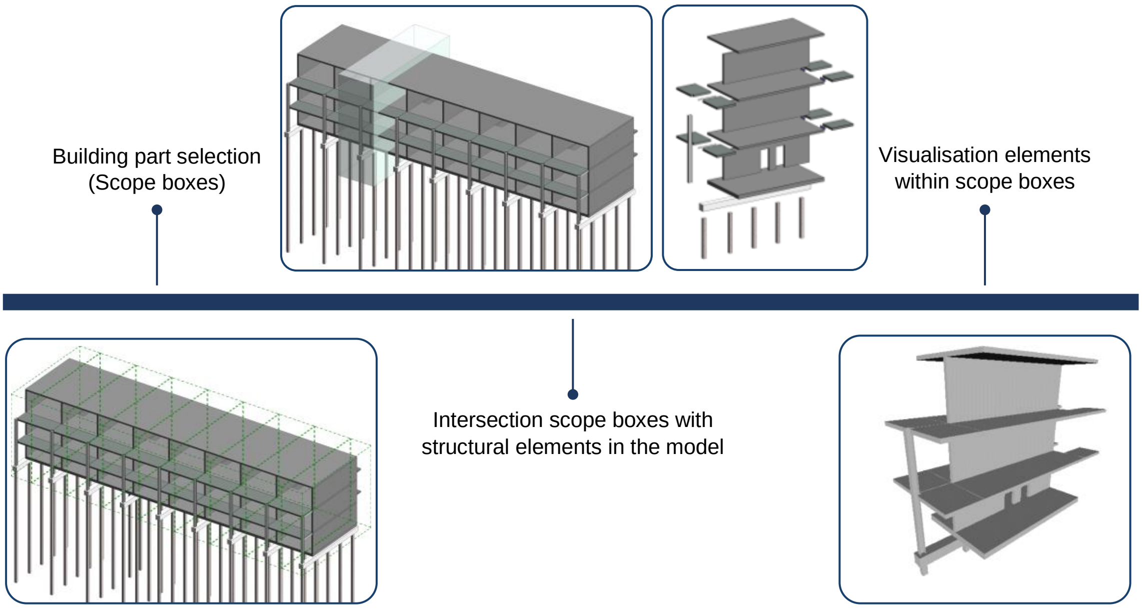


Figure 7.10: Timeline indicating the development of the ‘Building Part Selection’-process

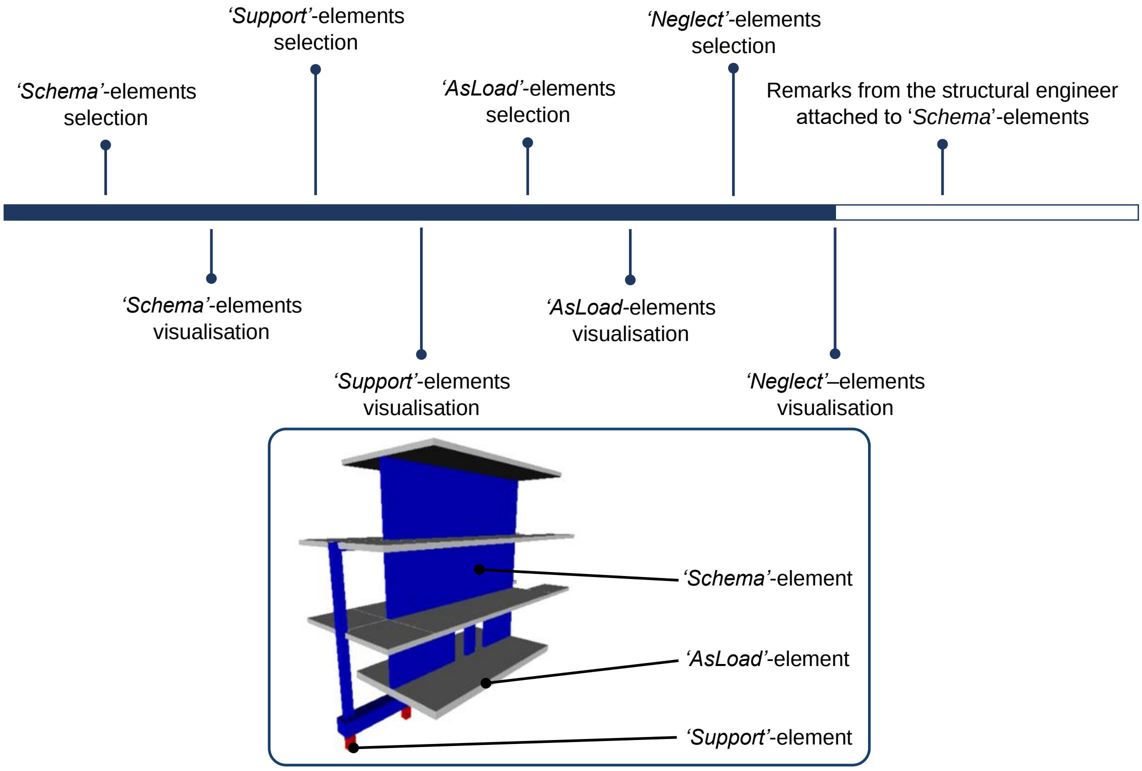

'Schema'-elements selection
'Support'-elements selection
'AsLoad'-elements selection
'Neglect'-elements selection
Remarks from the structural engineer attached to '*Schema*'-elements
'Schema'-elements visualisation
'Support'-elements visualisation
'AsLoad-elements visualisation
'Neglect'–elements visualisation
'Schema'-element
'AsLoad'-element
'Support'-element

Figure 7.11: Timeline indicating the development of the 'Schematisation Choices'-process

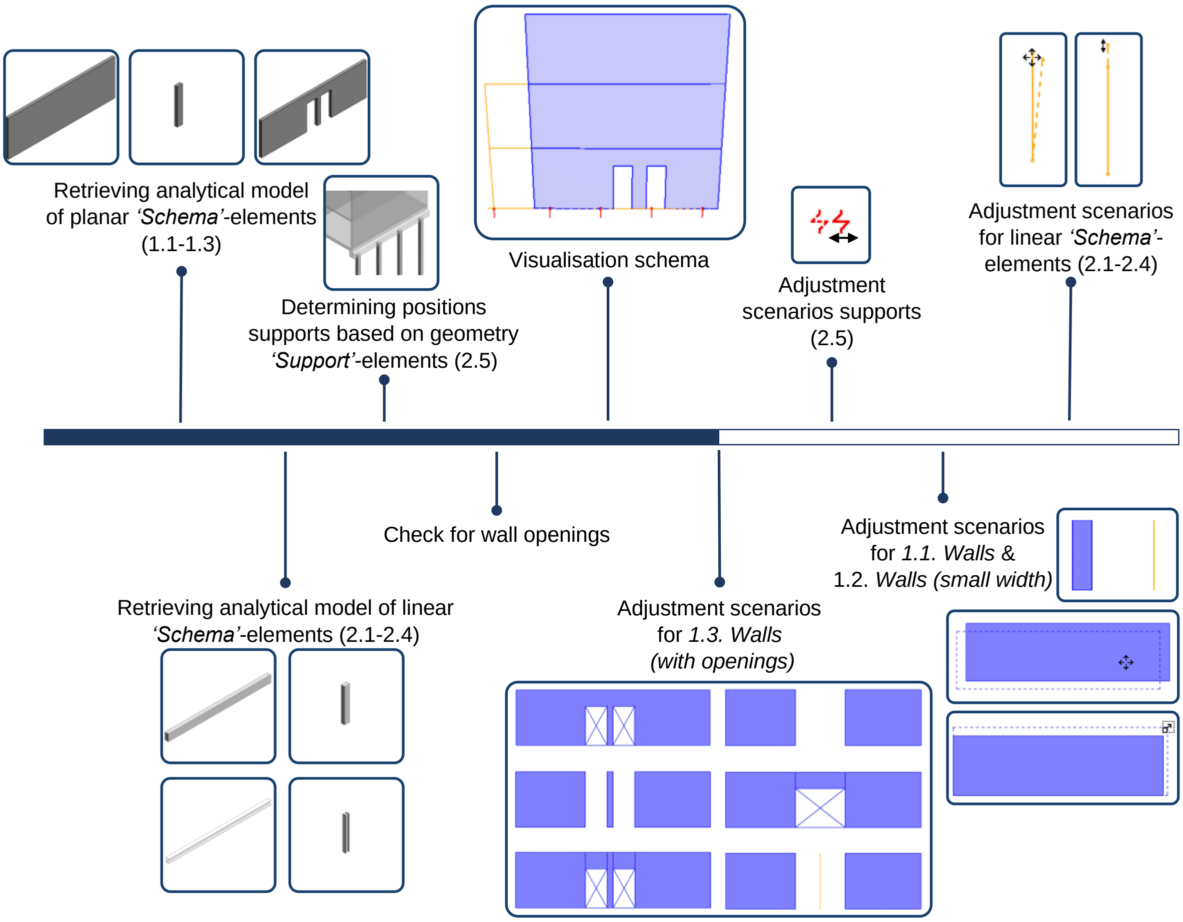

Retrieving analytical model of planar *'Schema'*-elements (1.1-1.3)
Determining positions supports based on geometry *'Support'*-elements (2.5)
Visualisation schema
Adjustment scenarios supports (2.5)
Adjustment scenarios for linear *'Schema'*-elements (2.1-2.4)
Retrieving analytical model of linear *'Schema'*-elements (2.1-2.4)
Check for wall openings
Adjustment scenarios for *1.3. Walls (with openings)*
Adjustment scenarios for *1.1. Walls* & 1.2. *Walls (small width)*

Figure 7.12: Timeline indicating the development of the 'Adjusting Analytical Model'-process

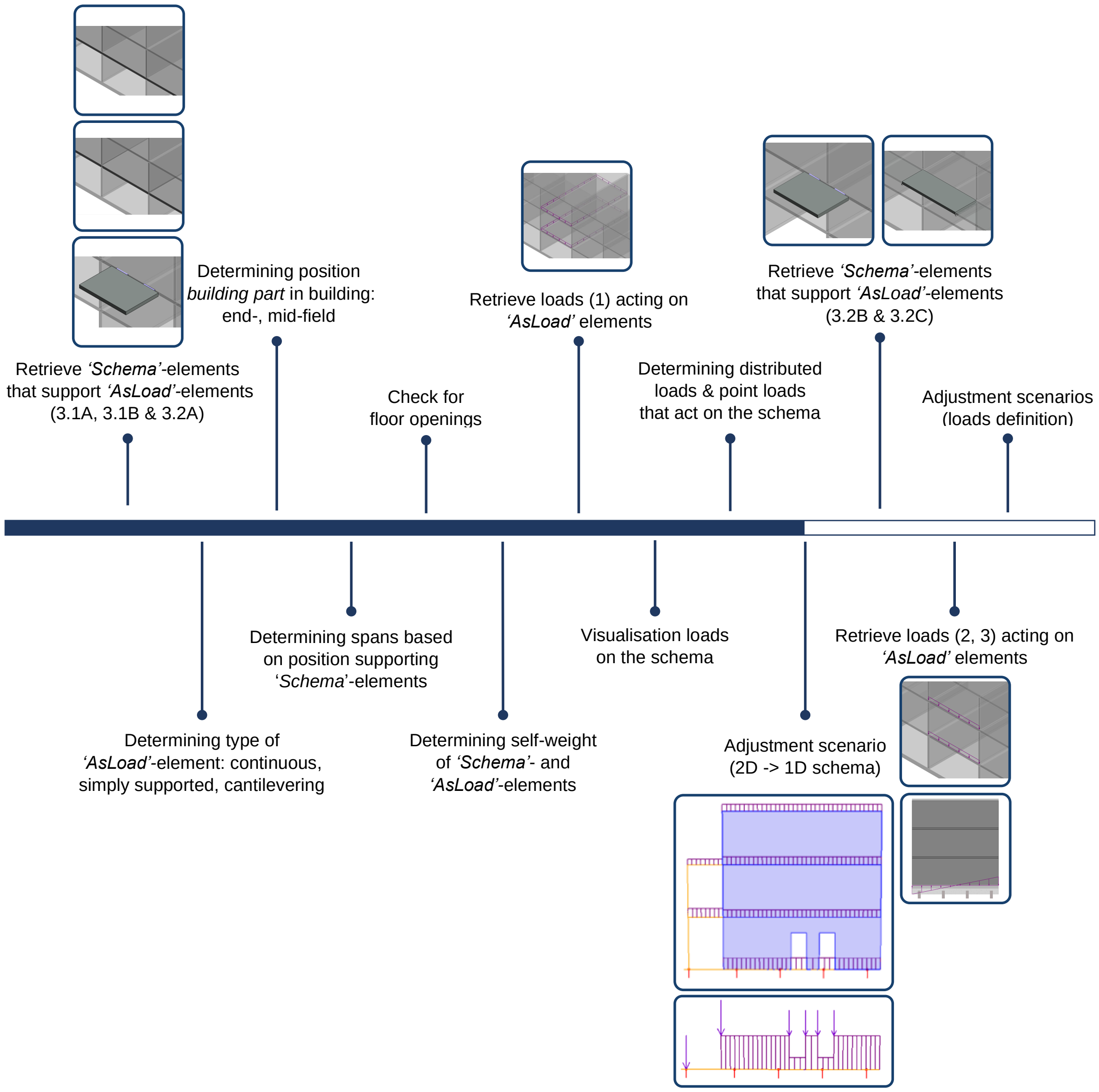

Retrieve *'Schema'*-elements that support *'AsLoad'*-elements (3.1A, 3.1B & 3.2A)
Determining type of *'AsLoad'*-element: continuous, simply supported, cantilevering
Determining position *building part* in building: end-, mid-field
Determining spans based on position supporting *'Schema'*-elements
Check for floor openings
Determining self-weight of *'Schema'*- and *'AsLoad'*-elements
Retrieve loads (1) acting on *'AsLoad'* elements
Visualisation loads on the schema
Determining distributed loads & point loads that act on the schema
Adjustment scenario (2D -> 1D schema)
Retrieve *'Schema'*-elements that support *'AsLoad'*-elements (3.2B & 3.2C)
Retrieve loads (2, 3) acting on *'AsLoad'* elements
Adjustment scenarios (loads definition)

Figure 7.13: Timeline indicating the development of the 'Loads Definition'-process

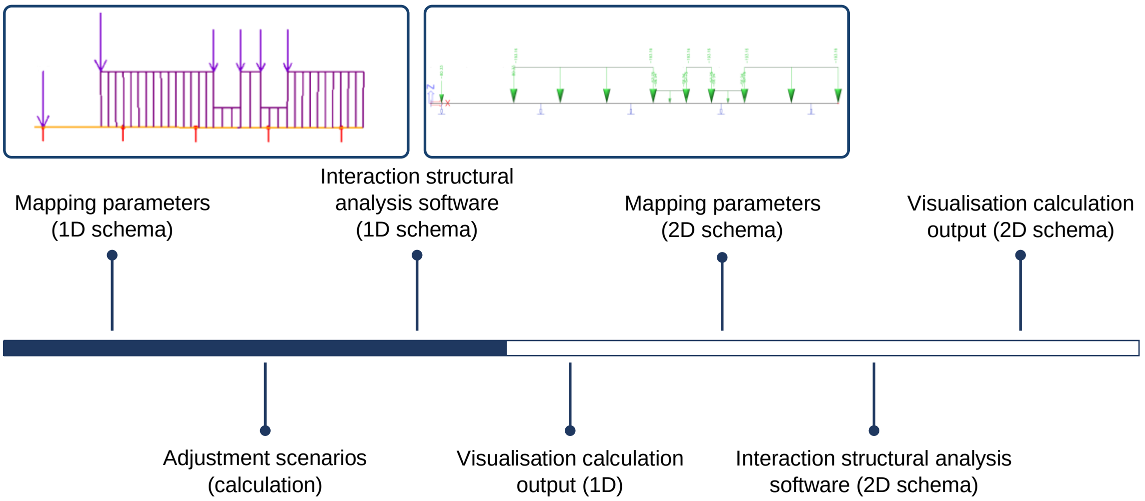

Mapping parameters (1D schema)
Adjustment scenarios (calculation)
Interaction structural analysis software (1D schema)
Visualisation calculation output (1D)
Mapping parameters (2D schema)
Interaction structural analysis software (2D schema)
Visualisation calculation output (2D schema)

Figure 7.14: Timeline indicating the development of the 'Calculation'-process

7.3. Discussion

In this chapter a prototype tool of the framework developed in Chapter 6 is build using Dynamo. Testing using several test cases demonstrated that the individual elements defined in the framework could be successfully implemented in a tool. It was also shown that the developed prototype tool could be successfully used to perform a vertical load calculation for a simple building. A more complex test case identified limitations that need attention in further development of the tool. Furthermore, user tests were performed to verify the practical applicability of both framework and tool. Both participants of these tests considered the tool useful as they valued the way the user can control the process through decision points and how design choices were visualised in 3D previews.

Lastly, a road map was developed, describing the current state of development of the prototype tool and providing a guideline for further development of the tool.

In the following chapters the tool and framework are discussed from the perspective of achieving the research objective and regarding their limitations. Furthermore, the answers to the research questions are given and recommendations for further research and development are summed up.

Part 5: Discussion, conclusion & recommendations

8

Discussion

The aim of this MSc thesis was to research and develop a method and tool that can be used to automate the vertical load calculation. To achieve this, a framework and prototype tool are developed. In this chapter, the implications that arise from the use of the framework and tool, as well as their limitations, are discussed.

Organisation of the vertical load calculation
There are two views on the influence the tool and framework have on the way of working of the structural engineers. On the one hand, the framework is based on the current organisation of the vertical load calculation, which helps the structural engineer recognise the different steps in the process. On the other hand, the structural engineer will have to use the BIM model more, and in a different way than currently the case. There is a shift in the division of responsibilities between the BIM modeller and the structural engineer. The BIM modeller becomes responsible for assigning loads to elements and the structural engineer has to incorporate design choices in the model. As the digital transformation of the structural engineering progresses, this trend is expected to continue. Due to the difference in educational background, the majority of this change will have to be accounted for by the structural engineer. Therefore, as Rolvink et al. (2014) already pointed out, structural engineers should acquaint themselves with these new technologies, in order to be able to use them. In this, the education of the next generation structural engineers is important. By providing courses on BIM, parametric engineering and programming, the TU Delft is moving in the right direction. For the future, it remains of importance to follow along the new developments in the field. But what about current practice? How should they adjust to these developments? Further research should identify how the changing interface between BIM modeller and structural engineer can be best organised. Is a specialised function needed that manages this interface and is the growing knowledge about digitalisation and automation of the new generation of structural engineers sufficient to cover such a function?

BIM-based automation tools
The outcomes of this MSc thesis demonstrate the potential of BIM-based automation of structural calculations. The developed prototype tool provides a substantial first step towards the BIM-based automation of the vertical load calculation. Using the tool, the structural engineer can quickly generate a schematisation with a high level of detail, as well as directly perform a calculation using the direct link to structural analysis software. Moreover, the structural engineer is supported in documenting design choices in an earlier stage of the design process and in a visual manner. This helps to prevent errors and saves time that can be used for more complex and detailed calculations.

As holds for any other software, the outcomes of automation tools should not be blindly trusted. Automation tools should therefore be set up in such a way that the structural engineer remains in control of the process. Besides, it is impossible to automate every part of a calculation process, as certain tasks cannot be represented in the form of logical rules but require expert engineering knowledge. To account for the above, the approach as followed in this MSc thesis is recommended. To make clear what is happening automation tools should display in-between results, supported by visualisations. Decision points should be specified, at which the structural engineer can adjust and determine the outcomes of the calculation process.

Lack of available previous research
The amount of useful literature available on the topic of BIM-based automation of structural calculations is limited. This might be due to the fact that if automation tools are developed within companies, information regarding those tools can be confidential and is therefore not publicly published. The literature that is available, might become outdated, as the research on the topic is quickly developing. Therefore, practical research is recommended as a means for establishing a theoretical background for further research.

Software and interoperability
In the building industry a wide range of software tools are in use. Each of those tools is specialised in a specific aspect of the design process of a building. It seems ideal to combine these features in one super-intelligent software, as this would reduce the loss of information due to data-exchange and minimise the need to manually re-enter information. On the contrary, if such a super-intelligent tool would exist, the user would be provided with a lot of information which is irrelevant for their specific use case. A separate software package is more specific and therefore more insightful. Besides, it is currently not possible to create such a super-intelligent software. To limit the introduction of errors and to improve working processes the interfaces between different software packages should be improved. For example in the way proposed by this MSc thesis, by mapping the different parameters between the software packages to prevent the loss of data due to misinterpretations. If more than two different software packages have to be connected, the BIM model should be used as a central model to which the others are connected. In the BIM model, all information can be related to the 3D geometry, which is relevant for all users and therefore a valuable means for collaboration.

Narrowed scope of the developed framework
In Chapter 6, the scope of the framework was narrowed to be able to give a more precise description of the different steps in the process and to allow for a clear assignment of responsibilities. The focus was on the approach to the vertical load calculation in which the building is divided into vertical parts according to the main load-bearing walls. Furthermore, the framework was specifically developed for the use of Revit as BIM software. This limits the use cases of the developed framework. The concept framework (Chapter 5) was set up without focussing on a specific approach or BIM software. It can therefore be used as a basis for the development of a more detailed framework in case a specific use case does not fit the narrowed scope of the developed framework.

Tool developed as proof of concept
The prototype tool was specifically developed as a proof of concept. Hence, only the elements that are essentially different from each other were implemented in the tool. The missing elements can be implemented in a similar manner as those already implemented. The same holds for the adjustment scenarios. The connection to structural analysis software also provides possibilities for further development, for the proof of concept a one-directional link for the 1D schema was implemented. In that way the functionality of the complete process described in the framework could be proofed. This can be extended to 2D schema’s and a bi-directional connection, which is especially beneficial for collaboration within a design team. Another feature that was irrelevant for the proof of concept, but is important for the eventual application of the framework in practice is repetition. In further development of the tool there should be accounted for both repetition within the project, as well as repetition within the building. Moreover, because it is an important characteristic of a vertical load calculation.

Dynamo in combination with Revit, provided an easy-to-use platform for the proof of concept. However, testing identified some limitations that have to be overcome in future development. Large and complex building models resulted in slow execution speeds or even failing of the graph. Dynamo's geometry preview is limited, as dimensions and explanatory texts cannot be displayed. In the tool, a work-around was used through which the schematisations could be shown in Revit. This work-around is inconvenient for practice, as it would require an undesirable additional action from the side of the structural engineer. For further development it is therefore recommended to consider using a more advanced programming language, like C# or Python, in combination with a graphical user interface that has more suitable features than Dynamo's.

Limited testing
The framework and tool were successfully tested with a limited number of test cases and using a limited number of user tests. The user tests focused on the structural engineer, meaning the steps in which the BIM modeller is involved were excluded. This was sufficient for the goal of these tests: proving the usefulness of the developed method. For further development however, more extensive testing is recommended. The complete process should be tested, including the process steps in which the BIM modeller is involved.

9 Conclusions

In this chapter the conclusions drawn from this master thesis are summed up. First the answers to the sub-research questions will be given, after which the main research question will be answered.

1. *How is the vertical load calculation currently organised and which software is used to support the calculation?*

- In a vertical load calculation the foundation of a building is dimensioned. To do this, the structural engineer uses a schematisation of the load-bearing structure to define how the loads acting on the building are transferred to the foundation.
- The vertical load calculation is supported by BIM models, which contains 3D geometry and metadata that become more accurate and elaborate as the design proceeds. Excel spreadsheets are used to support the calculation. In these, a part of the calculation is already automated, as the user only has to specify the project-specific information.
- From interviews with several structural engineers, it was concluded that the organisation of the vertical load calculation between different companies is similar, the main difference is the way in which buildings are divided into parts that are considered separately in the calculation: horizontally (per floor) or vertically (per load-bearing wall).

2. *What methods and software tools already exist that strive to automate similar processes?*

- From the literature review it was concluded that the amount of literature on BIM-based automation of structural calculations is limited. The research is mainly focussed on tools for the conceptual design and construction phase, while the vertical load calculation is mainly performed during the developed design phase.
- The interviews indicate that the level of automation and digitalisation is increasing in the Dutch structural engineering practice. Although the participants named various examples of tools that automate structural calculations, none of the participants knew of a BIM-based automation tool as proposed in this MSc Thesis.

3. *What currently available software can be used to automate the vertical load calculation?*

- In the literature review reference tools were identified. Most are based on object-oriented programming languages like Python and C#, that interact with both BIM and structural analysis software through their API. In some cases, parametric engineering software, like Dynamo or Grasshopper was used.
- These software were applied to a number of approaches, from which the following three are classified as suitable for the automation of the vertical load calculation: knowledge based engineering, file-based exchange and parametric engineering.

4. *What are requirements for a method and/or software tool that strives to automate the vertical load calculation?*

- From the interviews and literature review it was concluded that the following groups of requirements should be fulfilled:
    - Requirements related to the definition of the method: A well-defined scope is required to indicate for what use cases the method and tool can be applied. Each step in the process should be clearly defined: specifying the input, output, operations performed, as well as responsibilities.
    - Requirements related to user interaction: The structural engineer must remain in control of the process, and should therefore be able to incorporate and document design choices.
    - Requirements related to usability: The tool should be easy-to-use, it should be able to handle all input that fits within the scope and provide the user with the expected output.

5. *How should a method that strives to automate the vertical load calculation be composed?*

- A framework was developed, describing the different process steps of the vertical load calculation and assigning responsibilities to the persons involved. The calculation was divided into several automated parts, in-between which decision points are implemented to allow the structural engineer to adjust the automatically generated output of the previous part and incorporate design choices that are considered in the following parts.
- An important part is the *'Adjusting Analytical Model'*-process step. This step is added to overcome the representation difference between BIM and schematisation and to allow for the correction of the analytical model that is automatically created by the BIM software.

6. *Can such a method be successfully applied to practical cases?*

- A tool, developed as a proof of concept in Dynamo, showed that the framework can be successfully applied to a simple building. Limitations of the current tool were identified using a test case, with a larger, more complex building. The execution speed was reduced and some operations failed to succeed.
- User tests were successful, as the participants recognised the approach of the framework in the tool and were able to use them. The visualisation of design choices and the adjustment scenarios were seen as the main advantages. The main limitations identified are the required knowledge of Revit and Dynamo and the limited functionalities of the *'Calculation'*-step.

To answer the main research question:

> *How can manual calculation processes in structural calculations, specifically in the vertical load calculation of multi-storey residential buildings, be automated using a 3D BIM model?*

The vertical load calculation can be automated based on a BIM model. To do this, the input, the output and the operations involved in of each of the process steps should be clearly defined. Also, responsibilities should be allocated to the persons involved in the process. An automation tool should be set up in such a way that the structural engineer remains in control of the process. By providing decision points through which the structural engineer can incorporate design choices this can be achieved.

The developed BIM-based automation tool and framework, demonstrate the potential of BIM-based automation of structural calculations. Detailed schematisations can be derived quickly and calculations can be performed directly due to the connection to structural analysis software. Moreover, the structural engineer is supported in documenting design choices early. This prevents the introduction of errors and gives the structural engineer more time to focus on the more complex and detailed parts of structural calculations.

10
Recommendations

This chapter describes the possibilities for further study and development, for both the developed framework and the prototype tool. Recommendations for further research:

- This MSc thesis demonstrated that the digital transformation of structural engineering practice shifts responsibilities between BIM modellers and structural engineers. Structural engineers need to acquire knowledge about new technologies that come with these developments. Future research should identify how this can best be organised in practice. Is a specialised function needed within engineering firms and is the growing knowledge of automation and digitalisation of new structural engineers sufficient to cover such a function?

- The scope of the developed framework and tool is limited to the use of Revit and the approach to the vertical load calculation in which the building is divided into vertical parts. Further research should explore the possibilities of extending the concept framework (Chapter 5) for different BIM software and other approaches.

- The prototype tool was limited to one-directional exchange of data between BIM and structural analysis software, further research should identify how a bi-directional connection could be established, as this would provide advantages for collaboration within project teams.

Recommendations for practice:

- It is recommended to apply BIM-based automation of structural calculations in practice, as a lot of information already present in the BIM model can be used for calculation purposes. The partial automation of calculation processes and the need to document design choices early, prevent the introduction of errors and allow structural engineers to focus more on the detailed and complex parts of structural calculations.

- In order to keep automated structural calculations insightful and to assure the structural engineer remains in control of the process it is recommended to split the calculation into automatable parts. In-between these, the structural engineer can adjust and steer the outcomes at smartly chosen decision points.

- Further development of the prototype tool should extend and improve current functionalities. To overcome the limitations as described Chapter 8, it should be considered to use a more advanced programming language, like Python or C#, in combination with a graphical user interface with more functionalities than Dynamo's.

- Before the tool is applied in practice, it is recommended to perform more extensive testing of the complete process, also including the BIM modeller. Using these tests it can be identified if the functionalities described in the framework need to be improved or extended.

References

Agarwal, R., Chandrasekaran, S., & Sridhar, M. (2016). Imagining construction's digital future. *McKinsey & Company*.

Andia, A. & Spiegelhalter, T. (2014). *Post-parametric automation in design and construction*. Artech House.

Aram, S. (2015). *A knowledge-based system framework for semantic enrichment and automated detailed design in the AEC projects*. PhD thesis, Georgia Institute of Technology.

Arayici, Y., Fernando, T., Munoz, V., & Bassanino, M. (2018). Interoperability specification development for integrated BIM use in performance based design. *Automation in Construction*, 85, 167–181.

BIM Loket (2019a). *NL-SfB.* Retrieved from: https://www.autodesk.nl/products/revit/overview. Last accessed 30/10/2019.

BIM Loket (2019b). *Wat is BIM eigenlijk?* Retrieved from: https://www.bimloket.nl/watisbim. Last accessed 08/01/2020.

Bogani, E., Boeykens, S., et al. (2008). Metadata for 3D Models. How to search in 3D Model repositories? In *ICERI 2008-International Conference of Education, Research and Innovation* (pp. 1–11).: ESP.

Borrmann, A., König, M., Koch, C., & Beetz, J. (2018). *Building Information Modeling: Technology Foundations and Industry Practice*. Springer.

Bovenberg, A. (2015). StructuralComponents 4: Conceptual building models with structural design justification. Master's thesis, Delft University of Technology.

Braun, V. & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2), 77–101.

Breakwell, G. M., Hammond, S. E., & Fife-Schaw, C. E. (2000). *Research methods in psychology*. Sage Publications Ltd.

buildingSMART (2020a). *Industry Foundation Classes (IFC).* Retrieved from: https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/. Last accessed 10/01/2020.

buildingSMART (2020b). *Information Delivery Manual (IDM).* Retrieved from: https://technical.buildingsmart.org/resources/information-delivery-manual/. Last accessed 10/01/2020.

buildingSMART (2020c). *Model View Definition (MVD) - An Introduction.* Retrieved from: https://technical.buildingsmart.org/standards/mvd/. Last accessed 10/01/2020.

Cambridge Dictionary (2020). *Meaning of metadata in English.* Retrieved from: https://dictionary.cambridge.org/dictionary/english/me. Last accessed 08/01/2020.

Cavieres, A., Gentry, R., & Al-Haddad, T. (2011). Knowledge-based parametric tools for concrete masonry walls: Conceptual design and preliminary structural analysis. *Automation in Construction*, 20(6), 716–728.

Chi, H.-L., Wang, X., & Jiao, Y. (2015). BIM-enabled structural design: impacts and future developments in structural modelling, analysis and optimisation processes. *Archives of computational methods in engineering*, 22(1), 135–151.

Czmoch, I. & Pekala, A. (2014). Traditional design versus BIM based design. *Procedia Engineering*, 91, 210–215.

de Maat, E., Krabben, K., Winkels, R., et al. (2010). Machine learning versus knowledge based classification of legal texts. In *JURIX* (pp. 87–96).

Designing Buildings Ltd. (2019). *Technical design.* Retrieved from: https://www.designingbuildings.co.uk/wiki/4:_Technical_design. Last accessed 22/01/2020.

Dixon, J. R. (1995). Knowledge-based systems for design. *Journal of Vibration and Acoustics*, 117(B), 11–16.

Donaldson, T. (2003). Python as a first programming language for everyone. In *Western Canadian Conference on Computing Education*, volume 547 (pp. 2015).

Dym, C. L., Little, P., & Orwin, E. J. (2013). *Engineering Design: A Project-Based Introduction*. John Wiley & Sons, 4rd edition edition.

Eastman, C. M., Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2011). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons.

Eleftheriadis, S., Duffour, P., Stephenson, B., & Mumovic, D. (2018). Automated specification of steel reinforcement to support the optimisation of RC floors. *Automation in Construction*, 96, 366–377.

Eleftheriadis, S., Mumovic, D., & Greening, P. (2017). Life cycle energy efficiency in building structures: A review of current developments and future outlooks based on BIM capabilities. *Renewable and Sustainable Energy Reviews*, 67, 811–825.

Fangohr, H. (2004). A comparison of C, MATLAB, and Python as teaching languages in engineering. In *International Conference on Computational Science* (pp. 1210–1217).: Springer.

Farzaneh, A., Carriere, J., Forgues, D., & Monfet, D. (2018). Framework for using building information modeling to create a building energy model. *Journal of Architectural Engineering*, 24(2), 05018001.

Hasan, A. M., Torky, A. A., & Rashed, Y. F. (2019). Geometrically accurate structural analysis models in BIM-centered software. *Automation in Construction*, 104, 299–321.

Hohrath, B. L. (2018). StructuralComponents 5: Super element based tool for early design collaboration applied to mid-rise buildings. Master’s thesis, Delft University of Technology.

Hu, Z.-Z., Zhang, X.-Y., Wang, H.-W., & Kassem, M. (2016). Improving interoperability between architectural and structural design models: An industry foundation classes-based approach with web-based tools. *Automation in Construction*, 66, 29–42.

Hyun, C., Jin, C., Shen, Z., & Kim, H. (2018). Automated optimization of formwork design through spatial analysis in building information modeling. *Automation in Construction*, 95, 193–205.

Jabi, W. (2013). *Parametric design for architecture*. Laurence King Publ.

Kissling, E. A. (1996). Bleeding out loud: Communication about menstruation. *Feminism & Psychology*, 6(4), 481–504.

Lai, H. & Deng, X. (2018). Interoperability analysis of IFC-based data exchange between heterogeneous BIM software. *Journal of civil engineering and management*, 24(7), 537–555.

Lee, G. (2011). What information can or cannot be exchanged? *Journal of Computing in Civil Engineering*, 25(1), 1–9.

Liu, H., Singh, G., Lu, M., Bouferguene, A., & Al-Hussein, M. (2018). BIM-based automated design and planning for boarding of light-frame residential buildings. *Automation in Construction*, 89, 235–249.

Liu, Z., Osmani, M., Demian, P., & Baldwin, A. (2015). A bim-aided construction waste minimisation framework. *Automation in construction*, 59, 1–23.

Ma, H., Ha, K. M. E., Chung, C. K. J., & Amor, R. (2006). Testing semantic interoperability. In *Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, Canada*.

Mangal, M. & Cheng, J. C. (2018). Automated optimization of steel reinforcement in RC building frames using building information modeling and hybrid genetic algorithm. *Automation in Construction*, 90, 39–57.

Mora, R., Bédard, C., & Rivard, H. (2008). A geometric modelling framework for conceptual structural design from early digital architectural models. *Advanced Engineering Informatics*, 22(2), 254–270.

Nederlands Normalisatie-Instituut (2011). *Eurocode 2: Design of concrete Structures - Part 1-1: General rules and rules for buildings* (NEN-EN 1992-1-1). Retrieved from NEN Connect.

Nguyen, T.-H. & Vu, A.-T. (2020). Building information modeling based optimization of steel single-plate shear connections using differential evolution algorithm. In *CIGOS 2019, Innovation for Sustainable Infrastructure* (pp. 1199–1204). Springer.

NIBS (2019). *National BIM standard United States V3.* Retrieved from: https://www.nationalbimstandard.org/faqs#faq1. Last accessed 08/01/2020.

Preidel, C. & Borrmann, A. (2015). Automated code compliance checking based on a visual language and building information modeling. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 32 (pp.1).: IAARC Publications.

Qin, W. (2018). The Application of Building BIM Information Modeling in Large Domestic Projects. In *2018 3rd International Workshop on Materials Engineering and Computer Sciences (IWMECS 2018)*: Atlantis Press.

Ren, R., Zhang, J., & Dib, H. N. (2018). BIM interoperability for structural analysis. In *Construction Research Congress 2018: Construction Information Technology*.

RIBA (2012). *BIM Overlay to the RIBA Plan of Work*. RIBA Publishing.

RIBA (2013). *RIBA Plan of Work 2013.* Retrieved from: https://www.ribaplanofwork.com/Default.aspx. Last accessed 20/11/2019.

RIBA (2019). *National BIM Report 2019*. Technical report, NBS.

Rolvink, A., Mueller, C., & Coenders, J. (2014). State on the art of computational tools for conceptual structural design. In *Proceedings of IASS Annual Symposia*, volume 2014 (pp. 1–8).: International Association for Shell and Spatial Structures (IASS).

Romero, D. A. N. (2019). StructuralComponents 5: Super element based tool for early design collaboration applied to mid-rise buildings. Master’s thesis, Delft University of Technology.

Sacks, R. & Barak, R. (2008). Impact of three-dimensional parametric modeling of buildings on productivity in structural engineering practice. *Automation in Construction*, 17(4), 439–449.

Sandberg, M., Gerth, R., Lu, W., Jansson, G., Mukkavaara, J., & Olofsson, T. (2016). Design automation in construction: An overview. In *33rd CIB W78 Conference 2016, Oct. 31st–Nov. 2nd 2016, Brisbane, Australia*.

Steel, J., Drogemuller, R., & Toth, B. (2012). Model interoperability in building information modelling. *Software & Systems Modeling*, 11(1), 99–109.

Storn, R. & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341–359.

Tafraout, S., Bourahla, N., Bourahla, Y., & Mebarki, A. (2019). Automatic structural design of RC wall-slab buildings using a genetic algorithm with application in BIM environment. *Automation in Construction*, 106, 102901.

Talwalkar, M. M. A. R. A. (2018). *Foundations of Machine Learning*. MIT Press.

Van der Linden, L. P. L. (2018). *Parametric Engineering: Course material*. TU Delft.

Veenendaal, J. C. M. V. T. D. (2018). Wat is wat bij parametrisch ontwerpen? *Cement*, 7, 75.

Verhagen, W. J., Bermell-Garcia, P., Van Dijk, R. E., & Curran, R. (2012). A critical review of knowledge-based engineering: An identification of research challenges. *Advanced Engineering Informatics*, 26(1), 5–15.

Viik, L. D. (2019). StructuralComponents 6: An early-stage design tool for flexible topologies of mid-rise concrete buildings. Master's thesis, Delft University of Technology.

Wang, L., Shen, W., Xie, H., Neelamkavil, J., & Pardasani, A. (2002). Collaborative conceptual design - state of the art and future trends. *Computer-aided design*, 34(13), 981–996.

Yang, X., Koehl, M., & Grussenmeyer, P. (2018). Automating parametric modelling from reality-based data by Revit API development. In D. Gonzalez-Aguilera (Ed.), *Latest Developments in Reality-Based 3D Surveying and Modelling* (pp. 307–325).: MDPI.

Zawadzki, P. (2018). Methodology of KBE system development for automated design of multivariant products. In *Advances in Manufacturing* (pp. 239–248). Springer.

Zhang, L. & El-Gohary, N. M. (2018). Automated IFC-based building information modelling and extraction for supporting value analysis of buildings. *International Journal of Construction Management*, (pp. 1–20).

Software References

Aggarwal, A. (2019). *Efficiently working with large data sets in Dynamo.* Retrieved from: https://github.com/DynamoDS/Dynamo/wiki/Efficiently-Working-With-Large-Data-Sets-In-Dynamo. Last accessed 07/07/2020.

Allplan Deutschland GmbH (2020). *Data Exchange Formats Allplan 2020.* Retrieved from: https://www.allplan.com/system/data-exchange-formats/. Last accessed 10/01/2020.

Autodesk Inc. (2018). *Revit Programming for beginners: Easily access the Revit API using free tools.* Retrieved from: https://www.autodesk.com/autodesk-university/class/Revit-Programming-Beginners-Easily-Access-Revit-API-Using-Free-Tools-2018. Last accessed 04/11/2019.

Autodesk Inc. (2019a). *Revit. Functies. Building Information Modeling-software.* Retrieved from: https://www.autodesk.nl/products/revit/features. Last accessed 30/10/2019.

Autodesk Inc. (2019b). *Open source graphical software for design. Explore Dynamo.* Retrieved from: https://dynamobim.org/. Last accessed 30/10/2019.

Autodesk Inc. (2019c). *Revit API Developers Guide. FAQ.* Retrieved from: http://help.autodesk.com/view/RVT/2018/ENU/?guid=Revit_API_Revit_API_Developers_Guide_FAQ_html. Last accessed 04/11/2019.

Autodesk Inc. (2019d). *Revit. Hoogwaardigere en beter gecoordineerde ontwerpen met multidisciplinaire BIM-software.* Retrieved from: https://www.autodesk.nl/products/revit/overview. Last accessed 30/10/2019.

Autodesk Inc. (2019e). *Revit voor bouwtechniek.* Retrieved from: https://www.autodesk.nl/products/revit/structure. Last accessed 30/10/2019.

Autodesk Inc. (2019f). *The Dynamo Primer. Custom Nodes.* Retrieved from: https://primer.dynamobim.org/10_Custom-Nodes/10-1_Introduction.html. Last accessed 31/10/2019.

Autodesk Inc. (2019g). *The Dynamo Primer. Python.* Retrieved from: https://primer.dynamobim.org/10_Custom-Nodes/10-4_Python.html. Last accessed 31/10/2019.

Autodesk Inc. (2020a). *Autodesk Knowledge Network: Loads.* Retrieved from: ttps://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2021/ENU/Revit-Analyze/files/GUID-41B2EE62-B206-45A7-BAFF-BF9B19EAC07A-htm.html. Last accessed 07/07/2020.

Autodesk Inc. (2020b). *Dynamo Package Manager.* Retrieved from: https://dynamopackages.com/. Last accessed 10/01/2020.

Davidson, S. (2019a). *Grasshopper. Algorithmic Modeling for Rhino.* Retrieved from: https://www.grasshopper3d.com/. Last accessed 31/10/2019.

Davidson, S. (2019b). *Scripting and Code Tutorials.* Retrieved from: https://www.grasshopper3d.com/page/scripting-and-code-tutorials. Last accessed 31/10/2019.

Digital Structures (2019). *GH Python Remote*. Retrieved from: https://www.food4rhino.com/app/gh-python-remote. Last accessed 31/10/2019.

Dlubal Software GmbH (2019). *Building Information Modeling (BIM).* Retrieved from: https://www.dlubal.com/en/solutions/application-areas/building-information-modeling-bim/what-is-bim. Last accessed 10/01/2020.

Graphisoft (2019). *Import/Export File Formats in ARCHICAD.* Retrieved from: https://helpcenter.graphisoft.com/knowledgebase/25765/. Last accessed 10/01/2020.

InterCAD (2019). *AxisVM features.* Retrieved from: https://axisvm.eu/axisvm_products_overview.html. Last accessed 10/01/2020.

Matrix Software (2019). *MatrixFrame versie 5.4 SP4 - Release notes.* Retrieved from: https://www.matrix-software.com/updates/mxf/mxf54/prof/NL/MatrixFrame-5.4-release-note.pdf. Last accessed 10/01/2020.

Matrix Software (2020). *MatrixFrame Parametric Design.* Retrieved from: https://www.matrix-software.com/matrixframe_parametric_design. Last accessed 10/01/2020.

McNeel Europe (2020). *food4Rhino - Apps for Rhino and Grasshopper.* Retrieved from: https://www.food4rhino.com/. Last accessed 10/01/2020.

Pratap, A. (2017). *Dynamo BIM: Think Big: Geometry with Lots of Zeros.* Retrieved from: https://dynamobim.org/think-big-geometry-with-lots-of-zeros/. Last accessed 07/07/2020.

SCIA (2018). *SCIA Engineer 18 - Links met Tekla Structures en Revit.* Retrieved from: https://www.scia.net/nl/support/webinars/scia-engineer-18-links-tekla-structures-revit. Last accessed 24/01/2019.

SCIA (2020a). *SCIA Engineer - Features.* Retrieved from: https://www.scia.net/en/software/scia-engineer/key-features. Last accessed 10/01/2020.

SCIA (2020b). *SCIA Engineer - XML format description*. Retrieved from: https://help.scia.net/17.0/nl/tb/xml_format/xml_book.html. Last accessed 04/07/2020.

Sofistik AG (2019). *Sofistik 2020 New Features.* Retrieved from: https://www.sofistik.com/fileadmin/user_upload/Products/Highlights_V2020/New_Features_SOFiSTiK_2020_en.pdf. Last accessed 10/01/2020.

Talarico, G. (2020). *Revit API Docs: Online Documentation for the Revit API.* Retrieved from: https://www.revitapidocs.com/. Last accessed 07/07/2020.

Trimble (2019a). *Compatible file formats and software with Tekla Structures.* Retrieved from: https://teklastructures.support.tekla.com/2019i/en/int_compatible_software. Last accessed 10/01/2020.

Trimble (2019b). *Guides and Articles. Get started with Tekla Structures Open API.* Retrieved from: https://developer.tekla.com/tekla-structures/documentation/get-started-tekla-structures-open-api. Last accessed 04/11/2019.

Appendices

A

Documentation interviews

A.1. Interview 1-3: External structural engineers

(focused on automation)

Questions

Dutch:

- Wat voor berekeningen/processen zijn geautomatiseerd/automatiseren jullie?
- Maken jullie daarbij ook gebruik van BIM (modellen) of informatie uit deze modellen?
- Hoe staan jullie tegenover het gebruik van constructieve software in het algemeen?
- Wat zien jullie als de problemen/knelpunten van het automatiseren van constructieve berekeningen?

English:

- Which calculations/processes are automated/do you automate?
- Do you use BIM (models) for this or information from those models?
- How do you view the use of structural engineering software in general?
- What do you view as the most important challenges of automating structural calculation?

Transcripts of the interviews can be obtained from the author.

A.2. Interviews 4-6: Internal structural engineers

(focused on schematisation)

Questions

Dutch:

- Hoe pak jij het schematiseren aan in een gewichtsberekening? [*besproken aan de hand van een gemaakte gewichtsberekening*]
    - Wat is je aanpak? Welke stappen doorloop je?
- Waarom en wanneer maak je bepaalde schematiseringkeuzes? [*besproken aan de hand van een gemaakte gewichtsberekening*]
- Automatiseren helpt om processen minder tijdrovend te maken, fouten te verminderen en beter te organiseren [*vraag op reactie uit vindingen literatuur*]
    - Zou je kunnen aanwijzen welk aspect/deel van de huidige gewichtsberekening, het meest tijdrovend is?
    - En waar is foutbeperking te behalen binnen het huidige proces?
    - En waarvan jij denkt dat er beter kan aan de huidige organisatie van de gewichtsberekening?

English:

- How do you approach the schematisation part of the vertical load calculation? [*discussed based on an actual vertical load calculation*]
    - What is your approach? Which steps do you take?
- Why and when do you make certain design choices in the process of schematisation? [*discussed based on an actual vertical load calculation*]
- [*From literature review*] Automation helps to make processes less time-consuming, reduce errors and improve the organisation of processes [*asked engineers how they viewed this*]
    - Could you indicate an aspect/part of the current organisation of the vertical load calculation that is the most time-consuming?
    - And where can the introduction of errors be reduced in the current process?
    - And where do you think the current organisation of the vertical load calculation can be improved?

Transcripts of the interviews can be obtained from the author.

A.3. Thematic analysis

[Pn] = Participant number n
[C] = Company of participant
[I] = Interviewer

Theme: Current practice

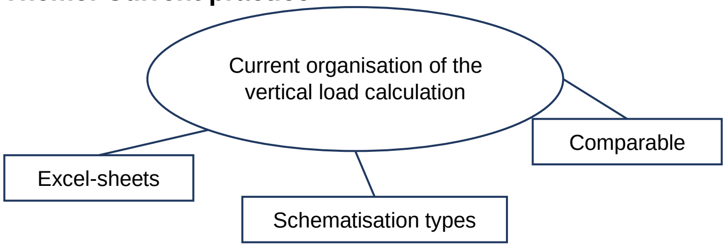


Figure A.1: Sub-theme: Current organisation of the vertical load calculation

Excerpts from dataset:

- Excel-sheets
  - "Bij [*C*] gebruiken ze nu ook Excel sheets voor de gewichtsberekening". (Interview 3)
- Schematisation types
  - "[*P*] geeft aan de schematisering … voor de gewichtsberekening te herkennen". (Interview 1)
  - "2D-plaat kan ook nog [*als schematiseringstype*]". (Interview 1)
  - "[*P*] benadrukt dat in zijn opinie wel in ieder geval een 2D schema nodig is". (Interview 3)
- Comparable
  - "[*P*] geeft aan … [*de*] procedure voor de gewichtsberekening te herkennen". (Interview 1)
  - "[*P1*] en [*P2*] kunnen zich vinden in wat [*I*] besproken heeft". (Interview 2)

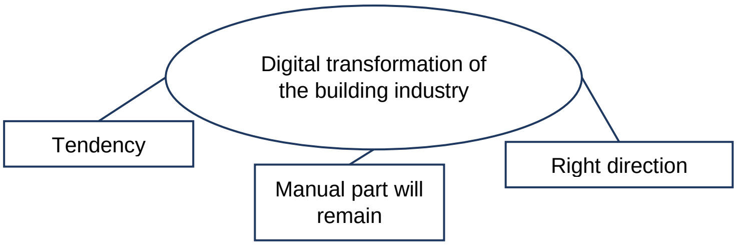


Figure A.2: Sub-theme: Digital transformation of the building industry

Excerpts from dataset:

- Tendency
  - "En verder zit hij ook wel dat het er aan zit te komen dat er steeds meer geautomatiseerd gaat worden, maar of dat binnen 6 maanden is of 3 jaar weet hij niet". (Interview 1)
  - "Toekomstperspectief van [*P1*] en [*P2*]: Het gaat wel die kant [*van automatisering/digitalisering*] uit". (Interview 2)
  - "Digital tranformation is wel een ding dat gaande is, als je nu niet inhaakt een dan mis je wat als bedrijf". (Interview 3)
- Manual part will remain
  - "[*P1*] denkt dat er altijd wel gewichtsberekeningen zullen blijven die met de hand gedaan worden. [*P2*] weet dat niet, als je een oplossing voor het informatie probleem kan vinden …" (Interview 2)

- Right direction
  - “Het is de moeite waard om te gaan automatiseren (de initiële investering verdient zich terug), als je berekening in een half uur kan doen in plaats van in een halve dag/één dag/drie dagen”. (Interview 3)

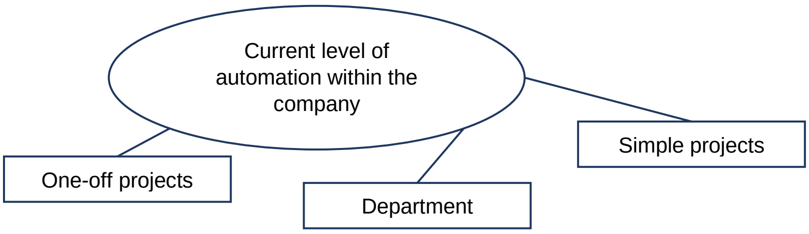


Figure A.3: Sub-theme: Current level of automation within the company

Excerpts from dataset:

- One-off projects
  - “Binnen [*C*] zijn ze nog niet (veel) bezig met automatiseren van berekeningen. [*Dat*] komt ook omdat ze vaak projecten aannemen met bijzonderheden, gekke uitkragingen/overspanningen -> one-off cases -> daarbij is ingenieursinzicht vereist, complexer om te automatiseren”. (Interview 1)
- Simple projects
  - “Huidige situatie [*C*] -> nog niet [*geautomatiseerd*] omdat veel (simpele, kleinere) projecten met de hand veel makkelijker en inzichtelijker te berekenen zijn”. (Interview 2)
- Department
  - “Automatiseringsprocessen bij [*C*] … via computational design afdeling van [*C*]”. (Interview 3)

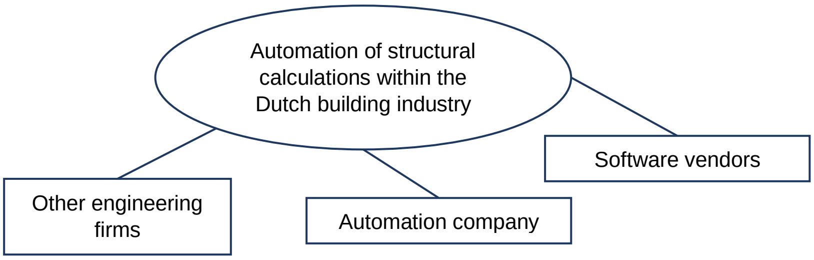

Automation of structural calculations within the Dutch building industry
Other engineering firms
Automation company
Software vendors

Figure A.4: Sub-theme: Automation of structural calculations within the Dutch building industry

Excerpts from dataset:

- Other engineering firms
  - “[*P*] kent geen voorbeeld van gewichtsberekening specifiek, maar ziet wel dat de meer standaard dingen steeds meer geautomatiseerd worden. [*P*] noemt [*andere Nederlandse ingenieursbureaus*] die heel erg met automatisering bezig zijn”. (Interview 1)
  - “Hij noemt ook [*een ander Nederlands ingenieursbureau*], die het hele ontwerpproces voor ‘standaard’ woningbouw [*rijtjeshuizen/twee-onder-één-kap/etc.*] automatiseren”(Interview 1)
- Software vendors
  - “Analytische lijn van Revit is al steeds meer als de schematisering moet zijn. MatrixFrame en SCIA Engineer zijn bezig om het automatisch genereren op basis van het analytische model in Revit te verbeteren.” (Interview 2)
- Automation company
  - “Niet gewichtsberekening specifiek, maar [P] noemt bedrijf [*die automatiseringstools opzet*] en wordt ingehuurd door [*verschillende Nederlandse aannemers*] voor het automatiseren van berekeningen.” (Interview 3)

Theme: Challenges & limitations of automation using BIM

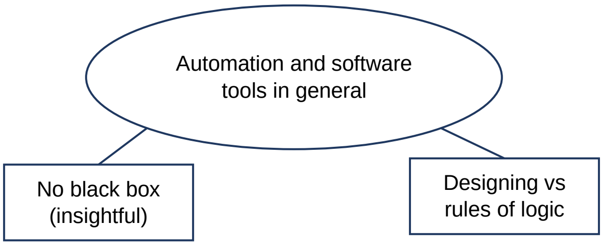

Automation and software tools in general
No black box (insightful)
Designing vs rules of logic

Figure A.5: Sub-theme: Automation and software tools in general

Excerpts from dataset:

- No black box (insightful)
  - "… die hele ontwerpproces voor 'standaard' woningbouw automatiseren; -> Nadeel/kritiekpuntje, het hele proces wordt ook voor je gedaan, je krijgt niet te zien hoe ze bij de rapportage komen die er uiteindelijk uitrolt en of het hele proces dus wel volledig geautomatiseerd is". (Interview 1)
  - "[*P1*] … Nog een reden waarom constructeurs niet van tools houden (klopt het wel wat eruit komt rollen) black box verhaal. Geen black box, zorg dat het inzichtelijk wordt wat er gebeurt voor de gebruiker." (Interview 2)
  - "Inzichtelijk voor gebruiker" (Interview 3)
  - [*over software*]: "Ik toets ook altijd aan het begin een paar dingen van de vervorming en even de permanente belasting, van snap ik het zelf en dan pas vertrouw ik op die uitkomst, gewoon met een kladblaadje en een rekenmachine, .., van dit moet er ongeveer uitkomen, wijkt het minder dan 10% af dan is het fijn, en dan vertrouw ik het". (Interview 6)
- Designing vs rules of logic
  - "Kun je wel alles programmeren?" (Interview 2)

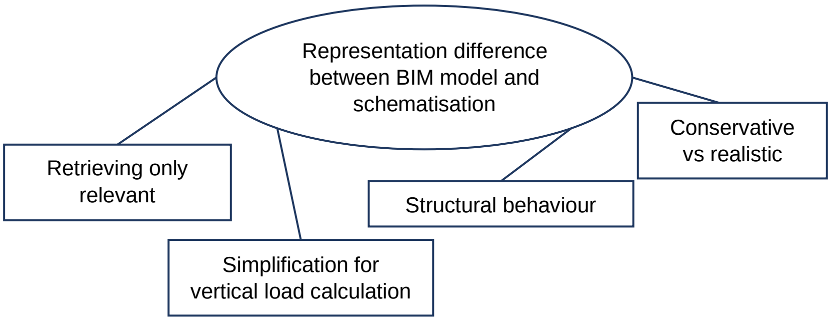

Representation difference between BIM model and schematisation
Retrieving only relevant
Conservative vs realistic
Structural behaviour
Simplification for vertical load calculation

Figure A.6: Sub-theme: Representation difference between BIM model and schematisation

Excerpts from dataset:

- Retrieving only relevant information
  - "In je rekenmodel hoeft niet alles uit je BIM model te staan, bepaalde onderdelen wil je niet in je rekenmodel hebben (koude-bruggen bijv.., etc.)."(Interview 1)
- Simplification for vertical load calculation
  - "Versimpeling in berekening omdat praktischer -> voorbeeld -> positie kolommen, doorsnede wordt kleiner met toenemende hoogte [*analytische lijn in hart doorsnede verspringt*], in een gewichtsberekening voor je gehele gebouw wil je niet alle lokale effecten door bijv. een kleine excentriciteit meenemen." (Interview 1)

  - [*P2*] was met zijn [*reken*]model veel meer moeite en tijd kwijt, door de manier waarop rekensoftware werkt (effecten hoogbouw vervormingen vloeren/kolommen lagere verdiepingen, zorgen voor extra effecten) dan [*P1*] die met een simpele schematisering zo klaar was. In het echt heb je geen last van deze effecten, omdat ze daar in de bouw rekening mee houden, in een ("simpele") schematisering houd je er ook rekening mee, in een compleet rekenmodel van een gebouw doe je dat niet." (Interview 2)
  - " [*P*]…er hier een apart vloertje zit, dat wordt nu even buiten beschouwing gelaten …, om redenen van eenvoud, omdat je het toch wel ziet, ik zit hier toch al met drie lijnlasten en een aantal puntlasten. [*I*]: Dat zal niet uitmaken. [*P*]: Voor de fundering. Het is een bovengrens, die je nu kiest." (Interview 4)
  - *"*…, nu heb je zeg maar 4 q-lasten, maar dat waren er ook wel zeven ofzo, toen hebben we wel echt gezegd, dat moeten we wel gaan beperken, want dan heb je een q-last, van de ene is 90 en de andere is 92, ik noem maar iets. En dat is echt niet interessant, dat verschil, dus daar hebben we wel bewust de keuze gemaakt, van we trekken dat gewoon even gelijk." (Interview 6)
- Structural behaviour
  - "… bepaalde keuzes van de constructeur vanwege gedrag constructie -> Wand met opening (bijv. van kern) wordt door modelleur gemodelleerd als één wand en daar wordt vervolgens een sparing ingezet -> terwijl je voor de berekening 3 elementjes wil hebben, …, omdat dat balkje [*latei boven deuropening*] zich anders gedraagt dan het wandgedeelte -> lagere E-modulus beton (op basis van ervaring bijvoorbeeld)."(Interview 1)
  - "hier hebben we dan het windmoment gerekend over die hele wand, maar hier zit wel een opening met een lateitje, dus iets wat je in die beginfase al moet checken is of dat moment ten gevolge van die windbelasting daar doorheen kan, want als dat niet zo is, dan ga je in het model al verder met hier een scharniertje, dan is hier eigenlijk al de vraag hoe neem je die latei mee?" (Interview 5)
- Conservative versus realistic
  - "… constructieve veiligheid inbouwen, waardevol aan een gewichtsberekening." (Interview 1)
  - "Constructeursslimmigheden(= ook ervaring)/veiligheid/robuustheid) -> Naar boven afronden, meest negatieve situatie overal meenemen. … " (Interview 2)
  - "Conservatief wel makkelijk, snel met de hand, maar niet een reden waarom je dat ook in je automatisering wil hebben ([*P*] noemt het zelfs laf). Realistisch -> als het toch kan via de computer waarom zou je dat niet doen, indien je zorgt dat de informatie juist in de computer staat dan zijn de uitkomsten alleen maar accurater." (Interview 3)
  - [*Over gedetailleerdere berekeningen maken in een gewichtsberekening*]: "… denk ik wel bepaalde delen veel complexer dan ze soms misschien, zouden zijn, dus dat is wel weer een uitdaging, om dat niet te complex te maken."(Interview 4)
  - [*Over tweede orde factor*] "Je moet ook niet met 1,2 gaan rekenen als je het 1,1 kan zijn, dan ben je ook iets te conservatief bezig. Dat was hierbij [*doelend op voorbeeld project*] ook nog wel een dingetje … Ja en omgekeerd als je met 1,1 rekent en je krijgt later nog vragen waaruit blijkt dat het eigenlijk 1,2 had moeten zijn…. Je moet dat zelf helder hebben…" (Interview 6)
  - "[*I*]: …als je alles direct uit Revit haalt wordt het heel exact, maar dat wil je niet altijd. [*P*]: Ja, en dat is ook zo, wat erin gestopt wordt, of dat dan ook, …, dat is ook wel het gevaar bij 3D-modellen, als het heel complex wordt, snap je de uitkomst nog." (Interview 6)

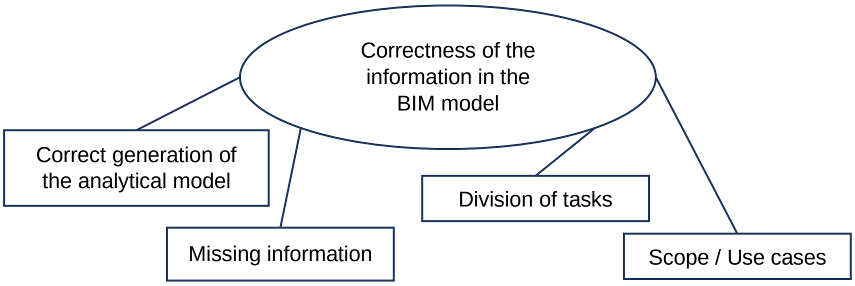

Correctness of the information in the BIM model
Correct generation of the analytical model
Missing information
Division of tasks
Scope / Use cases

Figure A.7: Sub-theme: Correctness of the information in the BIM model

Excerpts from dataset:

- Correct generation of the analytical model
    - “Vaak vereist/vereiste Revit naar rekensoftware omzetting nog veel voorbereidingswerk.”(Interview 2)
    - “… aansluitingen van analytische lijnen daar heb je mee te maken.” (Interview 3)
    - “… BIM model en analytisch model aan elkaar vastknopen.” (Interview 3)
- Missing information
    - “Mogelijk probleem bij automatiseren gewichtsberekening -> gebrek aan info -> beperkte info… Wat doe je met ontbrekende informatie? Hoe ga je daarmee om?” (Interview 2)
    - “Hoe ga je rekening houden met informatie die mist(/verandert)?” (Interview 3)
- Division of tasks
    - “Goed opstellen wat voor informatie de verschillende rollen (constructeur/modelleur) bieden, welke informatie de verschillende rollen er op een bepaalde manier in gaan stoppen.” (Interview 3)
- Scope / Use case(s)
    - “Goed vastleggen in je framework, wat voor de scope (omvang, strekking, bereik), waar specifiek je tool voor werkt: Hoe zit het met je stabiliteit, [*P*] raadt aan dit niet mee te nemen, op een gegeven moment wordt je gebouw dusdanig hoog dat (afhankelijk van je systeem) je stabiliteitsberekening bepalende wordt voor je paalkrachten. “ (Interview 1)
    - “Goed scope voor je protocol afbakenen.” (Interview 2)
    - “Gevoeligheid script -> script moet om kunnen gaan met alle input -> hergebruik van het script”. (Interview 3)

Theme: Functionalities of a tool that automates the vertical load calculation

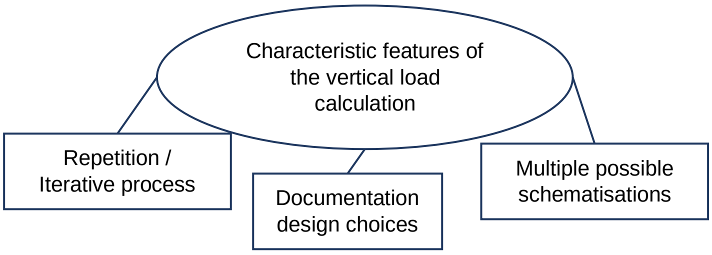

Characteristic features of the vertical load calculation
Repetition / Iterative process
Documentation design choices
Multiple possible schematisations

Figure A.8: Sub-theme: Characteristic features of the vertical load calculation

Excerpts from dataset:

- Repetition / iterative process
  - "… meenemen van repetitie binnen een project, meerdere keren dezelfde schema's binnen hetzelfde gebouw." (Interview 2)
  - "Aansluitend op repetitie binnen gebouwen, kan meerdere dezelfde assen zijn, maar bijvoorbeeld ook meerdere keren dezelfde vloeren, dat kan je in één keer schematiseren."(Interview 2)
  - "[*P1 & P2*] zien voordeel repetitie in, je kan nog eens snel, of bij een wijziging je aangepast BIM model erdoorheen gooien (ook nuttig/interessant bij hergebruik of nieuwe lasten (bijv. zonnepanelen) + dus ook de repetitie binnen je project." (Interview 2)
  - "Ja, het is natuurlijk in een bepaalde mate, een beetje iteratief. En daar is natuurlijk ook de vraag, in zo'n iteratief proces, daar is wel een hoop winst te behalen wanneer je het automatiseert." (Interview 4)
  - "En je zit natuurlijk ook in een iteratief proces". (Interview 6)
- Documentation design choices
  - "Als je het 3D model wil gebruiken voor berekeningen, dan moeten daar ook ontwerpkeuzes van de constructeurs in worden aangegeven. " (Interview 3)
  - "… zwart-op-wit hebben staan. Maar dan heb je er over nagedacht, dan is het gewoon gedocumenteerd."(Interview 6)
- Multiple possible schematisations
  - "Als jezelf een aanname doet, waarom doe je dan die aanname (er zijn meerdere mogelijkheden, het is niet zo dat één gewichtsberekening de goede is)." (Interview 2)
  - "… en natuurlijk valt daar altijd over te twisten, over een lijnlast nou net even iets anders gekozen zou moeten worden?" (Interview 4)

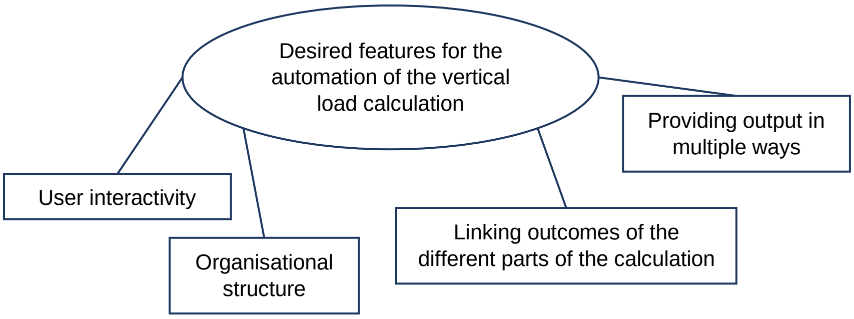

Desired features for the automation of the vertical load calculation
Providing output in multiple ways
User interactivity
Organisational structure
Linking outcomes of the different parts of the calculation

Figure A.9: Sub-theme: Desired features for the automation of the vertical load calculation

Excerpts from dataset:

- User interactivity
  - " [*P*]: … ik weet ook niet of het dan verstandig is om het in BIM automatisch te laten doen. Maar een keuzeoptie zou veel prettiger zijn. [*I*]: … die keuze … die moet er nog wel zijn. [*P*]: … die moet niet volledig verdwijnen. Dat je wel nog weet waar je mee bezig bent. En dat je dat nog wel kan afwegen. " (Interview 5)
  - [*over schematiseringskeuzes*] "[*P*]: … dat zijn wel keuzes die… [*I*]: Die je zelf moet maken nog." (Interview 6)
- Organisational structure
  - "Bij [*C*] gebruiken ze nu ook Excel sheets voor de gewichtsberekening (ook een soort automatisering), in [*P*] zijn opinie gebeurt dit nu al redelijk efficient." (Interview 3)
  - "…voor iedere as maak je in ieder geval een schema, tenzij het een hele simpele as is dan kan het misschien ook op een makkelijkere manier. Dan zien we hier meerdere schema's per as, dat is wel een hoop, in dit geval een hoop losse schema's die bij elkaar komen, en op het moment dat je dat wat meer gebundeld krijgt, is dat ook weer makkelijker." (Interview 4)
  - "… gewoon een duidelijker stroomschema … ook met veel jonge mensen nu in dienst, dat is wel belangrijk, dat het dan gewoon duidelijk is." (Interview 6)
- Linking outcomes of the different parts of the calculations
  - "Op het moment dat jij die berekening al een heel end hebt liggen, is het natuurlijk makkelijker om aan te passen, als dat allemaal automatisch gaat." (Interview 4)
  - "…als je een wand hebt die bestaat uit verschillende lagen [*wanden die zijn gesplitst in meerdere 1D schema's*]. Bijvoorbeeld als deze [*wijst een wand met meerdere 1D schema's aan*] dan veranderen die reactiekrachten [*vanuit het bovenste 1D schema*] allemaal, maar die staan hierop [*het onderste 1D schema*], dus dan moet je dat tweede model ook weer wijzigen, en dat op zich veel werk." (Interview 5)
- Providing output in multiple possible ways
  - "…met je gewichtsberekening nu, berekening je je fundering. een deel daarvan zal je gebruiken om wandwapening te berekenen. Op het moment dat jij een hele uitgebreide berekening hebt waar je alle berekeningen uit kan trekken, …, en dat kost misschien net wat meer tijd dan allemaal verschillende losse berekeningen, dan heeft dat natuurlijk wel meerwaarde…" (Interview 4)
  - "…Er is dan een andere partij die die wanden uitwerkt, maar zij willen die belastingen per verdieping hebben, en dat hebben wij nu natuurlijk niet, omdat we gewoon een lijnlast opgeven voor het totaal, omdat wij één Technosoft modelletje hebben [*van een 1D schema*], helemaal onderaan die wand, dus dan moet je eigenlijk heel dat schema weer opknippen per verdieping, terwijl je dat eigenlijk hier [*wijst voorbeeld van andere bedrijf aan*] al zo hebt staan. Alleen het enige wat je hier dan weer niet in hebt is dat je niet echt een duidelijk aanzicht hebt van hoe de wand in elkaar zit en vooral die windbelasting meegenomen wordt… (Interview 5).

Theme: Schematisation

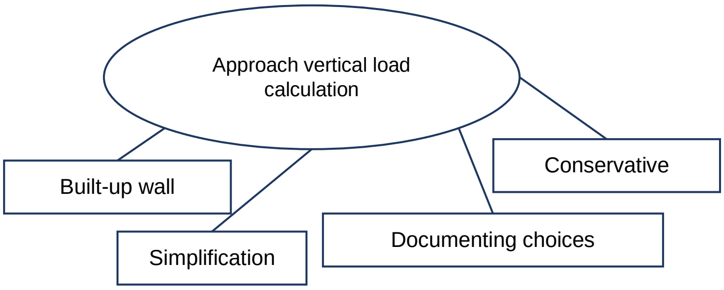


Figure A.10: Sub-theme: Approach vertical load calculation

Excerpts from dataset:

- Built-up wall
  - "… dan is het een kwestie van waar zit mijn gevel?...en als je die van boven doorzet, dat geeft dan een puntlast op een bepaalde locatie." (Interview 4)
  - "… van waar zitten mijn steunpunten van mijn wanden, hoeveel belasting gaat er heen, …" (Interview 5)
- Simplification
  - [*P*]: "… dat er hier [*wijst voorbeeldproject aan*] een apart vloertje zit, dat wordt nu even buiten beschouwing gelaten…, om redenen van eenvoud." [*I*]: "Dat zal niet heel veel uitmaken…" [*P*]: "Voor de fundering. Het is een bovengrens, die je nu kiest." (Interview 4)
  - "… ik zou zeggen maximaal 3 of 4 q-lasten, tenzij er een hele goede reden voor is om daar vanaf te wijken." (Interview 6)
- Documenting choices
  - "Je moet dat zelf helder hebben, en uitgezocht, en zwart-op-wit hebben staan. Maar dan heb je er over nagedacht, dan is het gewoon gedocumenteerd." (interview 6)
- Conservative
  - "…waar je van tevoren eigenlijk ook al rekening mee moet houden is dat je van tevoren niet heel nauwkeurig iets uit moet rekenen, maar dat je ook redelijk wat marge meeneemt."(Interview 5)
  - [*over tweede orde berekeningen*] "dat moet je later nog wel nagaan. Je moet ook niet met 1,2 gaan rekenen als het 1,1 kan zijn…" (Interview 6)

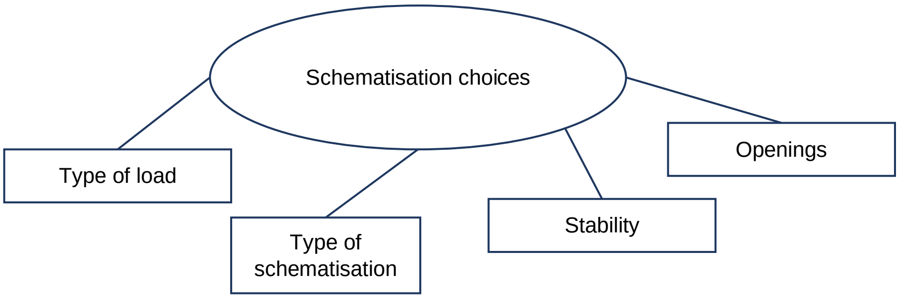


Figure A.11: Sub-theme: Schematisation choices

- Type of load
  - "…dat als je iets als een puntlast schematiseert, het meteen een hele duidelijke impact heeft, in je berekening, en als je voor een q-last kiest, of dat nou voor de wind is of voor een gewicht, dan spreidt zich dat veel meer. En geeft dat ook een ander beeld van je reactiekrachten, dan wanneer je voor een puntlast kiest." (Interview 4)

- Type of schematisation
    - "…we hebben gekozen om door de kelder te knippen [*twee losse 1D schema's*], en het bovenste deel schematiseren we als een wand van volgens mij 6 meter hoog uit mijn hoofd, die ondersteund wordt door een aantal veren ter plaatse van een kolom en de wand, …, als we die berekening maken komen daar de reactiekrachten uit… En die reactiekrachten nemen we vervolgens mee op het onderste schema." (Interview 4)
    - "… wanneer je een wand hebt met heel veel ingewikkelde sparingen, en wanneer het in vrij ingewikkeld proces wordt om dat te vereenvoudigen, dan zou je daar al sneller een 2D-model bij maken…" (Interview 5)
- Stability
    - "Voor stabiliteit, hebben de wanden waar gevelopeningen zitten eigenlijk geen functie, er is voor gekozen om de stabiliteitskracht te verdelen over het in deze wand drietal aanwezige elementjes. Waar we hier voor kiezen is om het moment er echt direct op te zetten. Terwijl we bij de andere wanden, die eigenlijk allemaal veel dichter zijn, we ervoor kiezen om het als een driehoekige q-last mee te nemen, omdat we daar veel meer spreiden over de hele lengte. Terwijl het in die eerdere as allemaal veel meer lokaal is. En dat we daarin zelf een bepaalde verdeling bepaling aan de hand van de stijfheid van de bepaalde wandelementjes." (Interview 4)
    - "…hier hebben we dan het windmoment gerekend over die hele wand, maar hier zit wel een opening met een lateitje, dus iets wat je in die beginfase al moet checken is of dat moment ten gevolge van die windbelasting daar doorheen kan, want als dat niet zo is, dan ga je in het model al verder met hier een scharniertje, dan is hier eigenlijk al de vraag hoe neem je die latei mee?" (Interview 5)
- Openings
    - "…er is hier voor gekozen om gewoon die hele wand dicht te rekenen, en om de sparingen die erin zitten…als puntlast omhoog aan te nemen, zo wordt gewicht er vanaf getrokken…" (Interview 4)
    - "…er is ook een vrij groot verschil moet ik zeggen, hoe je zo'n som aanpakt voor een dichte wand, zoals deze [*wijst voorbeeld as 9 aan*] of voor een wand met allerlei grote sparingen op de begane grond…" (Interview 6)

Theme: Advantages of automation related to the vertical load calculation

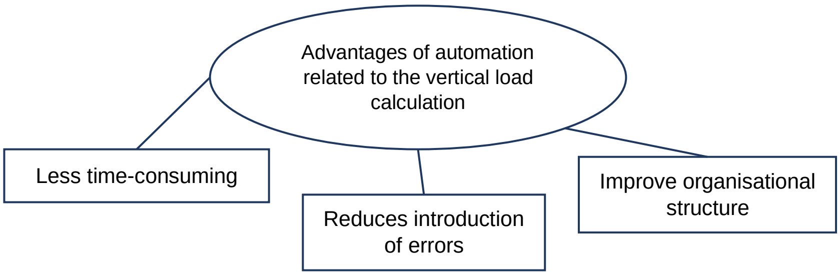

Advantages of automation related to the vertical load calculation
Less time-consuming
Reduces introduction of errors
Improve organisational structure

Figure A.12: Theme: Advantages of automation related to the vertical load calculation

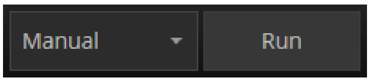
- Less time-consuming
  - "…na je schematisering, start je natuurlijk pas met je berekening, en het gebeurt natuurlijk ook, dat op het moment dat je berekening gaat doen, iets ziet wat je niet van tevoren bedacht hebt, waardoor je je schematisering weer gaat aanpassen, dus dat blijft natuurlijk wel een loopje…het is natuurlijk in een bepaalde mate, een beetje iteratief. En daar is natuurlijk ook de vraag, in zo'n iteratief proces, daar is wel een hoop winst te behalen wanneer je het automatiseert." (Interview 4)
  - "Sowieso met het schematiseren, van hoe ga je dat aanpakken. Maar dat is net als wat je zei niet één rechte lijn, maar vooral ook een terugkoppeling." (Interview 5)
  - "Wat ik denk dat het meeste tijd kost is dat we in een vrij vroeg stadium allemaal aannames doen en wel in een DO al een palenplan helemaal tekenen, zonder dat echt die schematisering al helemaal goed is gedaan, dus volgens mij gaat daar veel tijd mee verloren, dat je dat al hebt verwerkt en dat je dat dan later weer moet oppakken, bij voorkeur door dezelfde constructeur, dat je het eigenlijk weer opnieuw gaat doen. Je moet ook niet in dat hele vroege stadium alles wel al hebben uitgewerkt. Ik denk dat daar heel veel tijd mee verloren gaat. Ook omdat het dan een tijd weer stil ligt. En er kan ook tijd verloren gaan door in een vroeg stadium juist, wel al te veel te doen en dan verandert er weer iets." (Interview 6)
- Reduces introduction of errors
  - "…wat ik denk hoe het nu gebeurd, één as maak je helemaal af en dan controleer je die as, en dan ga je daar variatie in aanbrengen. En als je dan die eerste keer een fout gemaakt hebt, dan kopieer je je foutje ook door, en dan moet je dat terug, in alle assen weer gaan aanpassen, dus dat is een bepaalde gevoeligheid die erin zit. Dat je bepaalde checks hebt, na je eerste as controleren, op het moment dat die goed is gaan we pas verder kopiëren, want je wil niet fout op fout op fout door kopiëren. Wat natuurlijk zou kunnen is dat op het moment dat je zegt van hé, in die hele rij assen heb ik één foutje ontdekt, ik pas hem aan bij één, dat het dan in één keer in alle tien gebeurd, in plaats van dat je dat per as weer terug moet gaan doen, en daar is natuurlijk ook wel winst te behalen.“ (Interview 4)
  - "…met als je dit opsplitst [*wijst weer voorgaand voorbeeld aan*] en je moet het weer veranderen, en al die puntlasten moet je weer wijzigen. Al is dat niet erg ingewikkeld werk, je maakt er wel snel een typefoutje bij…" (Interview 5)
  - "…en qua fouten, ik denk dat wij intern vrij veel controles hebben dat dat eigenlijk nog wel meevalt, maar dat eerder fouten van ons, …, te laat die schematisering goed beetpakken en daar gaat dan weer tijd mee verloren. Want uiteindelijk ligt er wel een goed product, waar je extern meestal niemand over hoort." (Interview 6)
- Improve organisational structure
  - "…voor iedere as maak je in ieder geval een schema, tenzij het een hele simpele as is dan kan het misschien ook op een makkelijkere manier. Dan zien we hier meerdere schema's per as, dat is wel een hoop, in dit geval een hoop losse schema's die bij elkaar komen, en op het moment dat je dat wat meer gebundeld krijgt, is dat ook weer makkelijker." (Interview 4)
  - "…en gewoon een duidelijke stroomschema…gewoon welke stappen volg je…ook met veel jonge mensen nu in dienst, is dat wel belangrijk, dat het gewoon duidelijk is." (Interview 6).

B

Documentation testing

B.1. Manual user tests

Below an overview is given of the user interface of the tool in Dynamo. The user interface consists of several *orange* groups, each indicating a step in the vertical load calculation process in compliance with the flowchart for the automation of the vertical load calculation.

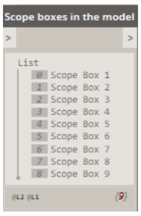


General

Run Dynamo is in *Manual* to assure the script is executed as intended The script can be executed by clicking the *Run*-button or *F5*. This is indicated in the bottom left corner of the screen. Below an overview is given of the different

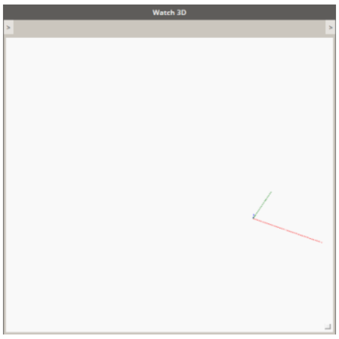
Informative nodes

Watch node
This node provides the user with lists of variables (values, elements, geometry, text).



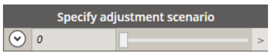
```
List
  0 Scope Box 1
  1 Scope Box 2
  2 Scope Box 3
  3 Scope Box 4
  4 Scope Box 5
  5 Scope Box 6
  6 Scope Box 7
  7 Scope Box 8
  8 Scope Box 9
@L2 @L1                {9}
```

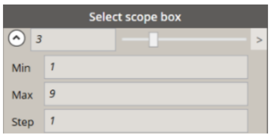
3D watch node
This nodes provides the user with a visualisation of the geometry. Using a varying range of colours, design choices are visualised. If no geometry is visible, use *right-click: Zoom to Fit*, to centre the geometry in the view. The view can be rotated by holding the *right mouse button* and panned by holding de *middle mouse button*.



Nodes to allow for user interaction

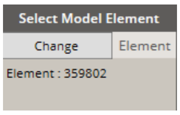
Slider
The slider node can be used to choose between multiple alternatives provided to the user. When the slider is unfolded, the user may change the maximum number to make sure it matches the number of alternatives provided by the tool.



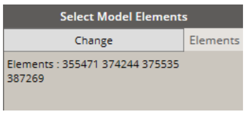


Value: 3
Min: 1
Max: 9
Step: 1

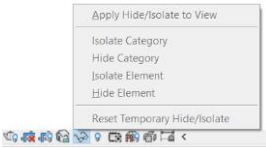
Boolean
The Boolean node is used to *toggle* certain actions that can be performed by the tool (selection/isolation of elements, adjusting of values, etc.). At the top of each of the *orange groups* a Boolean node is present, these node must be set to *True* to be able to continue with that part of the process.



Element : 359802

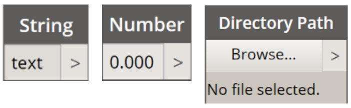


Elements : 355471 374244 375535 387269

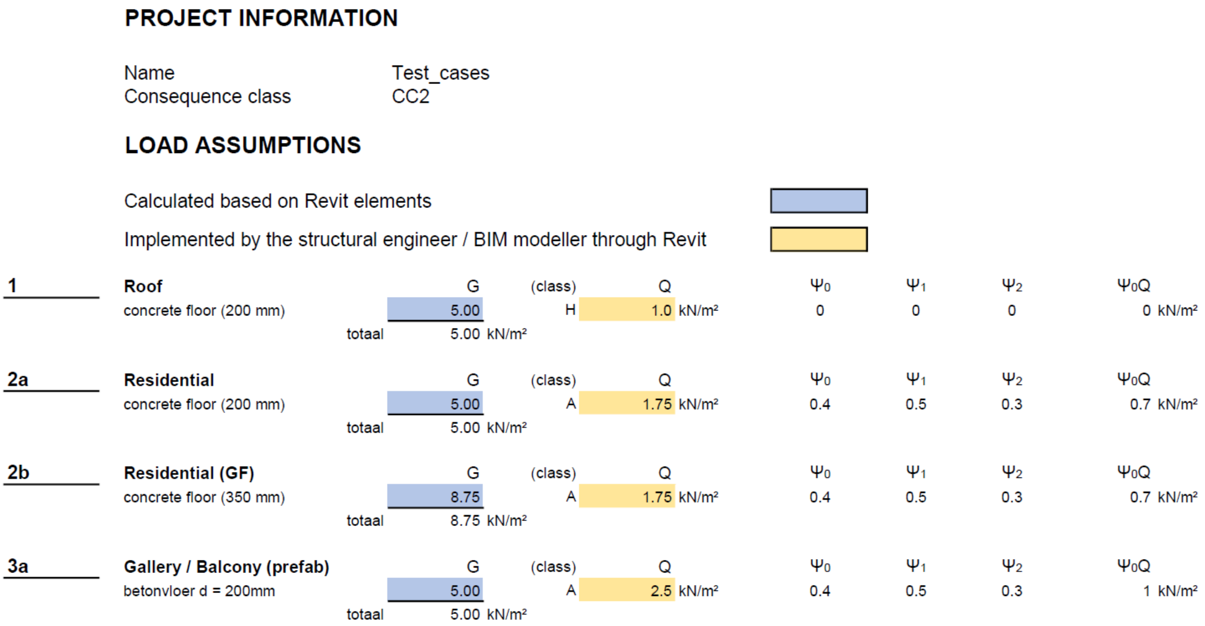


Select Model Element(s)
The select model element(s) nodes can be used to select model elements in the BIM model. To select the model elements the *select/change*-button needs to be clicked. In the Revit user interface the elements can then be selected by clicking on the element (in case of *Select Model Element*) or by dragging a box around the elements (in case of *Select Model Elements*). To make it easier to select the correct elements it is good practice to first *isolate* the elements in the Revit user interface.

Other nodes
The other nodes for user interaction are self-explanatory.



B.2. Load assumptions used for testing



PROJECT INFORMATION

Name	Test_cases
Consequence class	CC2

LOAD ASSUMPTIONS

Calculated based on Revit elements 🟦

Implemented by the structural engineer / BIM modeller through Revit 🟨

			G	(class)	Q	Ψ_0	Ψ_1	Ψ_2	Ψ_0Q
1	**Roof**		G	(class)	Q	Ψ_0	Ψ_1	Ψ_2	Ψ_0Q
	concrete floor (200 mm)		5.00	H	1.0 kN/m²	0	0	0	0 kN/m²
		totaal	5.00 kN/m²						
2a	**Residential**		G	(class)	Q	Ψ_0	Ψ_1	Ψ_2	Ψ_0Q
	concrete floor (200 mm)		5.00	A	1.75 kN/m²	0.4	0.5	0.3	0.7 kN/m²
		totaal	5.00 kN/m²						
2b	**Residential (GF)**		G	(class)	Q	Ψ_0	Ψ_1	Ψ_2	Ψ_0Q
	concrete floor (350 mm)		8.75	A	1.75 kN/m²	0.4	0.5	0.3	0.7 kN/m²
		totaal	8.75 kN/m²						
3a	**Gallery / Balcony (prefab)**		G	(class)	Q	Ψ_0	Ψ_1	Ψ_2	Ψ_0Q
	betonvloer d = 200mm		5.00	A	2.5 kN/m²	0.4	0.5	0.3	1 kN/m²
		totaal	5.00 kN/m²						

Figure B.1: Project information and load assumptions used for testing

B.3. Notes made during user tests

Test 1

- Building Part Selection
  - Number in 3D view -> relation number slider & scope boxes
- Schematisation Choices
  - Extend explanation (schema, asload, support-element)
  - The user should remain in control of the selection process (‘*Select all elements of* category’-feature can lead to this)
  - Clear & useful colouring of the choices in 3D preview
- Adjusting analytical model
  - Indicate in schematisation which element is selected
  - Image clarifying possible adjustment scenarios -> use
- General
  - Smarter organisation of the different nodes in groups -> limitation of the use of Dynamo -> user-friendly -> more explanations is often better
  - Recognises approach framework in tool

Test 2

- Building Part Selection
    - In Revit it is also possible to isolate scope boxes
- Schematisation Choices
    - Image does not change if you do not run the script (Manual mode)
    - *‘Support’*-element can be a wall -> should be impossible
    - *‘All elements of category selection’*-feature -> add a warning
    - Shorter explanation about Dynamo nodes
    - Different colours for different choices in the 3D preview is useful
- Loads Definition
    - Explanation about the reduction of the schema can be improved
    - Useful that it is indicated which elements belong to which load -> but, it needs to be clear how to read the lists
    - Unfortunate that tekst & dimensions cannot be displayed in 3D preview
- Calculation
    - Standard save location of the XML-file
    - Not only design values to SCIA Engineer -> also characteristic values
- General
    - Short overview of the framework would improve the usability of the manual
    - Explanation Revit user interface (hotkeys are useful)
    - Further explanation of Revit is useful (for example: cutouts are separate elements that are invisible)
    - Organisation of the different nodes in Dynamo can be improved