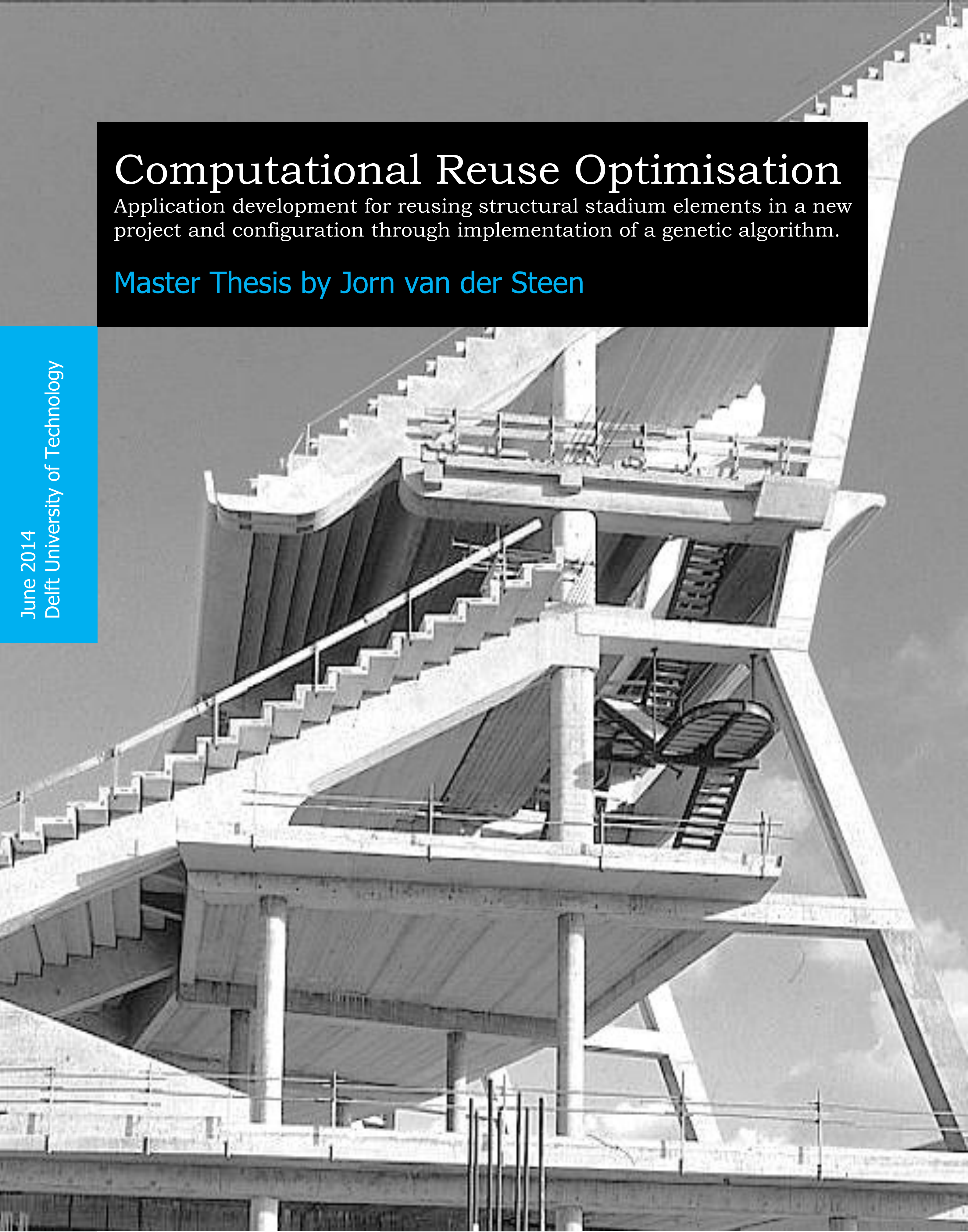


Computational Reuse Optimisation

Application development for reusing structural stadium elements in a new project and configuration through implementation of a genetic algorithm.

Master Thesis by Jorn van der Steen

June 2014
Delft University of Technology

Delft
University of
Technology

Challenge the future

MSc Graduation Thesis

Computational Reuse Optimisation

Application development for reusing structural stadium elements in a new project and configuration through implementation of a genetic algorithm.

Jorn van der Steen

June 2014

BEMNext Lab
Department of Building Engineering
Faculty of Civil Engineering and Geo-sciences
Delft University of Technology
www.tudelft.nl

Published in Delft, The Netherlands

©2014 Jorn van der Steen, Delft University of Technology, The Netherlands.

No part of this document may be reproduced for commercial purposes without written consent from the author. Commercial copying, hiring, lending and selling is prohibited without written consent from the author. For inquiries regarding this thesis please contact the author via e-mail: Jorn.vanderSteen@gmail.com

Colophon

Student Information

Jorn van der Steen
Studentnumber 1361473
Jorn.vanderSteen@gmail.com
Westlandseweg 35
2624 AB Delft

University Information

Delft University of Technology
Faculty of Civil Engineering and Geo-Sciences
Department of Design & Construction
Stevinweg 1
2628 CN Delft
www.tudelft.nl

Company Information

Royal BAM group
Runnenburg 9
3981 AZ Bunnik
www.bam.eu

Lab Information

BEMNext Lab
Stevinweg 1
2628 CN Delft
www.bemnext.org

Committee Information

Prof.dr.ir. J.G. Rots, TU Delft
Dr.ir. J.L. Coenders, White Lioness technologies & TU Delft, BEMNext Lab
Ir. S. Pasterkamp, TU Delft
Ir. A. Rolvink, White Lioness technologies & TU Delft, BEMNext Lab
Ir. J.A.M. van Steekelenburg, Royal BAM Group

Preface

This report, together with the created application, form my Master's thesis for the study Building Engineering. An MSc study at the faculty of Civil Engineering and Geosciences at the Delft University of Technology.
The thesis research focuses on exploring and testing a new strategy for reusing structural elements through an optimisation algorithm. The algorithm seeks to reuse structural elements, while also utilising the structural capacity of the reuseable elements and respecting the spacial requirements of the new structure. Leading to a new functional and material economic structure constructed out of reused elements.
Originally the research focused on finding a new reuse strategy for sport stadiums, during the research the topic shifted to finding a more broadly applicable reuse strategy with the Al Wakrah sport stadium as case to test the strategy.
The research was performed in collaboration with the BEMNext lab, providing computational knowledge for the computational application, and the Royal BAM Group providing input on the structural reuse aspects, environmental impact and a case to test the strategy in practice.
This thesis explores a new direction in its field, aiming to provide the reader with insight in the potential of this reuse strategy.

Jorn van der Steen, Delft, June 2014

Acknowledgments

This thesis is a collaborative product between the university, the BEMNext Lab and the Royal BAM group. This document however solely bares the name of the author. To give credit to those who deserve it, I would like to thank the following people.

Prof.dr.ir. J.G. (Jan) Rots
As chair of the committee you provided me with simple but effective guidance in the global process of the thesis. Your general view and feedback on the project helped me find the thesis direction and assured a more complete final result, thank you.

Dr.ir. J.L.(Jeroen) Coenders
During the project your vision on computational relevance and efficiency amazed me. During our brainstorms on the application design, some of your ideas were so brilliant that I only discovered after a few months how well they worked during implementation. I would like to thank you for these insights.

Ir. J.A.M.(Joost) van Steekelenburg
Joost, during the project I sometimes drifted towards implementing more and more computational gadgets. But thanks to your influence I believe the contribution the final application can have to the engineering world has increased. Furthermore I would also like to thank you for your positive attitude throughout the project, encouraging me to keep going and improving. And I'm sure this characteristic of yours will be of great value to BAM in your new function.

Ir. A.(Anke) Rolvink
Anke, without you I would probably still be stuck somewhere in the code right now. The computational tips you gave me were of great help to link the different software programs together. They also helped me to keep the code workable and become more efficient, allowing me to reach my goal faster. Besides the computational help I would also like to thank you for your tips of making this thesis more readable and encouraging me to get it to a higher level.

Ir. S.(Sander) Pasterkamp
Thank you for sharing your knowledge and sources on the topic of reuse. Especially hosting a synergy meeting with a student performing research in a similar direction. This helped me out allot while setting up the requirements of the application from a theoretical reuse point of view. Also your support in getting the formalities around graduation done were very much appreciated.

Ir. S.F. (Steven) Wilbrenninck
Thank you for making me understand how to sell the project. Especially on the aspects where my more idealistic preferences and the commercial needs did not match. However not always smoothly, these sessions taught me a lot and I believe they will be of use during my career. In return I hope the final product will aid you in future stadium ventures.

Ir. J. (Joris) Hesselink
I would like to express my gratitude in your aid on the practical side of implementing reuse. Your reflection on the connection design and construction process greatly improved the practical possibilities of the project.

BEMNext Lab students
The help provided by you guys during the start of my programming adventures really helped me getting started quickly and your stories helped me to keep pursuing my goals, thank you.

Friends and Family
Dear friends and family, I would like to thank you all for your jokes and support during the process of writing this thesis. Especially I would like to thank my sister. Merle thank you for checking my English during the writing process, you really helped getting this document from barbaric to an elegant and educational worthy level.

Summary

This master's thesis covers the research and design of a computational reuse application. The research starts with reuse incentive investigation: 'Why is it pursued and how should it be applied?' The research continues by identifying the struggles defining reuse, presenting the methods used to overcome these struggles and finally indicating the practical implications for the theoretical strategy.

With this literary background, a strategy has been developed which is implemented in the Grasshopper environment using custom made components. After the model is build and calibrated it is applied to a stadium project in Al Wakrah, for which the owner wants to disassemble the second ring.

In the stadium industry, big sporting events can require new venues, venues not fitting the demand for the total technical lifetime of the structure. When this occurs there are four possible outcomes. One of these outcomes is reuse, which has not been successfully applied in practice. However for a variety of incentives presented in Chapter 2, reuse is sometimes the only option. Therefore this thesis will develop a reuse strategy for the stadium industry.

The reuse strategy creates the possibility to (partly) deconstruct the stadium and create a new structure from these elements. This new structure is not only reusing the available elements but also respects the new project list of requirements and matches the demanded shape.

In doing so, the strategy provides a solution to the main criticism for hosting sporting events with new, overcapacitated stadiums. Simultaneously creating a strategy that allows the design of new structures in a more sustainable manner than currently available in the building industry.

Due to element life time differentiation caused by user requirement changes and because structural elements on average account for 90% of the building's environmental footprint [Naber, 2012]. The literature study concluded that the application focuses on the reuse of structural elements.

Furthermore, based on their generic character, the structural elements that will be reused are columns, beams and floor plates.

The research also indicates that designing with reusable elements strains the design process. Caused by the reusable element set reducing the design freedom and flexibility while increasing the amount of design constraints by the wish to apply the reusable set. These complications are answered to by implementing a genetic algorithm in the application. This type of algorithm moves through the

project solution space in search of the solution best fitting the given boundary conditions.
This search approach finds 'fit' solutions faster than can be expected from testing each possible outcome, through the algorithm components 'crossover', 'mutation', 'selection' and 'elitsm'. Meaning the algorithm is a more effective way of scouting the search space and finding a result satisfactory to the user. And in doing so counters the strain caused in reuse design. However since not each possible outcome is tested, it cannot be said with 100% certainty that the fittest solution in the search space has been found.
Furthermore the application is applied during the preliminary design stage meaning there is a lot of design flexibility. Increasing the chance a satisfactory solution is inside the search space.

The algorithm will be applied on a grid of building sections called subsets. The algorithm parameters consist of the element types assigned to each subset and the amount of subsets in each building direction. These parameters allow the algorithm to search for the optimal configuration using the reusable set, while meeting the floorspace requirements and following the shape set by the building mass.

After calibration this algorithmic strategy is applied to the Al Wakrah stadium, leading to four structures made from reusable elements. These structures will be build for a housing and shopping function in the South African townships. The four structures together use 100% of the available set, while deviating around 1% from the required floorspace and 3% from the building mass shape. The created frames are however not directly translatable to practice, to this end the building services, vertical transport, stability elements and non-structural elements need to be defined first.

Contents

1 | Introduction

"The key to growth is the introduction of higher dimensions of consciousness into our awareness."

- Lao Tzu -

1.1 Motivation

Stadium Reuse

Hosting sporting events like the FIFA World Cup and the Olympic Games is a prestigious matter for countries. These events have budget estimates and societal impact on a scale that require national politics to get involved in the decision making. During this decision process the total economic and societal impact is estimated and based on the estimated outcome, the country either chooses to bring out a bid or refrains from hosting the event.
However this estimation is filled with uncertainties, leaving decision makers with the difficulty of making a choice without knowing whether the country will benefit or not. One of the main uncertainties leading to conflicting outcomes of these estimates, is the utilisation of venues after the event [Blake, 2005].
The stadium realisation costs are included in the event bid, however they cannot be covered by the event exploitation. These venue costs have to be earned back over a longer period of time. Meaning utilisation of the event venues during their total lifetime influences the economic impact of the bid on society [Klomp, 2013].

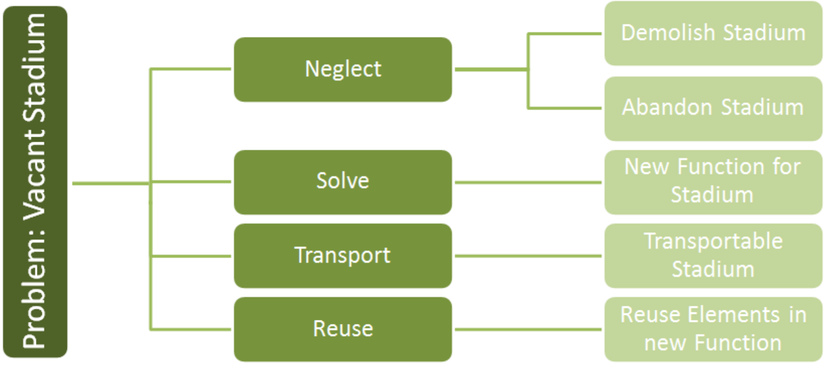


Figure 1.1: Possible outcomes for sport stadiums after the event [Loosjes, 2011].

Loosjes proposes that every stadium built for an event has the same status after the closing ceremony: it is vacant, creating zero revenues to cover the investment and meanwhile maintenance costs keep stacking up [Loosjes, 2011]. From this status, 4 possible directions are given (Figure 1.1) [1]:

- Neglect: By either abandoning or demolishing the stadium.
- Solve: The stadium gets a new function, creating a new revenue stream.
- Transport: The stadium is designed and realised as a transportable structure. After the event the stadium can be moved to a new location to generate revenues.
- Reuse: (Partly) deconstruct the stadium and reuse the elements that are extracted.

---

[1] The orignal model named the strategy 'Transport' as 'Prevent' and 'Reuse' as 'Avoid', for clarity purposes these names have been altered.

From these 4 options 'Neglect' is the least favorable one, since no revenues will be created and as stated by Klomp, these revenues are needed to cover the total project costs [Klomp, 2013]. However the 'Neglect' scenario has occured in practice as is shown by Figure 1.2, leaving the owner with a project that has a negative return on investment.

Figure 1.2: Three examples of abandoned venues. From left to right: Turner Field Baseball Stadium for the Atlanta Olympics 1996, Olympic Aquatic Centre for the Athens Olympics 2004 & Shunyi Canoeing Park for Beijing Olymics 2008

The 'Solve' option requires the lowest investments and energy expenditure after the event while ensuring an income stream for the stadium. However competition between bidding countries and expectations due to previous event editions, cause nations to offer hosting a high quality event with top facilities and stadiums, possibly leading to the realisation of venues which offer a greater capacity and quality level than needed for the lifetime after the event, meaning 'Solve' is not always an option [den Hollander, 2010].
An example illustrating the situation where a project had a to high capacity and 'Solve' was not a real option, is the Montreal Olympic Stadium. After the Olympics the venue was assigned to host baseball matches. However, the atmosphere in the stadium was terrible due to the overcapacity (Figure 1.3). Which lead to reducing ticket sales while having relatively large maintenance costs, causing financial problems for the stadium. In this case partly deconstructing the stadium could possibly have prevented this from happening.

Figure 1.3: Montreal stadium during the Games on the left and during a baseball match on the right.

When no new user for the venue can be found, the 'Transport' and 'Reuse' options can be applied. 'Transport' means creating a transportable venue that can be used at multiple events on different locations. Studies in this direction focused on creating a stadium suited for container transport. A concept either based on a given architectural design [den Hollander, 2010] [Loosjes, 2011], or starting from scratch [Klomp, 2013]. Other research in this field led to the creation of a floating stadium, consisting of large components requiring less (de-)construction efforts compared to a stadium suited for container transport [Fransen & Vermeulen, 2012]. What these studies have in common is that the stadium after deconstruction is rebuild to fulfill the same function. A concept applied by Ballast Nedam in their Plug & Play strategy [Ballast Nedam, 2013]. These studies assume that there is enough demand to reuse the stadium at future events to make the concept economically feasible. However if this is it not the case, the stadium remains vacant and could possibly shift to the 'Neglect' scenario. Due to this concept risk another direction is investigated in this thesis, aiming to increase the stadium reuse potential, so the shift to 'Neglect' is less likely to occur. This other direction is the 'Reuse' strategy, which means (partly) deconstructing the stadium to fit new user requirements. The 'Reuse' strategy has the advantage that the elements can be reused in a more flexible manner. The reuse structure can have a different structural configuration than the original building. This means the reuseable elements or components appeal to a bigger market, thus increasing their reuse potential [Crowther, 1999].

Therefore the 'Reuse' strategy is explored in this thesis, aiming to increase the stadium reuse potential and by doing so, averting the 'Transport' concept risk. However reusing the elements for a new structure comes with its own set of challenges. These challenges are identified by looking into reuse theories in the next section: 'Design for Reuse'.

Design for Reuse

Before the reuse challenges are described, first the incentives are researched in a broader setting than the stadium industry. Leading to a more comprehensive view on reuse and its possible challenges.

Reuse Incentives

A building is not a static entity, it is constantly subjected to change and a structure should be able to cope with these demand changes instead of losing its functional purpose. The fact that building materials have different life cycles and that the durability of most materials is longer than the durability of their functions, forms the incentive for transformation. Therefore, the specification and arrangement of materials accounts for the structure's transformation capacity and for the recycling possibilities of the materials, which is an important aspect for Industrial Flexible Design (IFD) [Durmisevic, 2006].

Another theory called 'Legolisering' elaborates on this component life-cycle focus. It states that the building industry should strive to reach a state of industrial tailor-made design. Which means the building sector changes from creating single man-made unique products for end users, into a sector that makes unique buildings through the assembly of standard industrialised building components. This way the individual needs will be suited better and the economical lifetime of the building increases, making the building more durable. The emphasis is no longer on the life cycle of buildings but on the life cycle of the building components [de Ridder, 2011].

In 1997 Damen consultants conducted a market research on the prospects of the IFD concept. This research showed that IFD in the Netherlands is feasible with three conclusions from the viewpoint of different stakeholders why it was feasible [Zeiler, 2009].

1. The consumers, asking for more flexibility due to changing demands.
2. The contractor industry, able to realize a more efficient production process if the marketshare of IFD buildings increases.
3. Society as a whole, which has a need for sustainable development, which is granted by demountable construction as shown in the previous paragraph

These reuse incentives will be translated into a strategy. For this strategy to be successful the reuse challenges are addressed in the next section.

Reuse Challenges

The non existence of a market for reuse elements is the foremost reason why the IFD building strategy is never applied on a large scale. This market does not exist because the building components cannot be stored and wait until a reuse function is at hand. This is caused by the loss of investment due to the time value of money and the costs for deconstruction plus storage [Roders, 2003].
Another challenge is defining the ideal (de)mountable module. The top boundary of the size is given by the means of transportation and the lower boundary by the increasing dis-assembly costs per action [Zeiler, 2009].
A third reason was lack of digital power to link the virtual modules to real world construction projects [Lichtenberg, 2005].
The last identified challenge is stated by Durmisevic, claiming that a building's reuse capacity depends on the structure's flexibility and potential to react to the demanded changes. A problem with this statement is that on the building site, every project suffers from unexpected problems. These problems are normally solved by improvisation, often leading to monolithic solutions which reduce the building flexibility. To prevent this, preventive measures need to be taken, ensuring that the IFD principles are also maintained during construction [Durmisevic, 2006].
What all the reuse theories discussed in this paragraph have in common is that they only specify the incentives and challenges of reusing elements. Some also present abstract strategies. However, practical methods to design for reuse are not mentioned, reducing the practical usefulness of these theories (Figure 1.4).

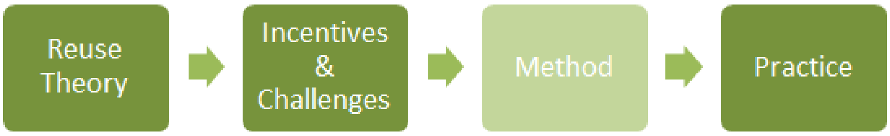


Figure 1.4: The available reuse theories do present the incentives and challenges for reusing elements, but a method to implement them is missing.

1.2 Previous Stadium Reuse Concepts

In the stadium industry the 'Reuse' direction has not yet been implemented successfully, it was attempted on the London 2012 Olympic Stadium but failed. Outside the stadium industry, the reuse of structural elements from an office building in a housing project has been studied. Both these reuse projects will be analysed in this section, looking into the methods applied for reusing the elements.

London 2012 Olympic Stadium

A stadium 'Reuse' was implemented during the London 2012 Olympic Games. For these Olympics sustainability was a key selling point, therefore the Olympic Stadium consisted of a permanent concrete lower ring and a temporary steel top ring (Figure 1.5). The temporary steel structure was to be removed after the games, reducing the stadium capacity to a size matching the demand.

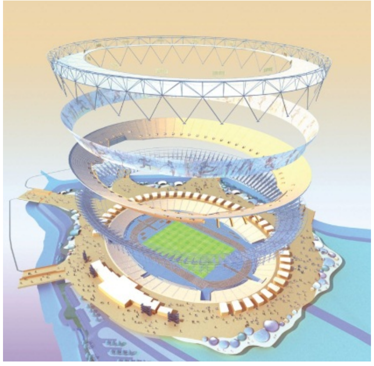


Figure 1.5: London Olympic Stadium

The steel ring is built with respect to the design for deconstruction aspects. So Dumisevic's (2006) requirement for reusing the structural elements is met. Also all the connections were made in a demountable way so the ring could efficiently be deconstructed in its components. But when looking at the other identified IFD challenge in section 1.1.2, no solution strategy is worked out to overcome the challenge of the non existing reuse market.
Because a destination for the reusable elements was not available, no party was interested in deconstructing the stadium and take responsibility; why invest when there is no clear option to create a return on the investment. Due to this lack of feasible reuse options, it was decided to let the steel ring remain and give the stadium a new function: Host the West Ham United soccer matches.
When looking at Figure 1.6, this meant switching from the 'Reuse' strategy to the 'Solve' strategy. In this case the switch seems a feasible solution, however this would be a problem if such a solution is not at hand and the 'Solve' strategy is not an option.

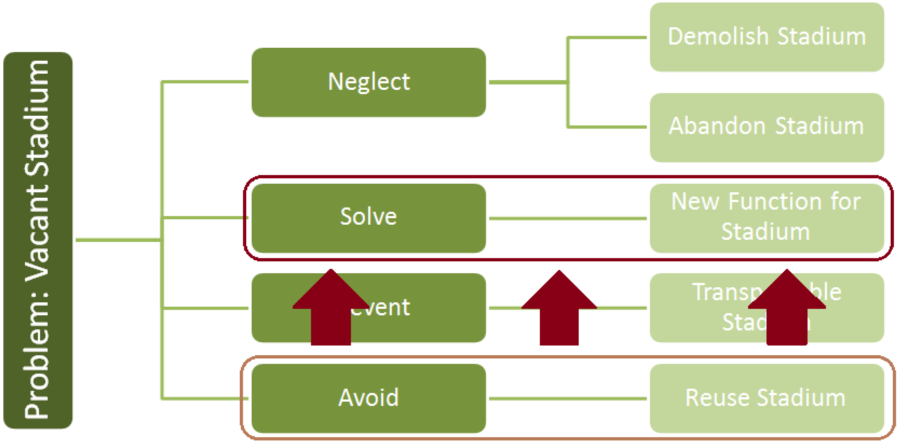


Figure 1.6: The Solution is changed from 'Avoid' to 'Solve'.

Related Reuse Research

The London case illustrates a challenge for designing with reused elements. This challenge is the thesis topic of Glias: 'The Donor Skelet'. The study focused on deconstructing an empty office building and reusing its concrete structural elements into a housing project, doing so in an attempt to solve the problem of the office vacancy rate in the Netherlands (14,6% at the end of 2012). A strategy is implemented where reusable elements in an existing building are identified by hand and based on these reusable elements, a new structure is designed. The new structure ideally build up completely from the reused elements.
Applying this strategy meant no market of reusable elements is required, since elements are reused directly in a new project. With this approach, roughly 90% of the elements can be reused on element level. The 10% which is not reusable, are mostly customised elements (e.g. facade beams), which due to their custom shape and function, are not flexible enough to fit in another structure.

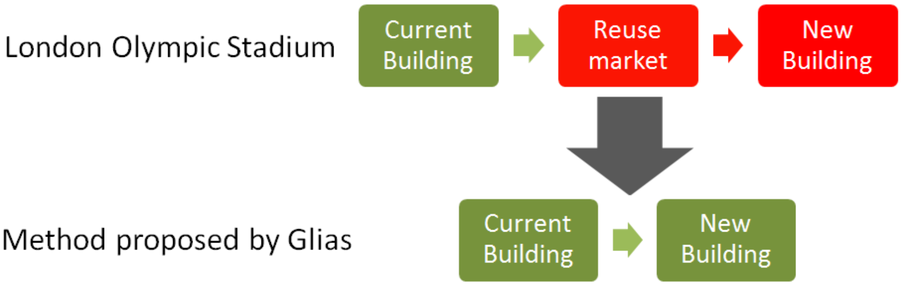


Figure 1.7: Skipping the need for the reuse market.

The strategy of Glias, reusing the elements directly into a new building, solves the challenge of the non existing reuse market 1.7. However an average of 10% of the building elements cannot be reused, due to their specialised functional shape, for example facade beams. Besides the 10% non reusable elements, the elements that can be reused lose an average of 25% of their volume due to modification requirements. These modification requirements consist of two components. The first loss is due to the connections not being suited for deconstruction. Therefore the connection needs to be destroyed, leading to a loss of 20cm at each connection. The second component is a loss in element length caused by resizing the element to fit into the new design. To minimise this, an intensive cooperation is needed between architect and structural engineer, creating a design suited for the available elements [Glias 2013].

1.3 Problem Definition

The 'Reuse' Strategy

As previously stated in Section 1.1.1, the 'Transport' strategy for stadium reuse, has a potential demand problem. The 'Reuse' strategy aims to increase the potential demand by reusing the elements in a more flexible manner, increasing the potential demand so the demand problem of the transportable stadium is less likely to occur.
For the 'Reuse' strategy multiple books have been written describing the incentives and challenges for applying reuse, however clear methods on how to implement this knowledge into practice is not given (Figure 1.4).

Reuse Methods

The London 2012 Olympic Stadium case (Section 1.2), illustrates the reuse challenge following from the non existing reuse market. A challenge addressed by Glias, who proposed to reuse the elements directly into a new structure (Figure 1.7).
In doing so the reuse challenge of the non existing market for reused elements is solved, this method will from now on be referred to as the 'Direct Reuse' method.

Direct Reuse Consequences

Working with a fixed set of elements reduces the design freedom, because the design needs to be shaped to fit the available set of elements. Since the element dimensions are fixed and the wish to create an economic structure, the structural configuration will play a more important role in the design procedure compared to designing with new elements. This requires implementation of the method in an early design phase where the shapes of the building still need to be determined (Figure 1.8). So in the preliminary design phase, when designing with reused elements, the input consists of the spatial requirements of the building from the list of requirements, a set of fixed structural elements and an architectural mass study.

Figure 1.9 [2] shows that during design with new elements, defining the structural dimensions and quantities provides design flexibility in the simulation phase due to the variety of element choices. However during design with reused elements, defining the dimensions and quantities moves from providing flexibility in the simulation phase to becoming an extra constraint in the analysis phase, leading to a more complex design procedure.

This increase in complexity requires intensive collaboration between architect and structural engineer when using the direct reuse method. When done by hand, this resulted in a loss of reused structural volume, because elements had to be sized to fit into the new design.

---

[2] This figure will be explained further in section 2.3

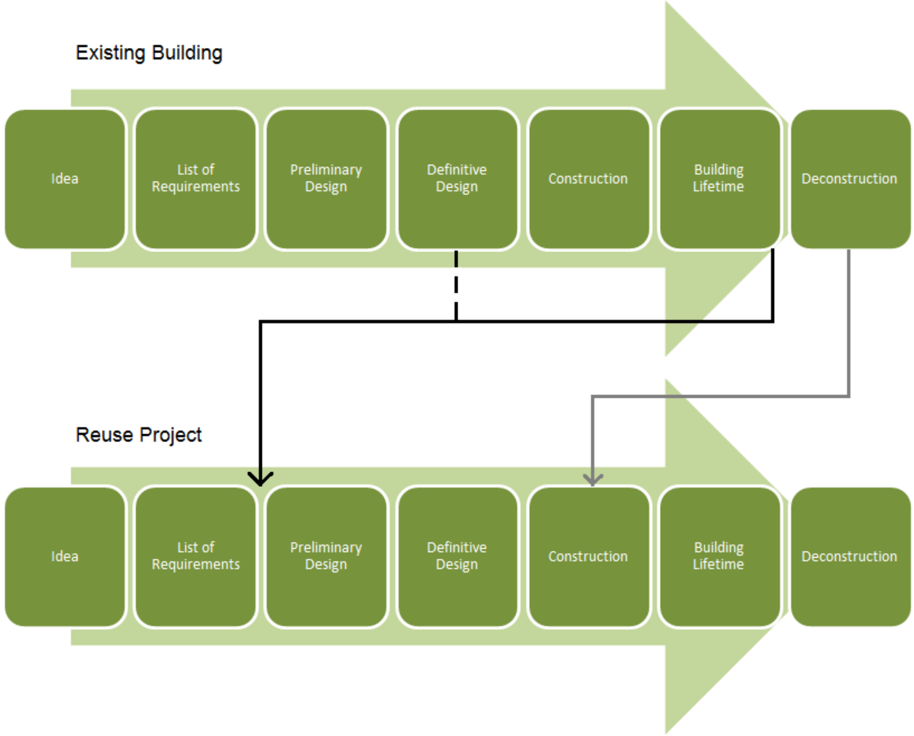


Figure 1.8: Due to the required flexibility the reusable elements need to be used as input for creating the preliminary design.

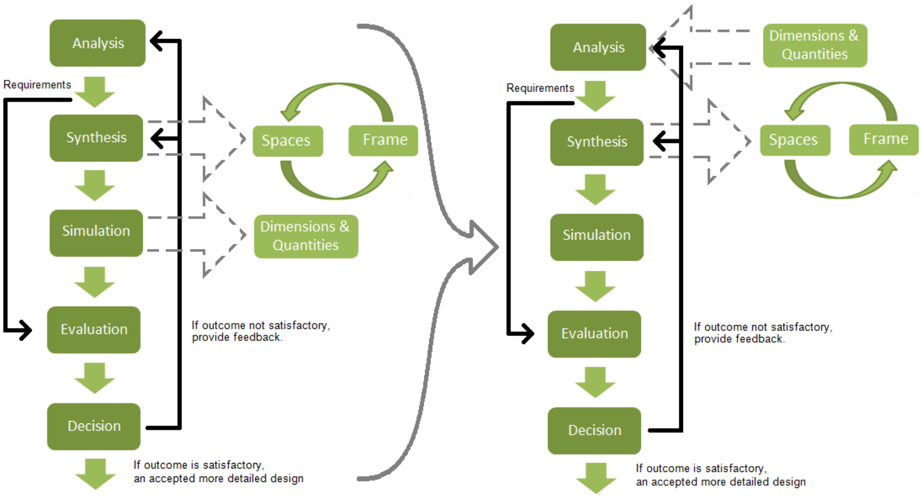


Figure 1.9: In design with reused elements, the design flexibility is reduced due to the shift of the dimensions and quantities in the design process.

Solving towards multiple design parameters, can be modeled as a large search space of solutions with each outcome having a certain solution fitness for solving the design problem. A solution space where each solution uses (part) of the reusable element set, utilises the structural capacity of these elements to a percentage of their maximum and meets the spacial requirements of the reuse design to a certain extend. Glias performed this search process using trial and error, a time intensive method that does not give information on whether the optimal reuse solution is found [Glias 2013].
This leads to the following problem statement:

A time efficient design method or application for simultaneously designing a structural frame with reused structural elements, while meeting the functional requirements of the spaces and respecting the architectural shape, does not exist.

1.4 Thesis Objectives

Reuse Concept Goal

The problem statement in the previous paragraph asks for a method or application that creates a structural frame. Furthermore the problem statement implies three different fitness parameters for this frame:

- Reusing as many elements from the original building as possible.
- Meeting the functional requirements of the spaces following from the list of requirements.
- Respecting the architectural shape.

These three parameters and their degrees of freedom create the solution space for designing with reusable elements. Creating a method or application that explores this solution space in search of the fittest solution, is a problem type suited for an algorithmic solution. The algorithm moves through the solution space searching for the fittest outcome. The converging speed of such an algorithm towards the fittest solution is potentially higher than searching the possible outcomes by hand. Therefore the method defined in the problem statement will be solved by the means of a computational application.[3]
The above points can be distilled into the following research goal:

Exploring the feasibility of a computational reuse strategy, linking the reusable elements directly to their new function, while taking into account the functional requirements of the structure and respecting the architectural shape through implementation of a genetic algorithm.

The reuse application will be used for a stadium project in Al Wakrah (city in the United Arabic Emirates), which will be partly deconstructed and its elements reused for the realisation of townships in Africa.
This thesis is the exploratory study towards reusing elements by linking them directly to their new function. Even though the concept is tested on a Stadium, the goal is to create a tool that can be used in a broader project spectrum than just the stadium industry. Because it is an exploratory study, showing the feasibility of the concept is the main focus.

Research Scope

This paragraph elaborates on the research fields explored to provide a scientific background for the reuse method. After this identification, the research scope is set, aiming to set clear boundaries for the research, while leaving enough room to create the depth needed to achieve non trivial results.

---

[3] Further elaboration on why an algorithm is a possible contribution is given in section 2.4

Research Topics

To create the optimisation tool satisfying the research goal, a literary study is performed to set the scientific background needed to build the application. The literary review is divided in five parts, each part forms a section in the next chapter, these sections are:

1. Reuse Incentives
   Looking into the potential of reuse from a cost and environmental perspective. This section provides information on why reuse could be feasible.
2. Element Reusability
   Going into what can be reused from a building project, by looking at the different types of elements and comparing these to the requirements of reusable elements.
3. Design Phase Implementation
   A section providing information on how to implement the strategy of reusing the elements form the viewpoint of the design process.
4. Computational Strategy Implementation
   The thesis goal ends by stating a genetic algorithm will be implemented. This section goes into why a genetic algorithm is suitable for reaching the thesis goal and how these algorithms work.
5. Practical Reuse Implementation
   The last section looks at the requirements of getting the defined theoretical method into practice.

Topics Outside Scope

In this paragraph the research boundaries are described. These boundaries are set in such a way that the research can produce non trivial results while remaining of appropriate size for a Master's thesis. The following boundaries are set for this research:

- Only spatial requirements for the total functional space are taken into account in the configuration of the structural frame. Not the requirements of the individual functions.
- Installation requirements will not be taken into account, only a recommendation on how the required holes and spacings could be realised are given.
- The literary study will show that the focus lies on structural element reuse, cladding and finishing will not be taken into account.
- Connections will be designed to show the feasibility of the concept. So feasibility calculations are done, however the translation to definitive design calculations are set out of scope.
- No structural calculations are implemented in the application, the feasibility of the frame will be proven after the application is done by means of hand calculations.

1.5 Methodology

This section presents the structure followed to reach the thesis goal. The thesis chapters are ordered in such a way that each chapter is based on the previous one and forms the foundation for the next chapter. This structuring of the different chapters, including the way they are connected, is presented in Figure 1.10.

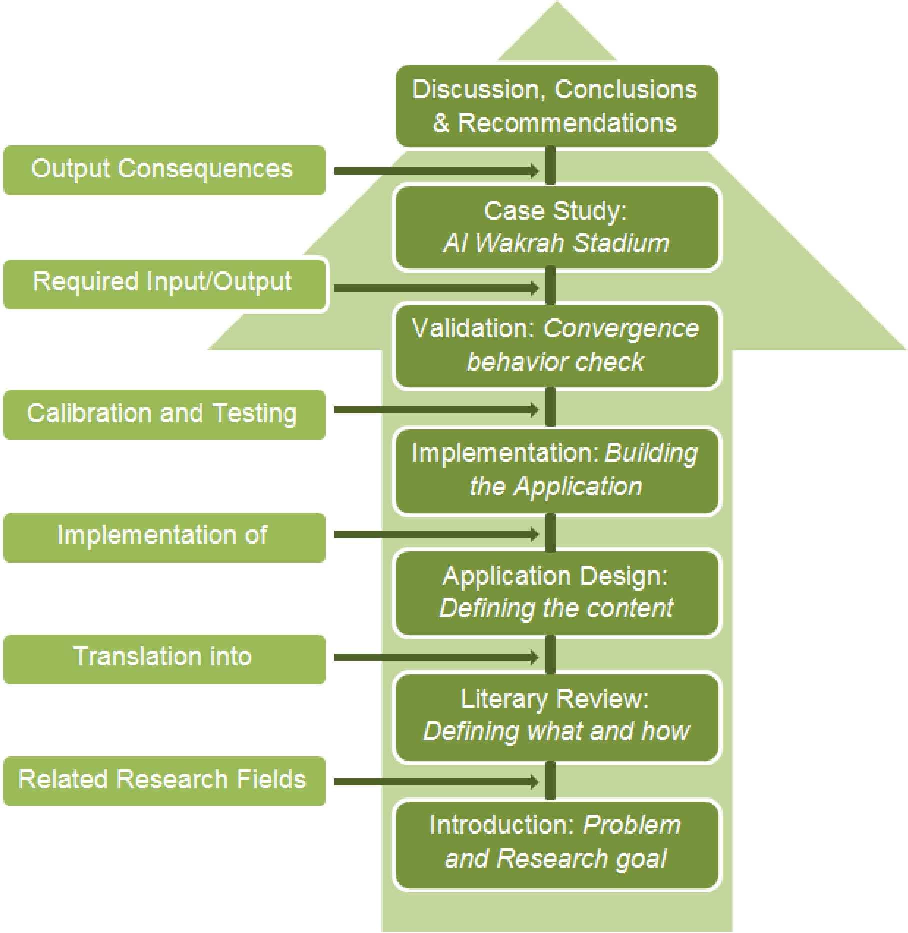


Figure 1.10: Reader Guide

Figure 1.10 starts at the bottom with the Introduction, this is the current chapter, presenting the research topic and explaining the thesis structure.
The second chapter is the literary study, which is connected to the introduction by the chosen research fields. The topics inside the defined scope are researched here. Figure 1.10 shows that in this chapter the topic is explored in a broad manner, giving a wide academic foundation for the rest of the thesis, from here on each chapter will zoom in on the set goals.

Chapter 3 gets into the tool design aspects, in this chapter the translation is made between the information from the literary study into a set-up for the application. Each aspect of the literary study shapes the application. The chapter closes with presenting the application, showing how the literary information is implemented into the application. This is done by diagrams showing the optimisation process but also by presenting the Grasshopper canvas, giving insight into the steps the application makes.
After the previous chapter is completed, the application implementation is completed, however the proper input for the application needs to be selected and imported, which is done in chapter 4 'Case Research'. When the in- and output of the application is collected, the program is executed. The chapter closes by showing the application results and by presenting a 3D-model to visualise the results.
Chapter 4, the application validation shows the correctness of the case study results as set-up for the last chapter. This last chapter holds the discussion and conclusion of the created application.

2 | Literary Review

"I want to move to theory. Everything works in theory."

- John Cash -

The literary review is split up in 5 sections, each contributing in sketching the literary background for the reuse application.
The first section provides an answer to the question: "Why reuse?", what makes it a relevant field for investigation; what are the incentives.
The section 2 of the literary review investigates which factors make a building element suitable for reuse. The section covers topics on "What to Reuse?"
Section 3 looks into the design process, creating a background for how to implement the proposed reuse strategy. This section sketches the changes in the design process due to the introduction of reusable elements and concludes by setting the implementation boundaries for the reuse application.
Section 4 elaborates on computational methods for the design of structures. How can these methods contribute during design, which method is chosen and how does it work.
The fifth section discusses the requirements for the practical implementation, required when using this method in practice.
The chapter concludes with a summary of the findings from the previous section, giving an overview of the scientific background of the reuse method.

2.1 Reuse Incentives

European and national legislation, like the 'Landelijk Afvalbeheerplan' in the Netherlands, encourage society to move in a sustainable direction. The incentives behind this movement are sketched in this section. The first two subsections 'Lifespans' and 'Waste Hierarchy' go into the environmental aspects of reuse, while the third subsection goes into the economical incentives, leading to a overview of the answers to the question why reuse should be pursued.

Lifespans

Each type of building element has its own (technical and economical) lifespan. The reason for possible under-utilisation of these lifespans is described by the living building concept (LBC) [de Ridder, 2006]. The LBC argues that the world is changing faster than the build environment. This can lead to parts of a building losing functional value faster than that the technical lifetime is reached. Possibly leading to buildings becoming useless because they do not meet user requirements anymore. Clients should be aware of changing circumstances and insights asking for 'living buildings', factors causing these changes are:

- Users with wishes and requirements.
- The surroundings with associated stakeholders.
- The regulations for what should be established and what is allowed in buildings.
- Technology is in a constant acceleration.
- Climate changes with rainfall, winds and temperatures.
- Changing financial situations.

These changes should be regarded as regular instead of exception and therefor be incorporated in building plans to ensure the technical lifetime of the elements are maximally utilised [de Ridder, 2006].

This under-use of the technical lifetime does not only occur at building level, but also at component level. Some parts of the construction lose their functional value faster than other components. Possibly leading to the situation where the entire building becomes vacant while a significant part of the structure is still in usable.

For example, while the structure of the building may have the service life of up to 75 years, the cladding of the building may only last 20 years. Similarly, services may only be adequate for 15 years, and the interior can change as frequently as every three years (Figure 2.1).

Nevertheless, it is common practice for parts with a relative short durability to be fixed in permanently, preventing easy disassembly. Leading to demolition instead of deconstruction. However when the potential of disassembly is recognised, it is possible to divert the flow of materials from disposal and save

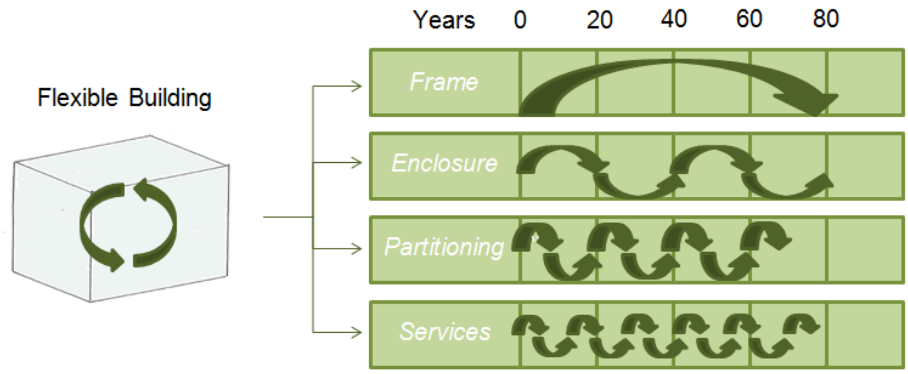


Figure 2.1: Different life cycles of element groups [Durmisevic, 2006].

the energy embodied in them by avoiding the demolition process [1]. This way the lifetime of the structural components can also be fully utilised, independent from the components with a lower life expectancy. This increased lifespan utilisation leads to a more sustainable use of materials [Lichtenberg, 2005].
Therefore an increase in sustainability can be created by reusing the building elements with a higher life expectancy than the structure as a whole.

Waste Hierarchy

The construction industry is responsible for 40% of the waste footprint in the Netherlands, thereby reducing the amount of construction and demolition (C&D) waste, the building industry could have a significant impact on the Netherlands total waste output [de Ridder, 2011].
C&D waste can be divided into multiple categories, where each category has a different amount of environmental impact. To indicate the amount of environmental impact, the waste categories are placed in a ladder with the least sustainable category at the bottom and the most sustainable at the top. There are multiple waste ladders that indicate the level of sustainability. The ladders differ at the bottom (Figure 2.2). Since the 'Delft Ladder' gives a more detailed division of the categories, this ladder will be discussed further [Hendriks & Janssen, 2013].

The terms in the 'Delft Ladder' mean the following:

- Reduction: Through design with a construction configuration requiring minimal material volume.
- Reuse: Prolonging the life of a building or component by dismantling the building components at the end of their functional life cycle and reusing them in a new structure.
- Re-Cycling: Turning waste material into new products and by doing so, reducing the demand for new raw materials.

---

[1] The relation of embodied energy to sustainability is explained in the following subsection 'Waste Hierarchy'

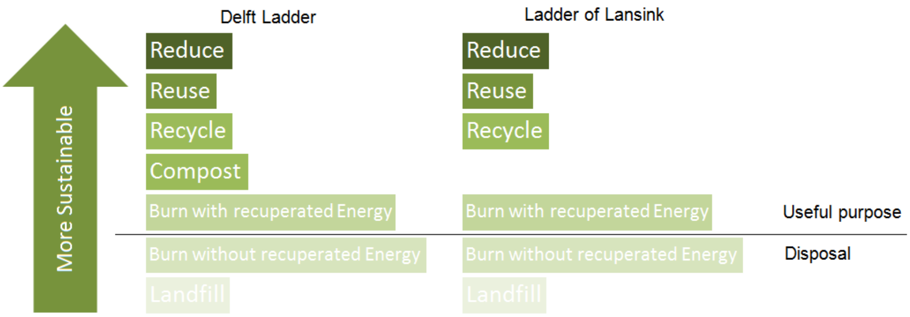


Figure 2.2: Examples of wasteladder, the 'Delft Ladder' will be discussed in this thesis.

- Compost: Breaking down vegetable waste by bacteria (germs), the end product can be used to fertilise plants.
- Burn with recuperated energy: Burning waste material as an source of energy.
- Burn without recuperated energy: Again burning the waste material but in this case energy has to be added to burn the waste material.
- Landfill: Non combustible materials are disposed of as landfill.

The top of the ladder represents the most sustainable C&D waste management strategy, meaning besides reduce, reuse requires the least amount of energy and has the smallest carbon footprint. The strategies lower on the ladder are less sustainable, therefore from the environmental viewpoint, strategies high up on the ladder should be pursued.
The top strategy, 'Reduce', is not applicable here, since the strategy from chapter 1 starts from a already realised building. Next on the ladder are 'Reuse' and 'Recycle', from which the favorable one in terms of sustainability is described by the principle embodied energy (Figure 2.3).

Figure 2.3 shows reuse strategies down to the building level 'Element', from there on the strategies are recycling strategies, showing that reusing elements requires less energy than recycling elements [Naber, 2012].
The strategy opted for in Chapter 1 is reusing on 'Component' or 'Element' level. The 'Building' and 'Component' strategies represent the 'Transport' solution from Chapter 1, with require less energy but are out of scope because they are a different concept type, with a risk in possible demand for the reusable systems or buildings.
The choice for reusing on the 'Component' or 'Element' level is made in Section 2.2.

The waste hierarchies form a theoretical background indicating a sustainable direction for the building industry. To get the sector moving in this sustainable direction, waste handling legislationâĂŹs have been developed on a national and European scale. The European UnionâĂŹs Waste Framework Directive (WFD) requires 70% of each member stateâĂŹs C&D waste to be reused or

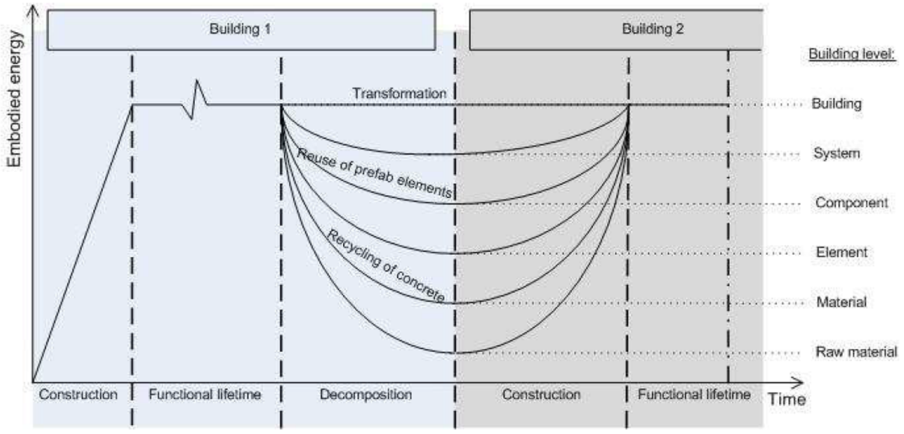
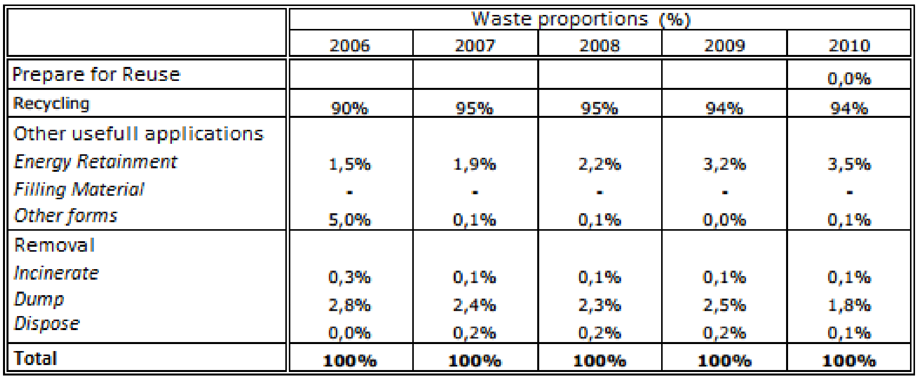


Figure 2.3: The Embodied energy in a structure, where each vertical movement of the line represents energy being used [Naber, 2012].

recycled by 2020. However, since each member state had to incorporate the requirements into their own legislation, the methods for achieving this target will vary across the continent. While Germany, Denmark, Ireland, the Netherlands and the United Kingdom have already surpassed the Waste Framework DirectiveâĂŹs C&D waste requirements (Figure 2.1), countries such as Spain, Poland and Greece have diversion rates which are currently below 20% [Jeffrey, 2011].



	Waste proportions (%)				
	2006	2007	2008	2009	2010
Prepare for Reuse					0,0%
Recycling	90%	95%	95%	94%	94%
Other usefull applications					
Energy Retainment	1,5%	1,9%	2,2%	3,2%	3,5%
Filling Material	-	-	-	-	-
Other forms	5,0%	0,1%	0,1%	0,0%	0,1%
Removal					
Incinerate	0,3%	0,1%	0,1%	0,1%	0,1%
Dump	2,8%	2,4%	2,3%	2,5%	1,8%
Dispose	0,0%	0,2%	0,2%	0,2%	0,1%
Total	**100%**	**100%**	**100%**	**100%**	**100%**

Table 2.1: C&D waste management percentages from 2006-2010 in the Netherlands [Rijkswaterstaat, 2013].

The sustainability goals in legislation refer to the recycling of C&D waste, the goals do not give an incentive towards reuse. However based on the 'Delft Ladder' and looking at embodied energy, reusing is a more sustainable solution than recycling. The difference in sustainability between the two is sketched in Figure 2.4. Comparing the recycle steps to reuse, it becomes clear that recycling requires more processing thus energy, which makes reuse environmentally more attractive. To put this difference in numbers, Glias has calculated that the environmental impact of reusing structural elements compared to elements made with 100% recycled concrete aggregate are reduced 75% [Glias 2013].

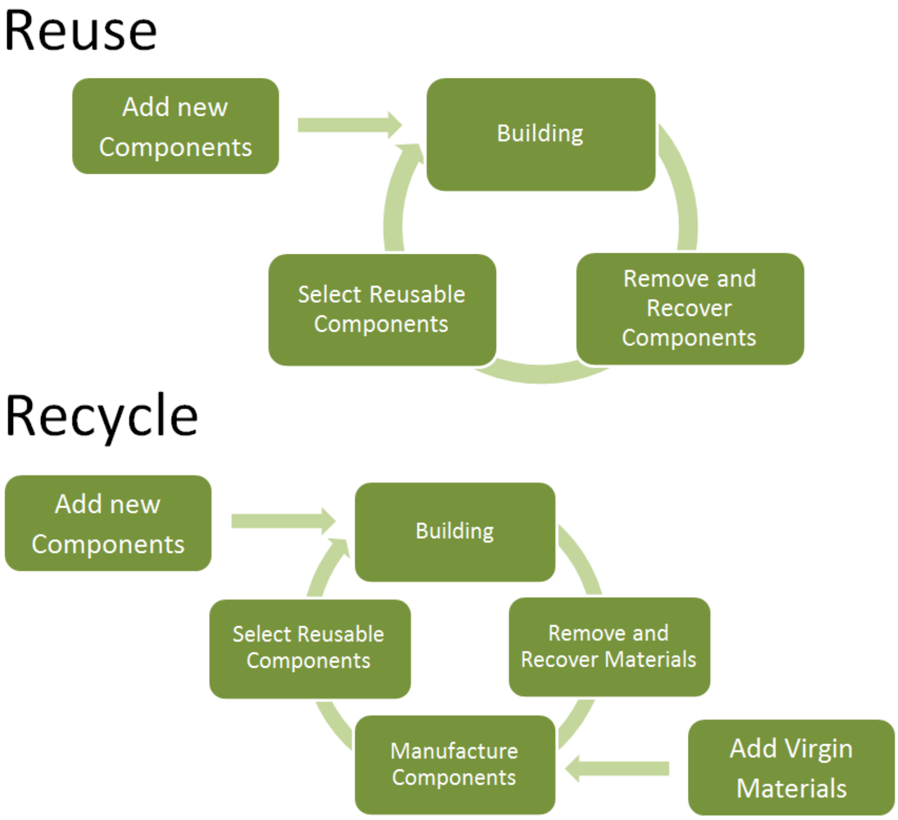


Figure 2.4: Difference between reuse and recycle [Durmisevic, 2006]

So while the waste management strategies for C&D waste focuse on recycling, a more sustainable design strategy would be to minimise the total mass of the material that will be recycled and to maximise the material that is reused. Now the incentives towards reuse from an environmental point of view are identified, the next subsection looks to the incentives from an economical viewpoint.

Economical Incentives

In 2005 the costs for demolishing, transporting and dumping the C&D waste of a building was 20 euro cents per kilogram, creating a new building costs 80 euro cent per kilogram. However even though the deconstruction and demolition costs equal 25% of the building costs ([Lichtenberg, 2005]), reusing elements is not common practice, partly due to acclaimed negative financial aspects.
One of these financial aspects is that creating a building designed for deconstruction (DfD) is more expensive than not doing so. These extra costs need to be made in the construction phase, an early phase of the buildings lifetime. Due to

the time value of money and the uncertainty whether the elements will be reused, investors are not likely to make these extra costs [te Dorsthorst et al., 2000].
To reduce the uncertainty of reuse, the strategy proposed in chapter 1, works from a situation where the reuse project is known beforehand, however this is not possible for every project. To reduce the reuse uncertainty, the government can implement legislation requiring reuse of buildings on 'Element' or 'Component' level (Figure 2.3).
The time value of money is a factor that cannot be reduced, however through legislation or subsidies for reusing elements, its influence towards not choosing DfD can be reduced.

When looking at the building elements, the difference in costs for DfD versus not applying DfD comes from two factors. The first is over-dimensioning of the elements. Either the elements are bigger, to later suit their reuse function or due to the higher load in the original function, the elements will not be loaded to maximum capacity during reuse [Huijbrechts, 2010].
The second factor is the difference in costs between the process of manufacturing new elements and the process of dis-assembling reusable elements. Where the factor of overdimensioning will always increase costs due to reuse, this second factor can also be favorably towards reuse as calculated by Glias. Glias concluded that a reused concrete element is 50% cheaper than a new one [2] because creating cement is 80% of primary energy expenditure, a procedure not required when reusing [Glias 2013].
During these calculations the donor building was not built for deconstruction and the elements had to be retracted by sawing them out of the frame. This relatively energy intensive retraction method could be avoided by applying less energy and labor intensive connections, increasing the financial gain of reusing elements.

This labor intensity is an important factor. Since the cost of materials are low in comparison with the cost of labor, it is âĂŸcheaperâĂŹ to waste materials rather than invest more time in using materials efficiently ([Hobbs, 2011]). Meaning the labor intensiveness of the connections and deconstruction process is vital to whether reuse is economically beneficial or not.
The paradigm here is that during construction, creating monolithic solutions is seen as a labor extensive procedure, leading to many monolithic connections, which later on make the deconstruction process labor intensive and thus financially unattractive.

Overview of Reuse incentives

The incentives for pursuing reuse are divided into environmental and economical incentives.
The environmental reuse incentives are that it leads to a higher element lifetime utilisation and that the embodied energy retained inside the elements is higher compared to recycling C&D waste.
Whether the economic incentives are positive or negative is less clear compared to the environmental ones. Theoretical studies claim reuse is economically ben-

---

[2] This percentage is based on a fictional test case.

eficial, however in practice this has not been proven yet. And due to the uncertainty on the return of investment, private parties are not yet applying DfD construction on a regular basis. What is certain is that the type of connections in a building and their labor intensiveness influence whether or not there will be economic benefits from reuse and by setting up reuse legislation or government funding, reuse can be made more attractive and reduce the uncertainty which is currently holding back the development of reuse projects in practice.

2.2 Element Re-usability

After previously looking into the incentives for reuse, the question "What can be reused?" will be the main topic for this section. To determine the suitability of an element for reuse, the first look will be into the Industrial Flexible Design (IFD) strategy. This is done due the need for flexibility identified in the subsection 'Lifespans'. Based on the IFD principles a further look will be given into the topics 'Element Levels' and 'DfD Element Requirements'.

IFD Strategy

The section on element lifespans identified that there occur differences in technical and economical lifetime between different components in the same building. Being able to update a building so it remains usable or to reuse the elements in a new structure, requires flexibility. One of the main strategies for flexibility in constructions is the Industrial Flexible Design (IFD) strategy.

IFD construction is an integrated approach for design and construction. It is a process where during the design phase the practical implementation of the design is thought through. While doing so also respecting possible future changes to the building. To achieve this, the building is created from demountable standardised industrial components. By using these components the realised structures can be easily adapted when the building demands change.
Because the building is now more flexible in following the whishes of the users, the lifespan of the building as a whole increases [Roders, 2003]. Alternatively by applying the IFD principles, the elements can also be disconnected from the frame and be reused in a different project.

The IFD strategy focuses on creating the components providing this flexibility. The main questions for these IFD components are which type of building components can be reused and what are the requirements for these elements [de Ridder, 2011]. These are the topics for the following two subsections.

Element Levels

Available Level Theories

There are multiple theories which define element levels or construction levels. These theories have in common that the identified element levels can be installed and replaced with minimal influence to the other levels, aiming to increase the inter-level flexibility. This will speed up the design process, because no complex adaptive system (for instance a building) will succeed in adapting in a reasonable amount of time unless the adaptation can occur subsystem by subsystem, each subsystem relatively independent of the others [Alexander, 1964].
The 'Open Bouwen' design methodology is one of these theories. It is primarily intended as an organised way of responding to the demands of diversity, adaptability and user involvement in the built environment. Making structures able to respond to the various needs of individual users.

’Open Bouwen’ approaches the built environment as a constantly changing product, much like the living building concept, with the changes resulting from decisions made at various levels. The levels are distinguished as: city structure, urban tissue, support, space and furniture. (Figure 2.5).
The ’Open Bouwen’ levels are set in such a way that the need for change at a lower level emerges faster than at upper levels. The method aims at a situation where decisions to be made at a lower level require no or only minor changes on a higher level [Zeiler, 2009].

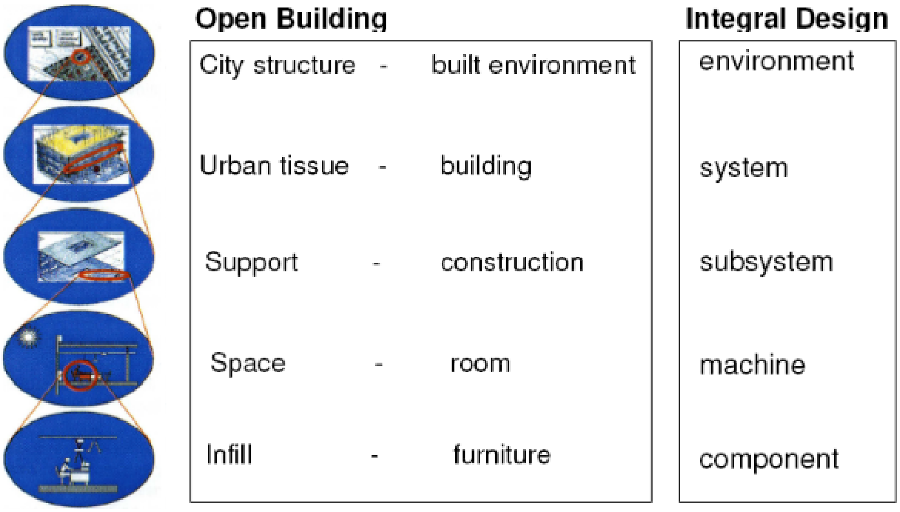


Figure 2.5: The different levels of the Open Bouwen method, compared to the levels of Integral Design.

The concept of ’Slim Bouwen’ agrees with the required divisions and presents a sequential building constructed from decoupled building elements, creating the opportunity for the lower level groups to be changed (Figure 2.6). A decoupling leading to high flexibiliy in later life stages of the building. The goal of this decoupled, sequential levels is to minimise the dependency of the process parts.

1. Casco (Structural Frame): The realisation of the structural frame, consisting of beams, columns, floors and load baring walls.
2. Gevel + Dak (Facade + Roof): The realisation of the roof and facade, forming the shell of the building.
3. Installaties (Installations): This are the installations for which in the previous phases, measures have been taken to fit these in.
4. Inbouw (Non Structural Elements): This is the realisation of the interior walls, deviding the spaces up in the desired configuration.
5. Oplevering (Delivering): Handover of the building to the user, after which he is able to import his desired furniture and start utilising the building.

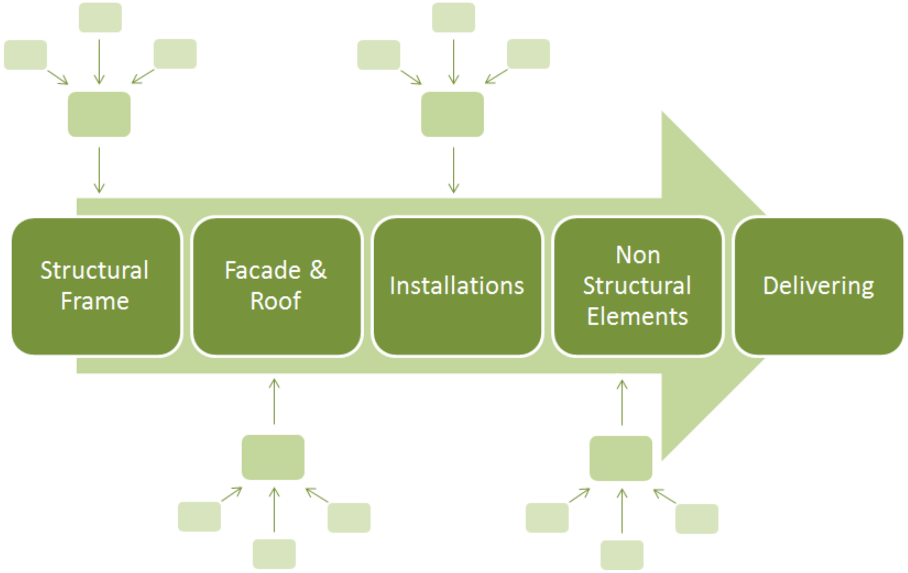


Figure 2.6: The sequential building process of the 'Slim Bouwen' Method, aiming to minimise the dependency of the building phases.

The theories of 'Slim Bouwen' and 'Open Bouwen', present respectively a hierarchic division in element levels and sequential installation process to create the required flexibility for implementing IFD in practice. Another flexibility factor is that the lifespans on each level are roughly the same (Figure 2.1). So reducing the inter-connectivity between the levels, provides the required accessibility for each type of change during the buildings lifetime.

Designer versus User Division

The 'Open Bouwen' methodology proposes a division between the support and the space level. The support level being the structural frame and the space level consisting of the interior walls, windows etc. defining the interior layout of the building. This is because the structural framework has a public function subjected to several building regulations.
On the other hand the interior building elements are products which can be totally decided upon by the user. In most cases people change their homes on interior level. Therefore interior building elements, in contrast to the structural framework, require a consumer aimed production and marketing process.
'Slim Bouwen' makes the same division between public and private levels. Spacial division and furniture should be determined by the client and are also the two groups most frequently dealing with changing requirements [Lichtenberg, 2005].

Reusable Element Levels

Based on the division between the structural framework and the lower element levels, the amount of expected change and technical element life expectancies, the decision is made to create a reuse strategy aimed on reusing structural elements.

While the theories of 'Open Bouwen' and 'Smart Bouwen' are written for creating a flexible building, the same principles are assumed to hold for reusing the elements in a new building.

This assumption is made based on the fact that in case of reuse, the required changes are originating at the level of 'City Structure', 'Urban Tissue' or 'Support' (Figure 2.5), invoking changes from that level down to the lowest level. However the requirements to the structural elements for the new project will probably not have changed that much as mentioned by 'Slim Bouwen', which means they can be reused.

The element levels below the structural frame are bound to user whishes and have lower life expectancies, the chance that the elements on these levels can be reused is allot smaller. Leading to the decision that these elements are not taken into account in the reuse strategy developed in this thesis.

The structural elements account for 90% of the structural mass, creating a big impact on the environmental footprint of the building [Naber, 2012]. Therefore developing a reuse method for these type of elements can have a significant environmental impact. Besides the environmental impact, the economical value during reuse will also be in these structural elements, due to the small changes in requirements on this element level and their long life expectancy. Combining this with the positive economical feasibility study from Glias, makes forming a reuse strategy around the structural elements an interesting business case.

DfD Element Requirements

Setting the reuse scope to structural elements, creates boundaries for the reuse model input. To further analyse the possible input boundaries, this subsection goes into the boundaries set by the influence of material and connection types, analyzing the possible types of reusable structural elements.

Material versus Connections

Looking at material types the most effective way for steel to support DfD is to increase the reuse potential of other materials by applying steel as intermediary component in connections [Edmons & Gorgolewski, 2000]. This statement suggests the reusability is not so much determined by the type of material but by the connection types available for this material type.

This statement is supported by Webster & Costello in their research to reusing concrete. Stating concrete is perhaps the most challenging material to disassemble for future reuse, because concrete elements are heavy and difficult to move. Also cast-in-place elements are not connected with joints, which can easily be disconnected and dismantled. Therefor precast concrete has greater potential for reuse because these elements are often connected by mechanical fasteners, increasing the dismantling options. However precast elements are often combined with cast-in-place toppings, making it harder to disassemble elements without damaging them [Webster & Costello, 2005].

For that reason using cast-in-place concrete should be avoided when applying DfD.

This statement is countered by Naber, stating that cast-in-place toppings crack

off when moving the elements during deconstruction, meaning an increase of feasibility for reusing these types of floors [Naber, 2012].
Glias calculated financial feasibility, the research studied reuse of a concrete structural frame without DfD principles. The calculations showed reuse in this case was still feasible, but that the monolithic character did cause a loss in reusable structural mass due to the need of destroying the area surounding the connections (20cm per connection) [Glias 2013].
The reusability of structural elements is therefor assumed to mainly depend on the connection type and less on the material type. Secondly 'dry' connections are favored. However if the lifespan of a material type is rather low, the reusability of this material reduces.
For the traditional building materials of steel and concrete the lifespan is assumed longer than the lifespan of the total building [de Ridder, 2011].

Element Type

Next the influence of the type of structural element is investigated. 'Slim Bouwen' states that the construction world has been building the same way for years. There is lots of innovation on a component level, but not in structural configuration [Lichtenberg, 2005].
This statement combined with the element levels, where the division is made between structural frame and finishing, suggests that structural elements placed for facades or roof support are to be excluded from the reuse model. Since these elements have a lower lifespan and are more subjected to change.

Reusable Elements

The consequence of this is that the elements that remain suitable for reuse are the columns, beams and floorplates. Which are used in roughly the same way for years and years as suggested by 'Slim Bouwen' and for the traditional building materials steel and concrete have a longer expected lifespan than the building as a whole. [3]

Overview of Reusable Element Types

Based on the division between the structural framework and the lower element levels, due to the amount of expected change and differences in life expectancies, the decision is made to create a reuse strategy aimed on reusing the structural elements.
Due to their generic character, the structural elements suitable for reuse are the columns, beams and floorplates.
These structural elements on average account for 90% of the structural mass, creating a big impact on the environmental footprint of the building [Naber, 2012]. Combined with the positive economical feasibility study from Glias, this indicates the relevance of forming a reuse strategy for structural elements.

---

[3] For other materials the lifespan has to be investigated, when the technical lifespan of these materials is higher than the lower level elements, this material is also suitable for reuse.

2.3 Design Phase Implementation

The previous section elaborated on ’What to Reuse?’, resulting in the reuse of the structural element set. The next question is ’How to Reuse?’. This question is studied in three sections, with this first section looking into the design process.
The main research topic for this part is: ’How to implement the fixed set of reusable elements in the design process, while respecting the application goals.’ To address this topic, the design process and its design freedom are analysed and the changes that occur in this process when a reusable set of elements is introduced.

Design Process

To investigate the implications of designing with reused elements first the project life-cycle for designing without reuse is dicussed, this cycle is presented in Figure 2.7 [Hertogh & Soons, 2012]. After the initiative for the project (based on a need or a wish), the project requirements are assembled in a list of requirements (LoR). This list forms the basis towards which each design proposal is evaluated.

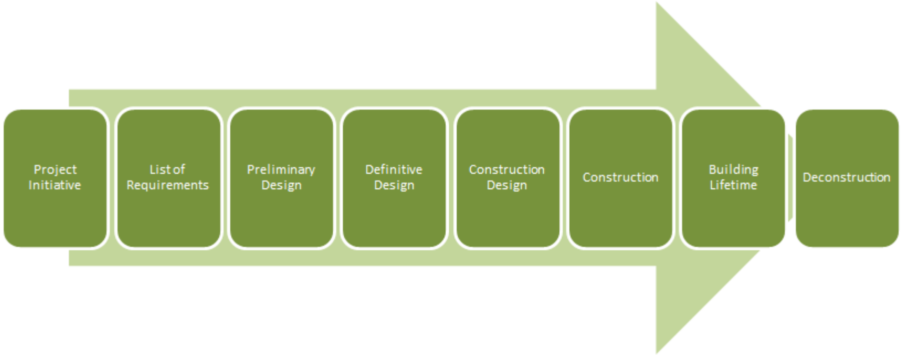


Figure 2.7: Total project life cycle, without reuse [Hertogh & Soons, 2012].

Design Process without Reuse

The design is based on the LoR and evolves in multiple steps, each step being more detailed than the previous with as final product the ’Construction Design’ containing all the information needed to construct the building. The process of detailing the design from the ’Preliminary Design’ all the way to the detail level of a ’Construction Design’ happens through an iterative cycle presented by Figure 2.8 [Hertogh & Soons, 2012]. Each design stage can take multiple runs through the loop, until the result for the current design stage is satisfactory to the designer.

Ideally these iteration cycles are done in cooperation between architect and engineer. The architect responsible for the representation of the esthetic aspects and the constructor responsible for the technical aspects [Terwel et al. 2009].

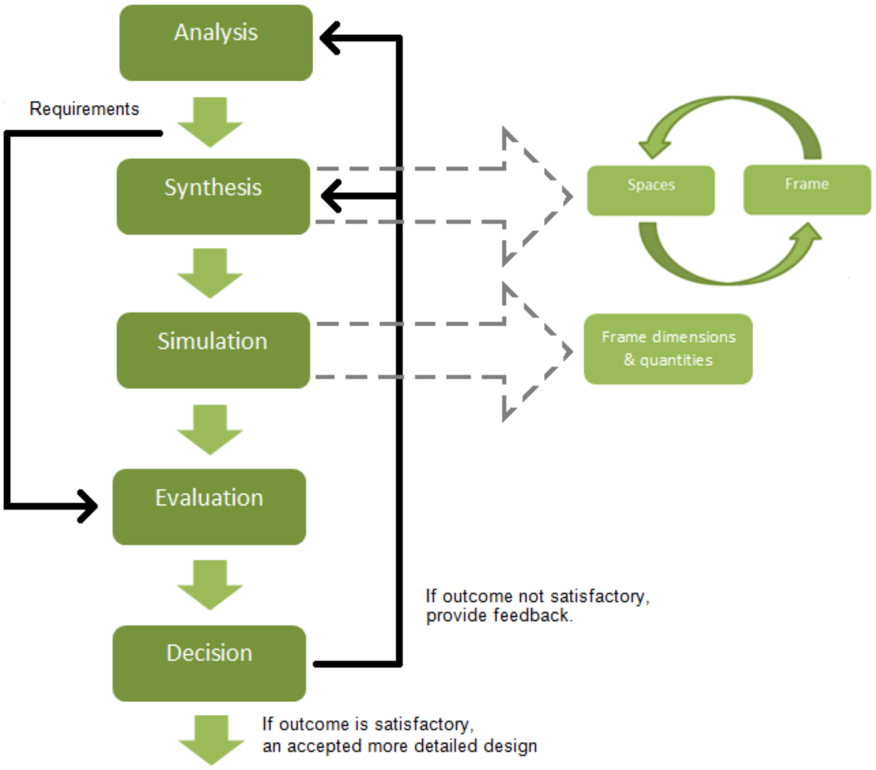


Figure 2.8: Design with new Elements iteration steps, the process to come to higher detail level in the design [Hertogh & Soons, 2012].

This cooperation and interaction takes places in multiple disciplines.[4]
For the structural elements, which will be reused, the interaction takes place between the 'Spaces' and the 'Frame'. 'Spaces' meaning the geometry of the total structure plus the rooms inside the building and 'Frame' being the structural elements forming the boundaries of the 'Spaces' (the boundaries in this scope only consisting of the structural element types: beams, columns and floor plates). Both disciplines influence each other as indicated by the circular arrows in Figure 2.8. The amount of influence and which of the two disciplines is governing to the other, differs per project [Kleinsman, 2006].
After synthesis the next step in the design iteration is simulation (Figure 2.8). In this step the dimensions for the elements in the structural frame are calculated based on the building codes. Due to the amount of element options, the dimensioning results in an element which is loaded near its structural capacity, leading to an economical element choice.

---

[4] Disciplines like building services and facades are not taken into account, since these should be implemented separately from the structural elements according to 'Slimbouwen' and 'Open Bouwen' [Lichtenberg, 2005][Zeiler, 2009].

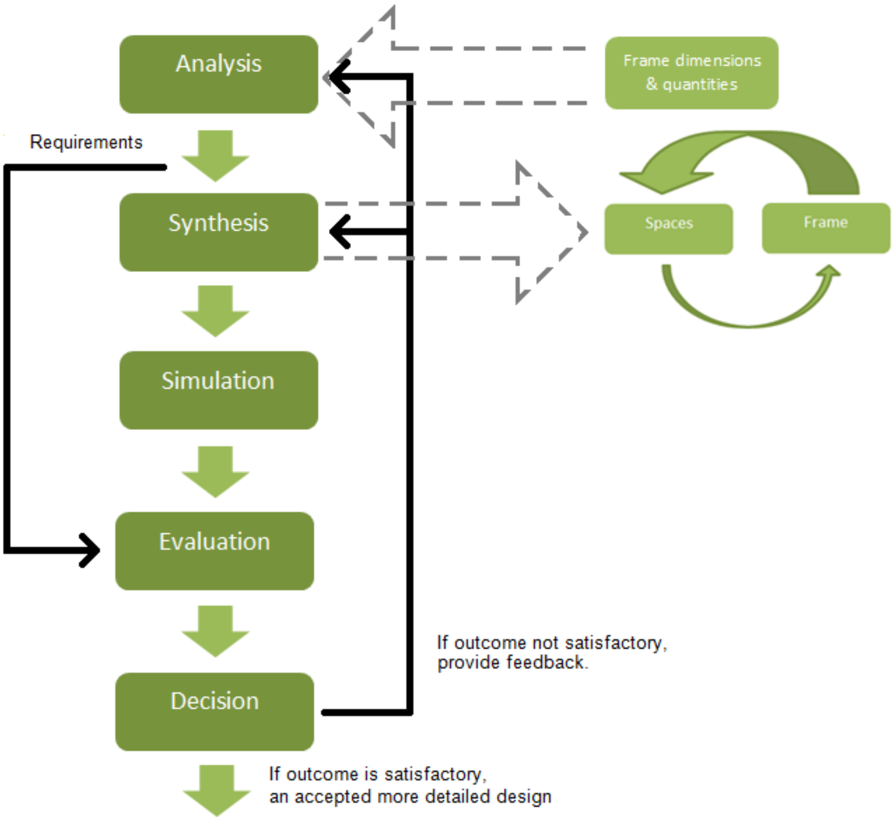


Figure 2.9: Design with Reusable Elements iteration steps, the process to come to higher detail level in the design.

Design Process with Reuse

When designing with reusable elements, changes occur in the iteration cycle. When applying the direct reuse method (section 1.3), the element dimensioning moves from providing flexibility in the simulation phase to becoming an extra constraint in the analysis (Figure 2.9). Leading to a reduction in design flexibility which occurs early in the project cycle, namely during assmbly of the LoR (Figure 2.7).
Besides this input and flexibility change, the process in the simulation phase changes. During the first design cycle, the 'Frame' knowledge is already in a very detailed stage, because the structural elements are already defined on the 'Construction Design' level. However the information on the 'Spaces' is still on a very conceptual level ('Preliminary Design' stage, Figure 2.7). This difference in detailing level forces the influence of the 'Frame' on the 'Spaces' to be relatively large compared to the influence in the other direction as indicated by the arrow size in Figure 2.9.

Information versus Flexibility

The difference in detailing level is sketched based on the MacLeamy curve (Figure 2.10) [Coenders, 2011]. Each design cycle (Figure 2.8) is a step to the right in Figure 2.10, gaining more information on the design and making choices based on this information, bringing the design closer to the final form and reducing the design flexibility.

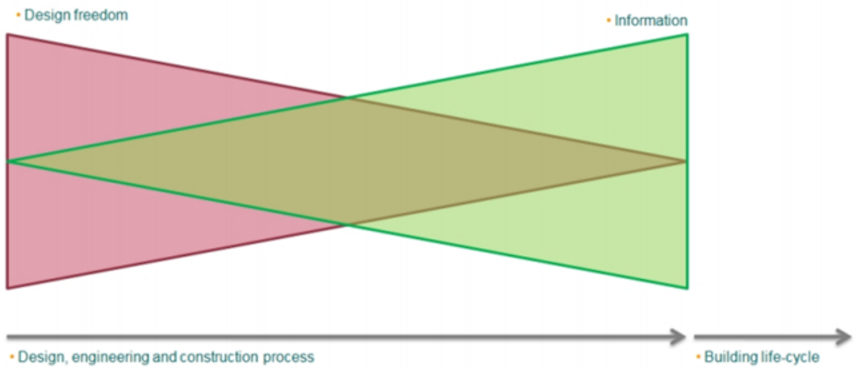


Figure 2.10: Development of the design flexibility and information during the design process. (Image Courtesy by Coenders)

In a design process with new elements, each design aspect is on roughly the same detailing level. During the design process when one aspect gets changed it influences the other aspects, causing them to change/require more detailing as well.

However as mentioned before, when designing with reusable elements the structural frame is in a more detailed stage ('Constructive Design' stage, Figure 2.7) than the other design aspects ('Preliminary Design' stage, Figure 2.7). This leads to a difference in design freedom and information between these aspects as shown in Figure 2.11.

Design Difficulty

This difference in detailing level between the different design aspects, forces the other design aspects to be shaped towards the structural elements, due to the reduced flexibility on this aspect. This causes a strain on the design process, since the other design aspects now balance between matching the available structural elements and finding the optimal building shape. This increased design difficulty is one of the bottlenecks identified when implementing IFD buildings in practice [Roders, 2003].

Glias identified this while researching the reuse of concrete elements from an office building into a housing project. Stating an intensive collaboration between architect and engineer is required for this kind of design process. The difficulty of reusing the structural element set, while respecting the other building aspects, resulted in reducing the lengths of multiple elements and not reusing a

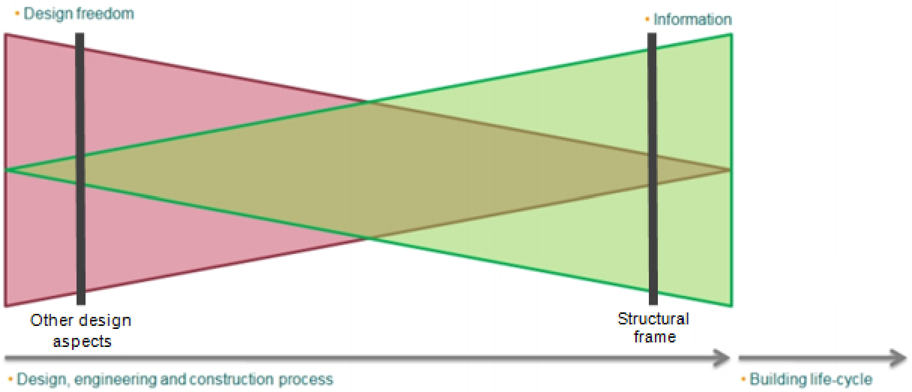


Figure 2.11: Difference in design stages between the structural frame and other design aspects.(Image Courtesy by Coenders)

number of elements at all [Glias 2013].
Both the last two measures are consolidations in terms of the reuse aspect, because ideally these elements are implemented without requiring extra actions like cutting the elements to the right dimensions. Also the elements in the second design being over dimensioned is economically and environmentally not ideal, again consolidations made due to the reduced flexibility of the structural element aspect.

The reduction in flexibility caused by the detailed structural elements, seems to influence the amount of effort required to come to a design suiting all design aspects. Coenders describes how computation in the building industry can influence the amount of flexibility and information during the design process [Coenders, 2011]. Since the design process is strained in this situation through a design flexibility issue, the possible contributions of computation in this field are researched next.

Software in the Design Process

One of the purposes of digital tools developed for the conceptual design stage is to aid the engineer in exploring the design space [van den Weerd et al. 2012]. This statement is further explained by Coenders, elaborating on the possible changes in design flexibility and information caused by digital tools as shown in Figure 2.12 [Coenders, 2011].

The top right figure in Figure 2.12 shows the possibilities provided by applying computational tools, the amount of design information is shifted towards the front of the design process, enabling the designer to make better informed decisions. And the applications also provide more flexibility in the later stages, again leading to better informed decisions (Figure 2.12 bottom left). The applications also provide design information during the building lifecycle (Figure 2.12 bottom right), informing the user about the current state of the building, so changes (for example in case of renovation or reuse projects) can be made based on accurate data.

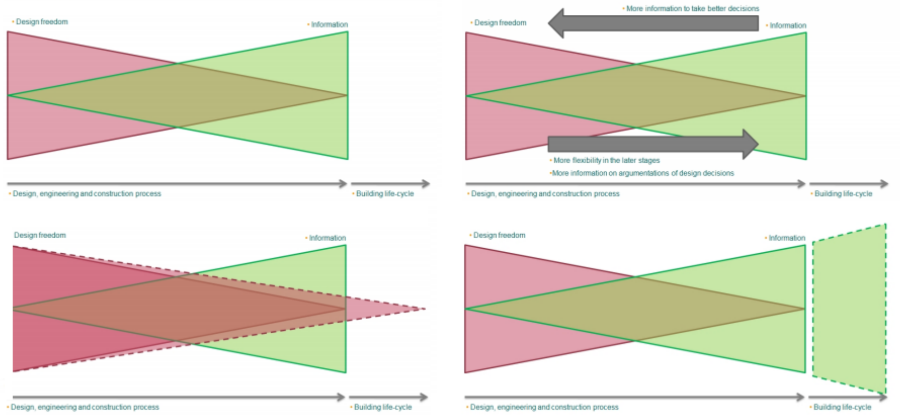


Figure 2.12: Possibilities in design flexibility and information due to use of computational applications in the design process.(Image Courtesy by Coenders)

Reuse Design Process

The complexity of designing with reusable elements is caused by the different detailing level of the design aspects during the design process. To this end, the shift in flexibility and information provided by computational design tools (Figure 2.12), could aid in reducing the identified complexity of different detailing levels.
The second part of the increased complexity is the wish to exactly reuse the available set of elements in a configuration matching the desired building shape. Both of the previously mentioned aspects increase the complexity of the design due to their required level of flexibility. Therefore, when looking at the design chain, the earlier the application is implemented in the design process, the better. Thus the application is implemented in the phase where the requirements from the LoR are translated into a preliminary design with a structural shape (Figure 2.13).

In Figure 2.13 the black arrows show the project phases where the application will be implemented. Meaning the available information at this stage is a list of requirements for the reuse project and a fixed set of elements from the existing building.
So far no architectural input is implemented, while the reuse project requires shape guidance to fit its new function. When looking at the first step in the design process of a building, the architect starts off with a building mass study. The application will be implementated after this mass study comes available, this study provides the guidance towards the demanded the building shape, while assuming enough freedom to create the structural frame.

The possible contribution of the computational application in reuse design is now identified, however the methods in the computational application are not yet defined. The available methods are researched in the next section.

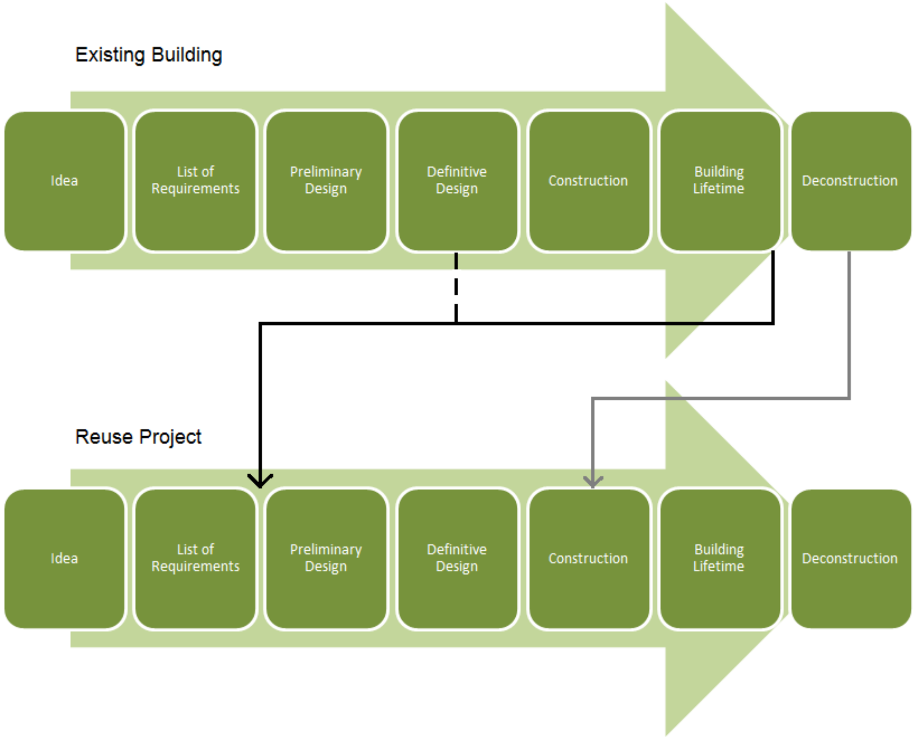


Figure 2.13: Ideal interaction moment between design 1 and design 2, based on required flexibility.

2.4 Computational Strategy

The previous section introduced computational design tools as an option for coping with the complications from designing with reusable elements. The first complicating issue being the difference in detail levels and the second component being the configuration problem, due to the wish to reuse all the elements available from the existing building.
A method to solve these kind of issues are algorithms. This paragraph elaborates on what this method type is, what its main components are and why this method is suited for the direct reuse strategy.

Algorithm Introduction

Algorithms can be seen as methods for solving well-specified computational problems. The problem statement specifies the desired input to output relation and the algorithm describes a specific computational procedure for achieving this relation. The problem statement can contain a wide variety of problems, with each problem requiring its own customised algorithm. The customised algorithm works when, for every input form the problem can have, it stops at the correct output [Cormen et al., 1990].
Due to the wide variety of problems, algorithms come in multiple forms. One of these forms is the 'Genetic Algorithm' (GA). GA's resemble the biological process of evolution. Evolution is a parallel search method, used to search among an enormous number of possibilities for the desired solutions. The enormous set of possibilities is the set of possible genetic sequences and the desired solutions are highly fit organisms well able to survive and reproduce in their environments [Mitchell, 1999].
With the parallelism of the search method is meant that, rather than testing one species at a time, evolution tests and changes millions of species in parallel, following the rules: evolution by means of random variation, followed by natural selection in which the fittest tend to survive and reproduce, thus propagating their genetic material to future generations. The fitness of an organism is typically defined as the probability that the organism will live to reproduce or as a function of the number of offspring the organism had.
This search method shows similarities with the design of structures, because a designer also strives to find the best fit solution among an enormous number of possibilities. Doing so by evolving the most feasible concepts into new ones and combining their strong points. The difference between the two is that the designer makes these decisions based on deduction and experience, while the algorithm does this based on random variation.
Due to the similarities between the GA method and designing, GA's are thought to be a suitable algorithm type for designing with reusable elements.

Genetic Algorithm Process

The species or organisms in the genetic algorithm can be thought of as chromosomes, in which each chromosome refers to a candidate solution of the problem. These chromosomes are divided into genes, a gene encodes for a particular element of the candidate solution called a trait, such as eye color. The different possible settings for a trait (e.g. blue, brown, etc.) are called alleles. Each gene is located at a particular locus (position) on the chromosome. (Figure 2.14)

Figure 2.14: Build up of a genetic algorithm chromosome.

Algorithm Search Space

All the possible combinations of the alleles from the genes together, form the search space of the algorithm. The algorithm itself is the method to move through this search space, looking for the solution with the highest fitness. The next candidate solutions to be tested will depend on the results of testing previous sequences, because most usefull algorithms assume that there will be some correlation between the quality of neighboring candidate solutions [Mitchell, 1999].

Algorithm Iteration Cycle

First a random population (group) of chromosomes is created. These chromosomes are tested by the fitness function, resulting in the fitness score of each chromosome in the population. (Figure 2.15)
In step 2 chromosomes from this population are selected, with high fitness chromosomes having a bigger chance to get picked compared to low fitness chromosomes. During this process the same chromosome can be chosen more than once. The choosing process continues untill a sufficiently large new population is created. The number of chromosomes to be sufficient has to be chosen by the algorithm programmer and is in practice often set to 4.
When the new population is created, this population will be modified in two steps. First the chromosomes will crossover and after that they will mutate. The crossover operator randomly chooses a locus and exchanges the sub-sequence before and after that locus, between two chromosomes, creating two new solutions, called offspring. For example the string 10000100 and 11111111 could be crossed over after the fourth locus, to produce the offspring 10011111 and 11100100. The crossover roughly mimics biological recombination between two single chromosome organisms.
After the crossover operator is done, the mutation operator randomly flips some of the bits in a chromosome. For example, the string 10001111 might be mutated in its second position to become 11001111. Mutation can occur at each bit position in a string with some probability.
When the crossover and mutation have occurred the new generation is ready to

be scored, after which an iteration cycle is completed.
These steps are repeated a set number of time or when a sufficient fitness degree is reached, this required fitness is set by the programmer[Mitchell, 1999].

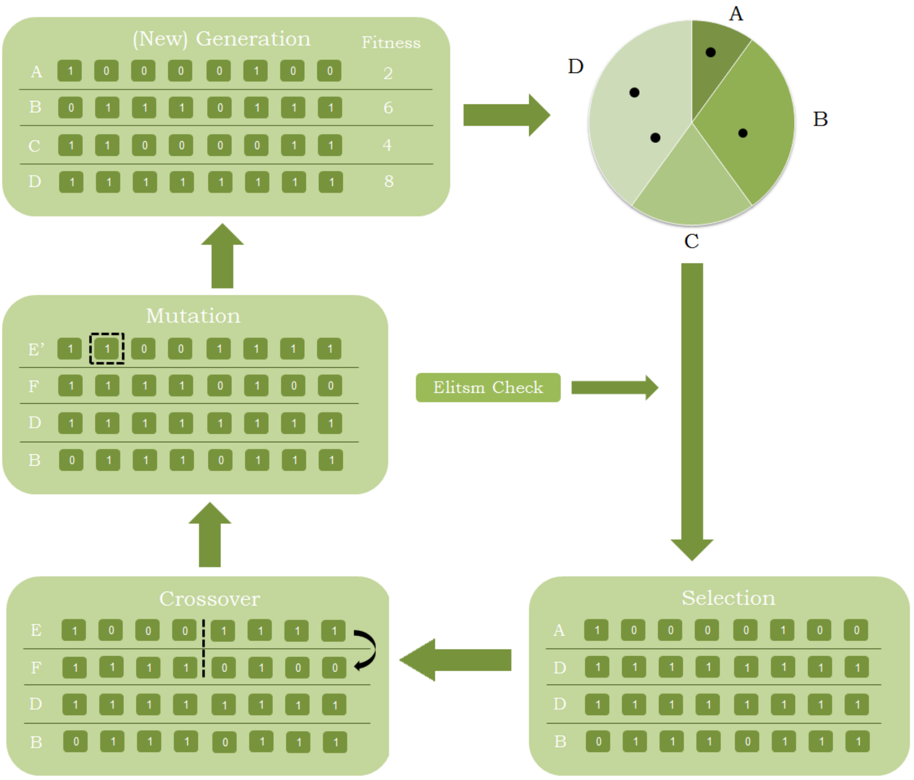


Figure 2.15: The cyclic process of a genetic algorithm

Elitsm

During the GA selection step, fit solutions have a high chance to reproduce to the next generation. However this reproduction is not guaranteed. To ensure that the fittest solution does not get lost, 'Elitism' was introduced. This method adds the fittest solution to the next generation in case the fittest solution was not picked. This procedure mainly increases the converging speed towards local optima, but also has a positive effect on the global converging speed of the GA [de Jong, 1975].

Algorithm Conclusion

Due to the similarities with the design process, a genetic algorithm will be implemented to search for the fittest solution. Different solutions for the problem will be combined by means of selection, crossover and mutation, while implementing elitism. These techniques are used to explore the solution space in order to find the optimal structural configuration for the reusable elements, while respecting the required amount of floorspace, available elements and given building mass.

After the frame configuration is found, the next step is to translate it into a 'Construction Design' (Figure 2.7). To this end, pratical knowlegde on how to create reusable connections is required, which is researched in the next section.

2.5 Practical Reuse Aspects

So far the literature study covered theoretical aspects, however one of the bottlenecks in IFD construction is the link towards practice [Roders, 2003]. This section covers this link to practice. Doing so by looking into the requirements for deconstruct-able elements, so that during the case study these aspects can be taken into account when making the design choices. The researched aspects in this section are: element sizes, connection requirements and transport. The material type will no longer be discussed, as it was concluded in section 2.3 that not the material, but the connection type is governing when looking at reusability.

DfD Element Sizes

Section 2.3 states that prefabricated structural elements from any material type can be reused, but what are the ideal sizes from the disassemblers point of view? Since disassembly is the opposite process of building, the construction should favorably be disassembled on the same level as which it was assembled. A higher level is harder to realize, because of transportation and a lower level is not favorable since this often means downgrading the elements [Naber, 2012].
For the dismantling party it is also easier to disassemble a building with several large elements rather than lots of smaller members, even though this requires heavier equipment. Large elements can easily resist small damages and with fewer elements the deconstruction time decreases.
Also, by reducing the amount of different components, the sorting process will be much easier. Besides, when large quantities of the same component become available after disassembling, they are more attractive for reuse compared to small quantities. And during mass production of these large quantities, the quality of the components can also be controlled better [Adams, 1989] [Hon et al., 1988].

Concluding for the element sizes, is that a set of large structural elements with high repetition is preferred. The upper limit of these sizes is caused by the transport aspect, discussed in section 2.5.3.

DfD Connection Requirements

When designing the connections, the demountability depends on the following guidelines:

- Apply reusable fixings [Crowther, 1999]
  When designing for deconstruction, reusable fixings are preferred. Bolted connections are easier to disassemble than welded connections, furthermore the bolts can be reused, while the weld material has less value after deconstruction. Also the risk on damaging the members is lower when using reusable connections (because the yield stress of the connection is often higher than the yield stress of the member). Therefore in DfD cast joints, glued fixation and elastic sealant should be used as least a possible.

- Easy and permanent access [Crowther, 1999]
  The connections should be easily accessible to ensure a fast deconstruction process. Also the elements should be able to act as independently as possible, because the more composite action there is, the more difficult it is to separate (reuse of the composite element as a whole can be the more economical choice).

- Standardized assembly techniques [CIRIA, 1983]
  Using well known assembly technologies ensures that no specialist labor and equipment is needed during the disassembly. This means that the process of dismounting the connections is faster/cheaper and has a lower chance of mistakes.

- Maximize repetition in the fixings [Crowther, 1999]
  Less connection types, meaning less actions/discussions are required during deconstruction. Thereby maximizing the repetition leads to a quicker and more economical deconstruction process. In the design phase this can be achieved by using the same type of connection, for example endplates for all beam to column connections or using one type of bolt in the connections.

- Design redundant connections [CIRIA, 1983]
  Applying redundant connections reduces component and material damage/deformation during (repeated) assembly and disassembly. Leading to a higher reuse percentage of the material and elements. Also tolerances should be provided ensuring the required movement during disassembly. These tolerances can be larger than those required for the initial manufacture or assembly process.

- Connections per material [Crowther, 1999]
  Steel construction with bolted connections provide the largest variety and flexibility of dismountable connections systems, which means it's most appropriate for disassembly. This also reflects in concrete structures where steel end fixings and bolted connections are preferred to grouted connections.

A side note to these guidelines is that they should not only be applied in the design, but also controlled during construction. One of the problems with DfD connections in practice is that buildings designed with a demountable system were not always assembled as such. Poor collaboration between precast manufacturer, engineer and building contractor was usually the cause. Leading to improvisation, where in some cases the connections were fixed with cement, straining the deconstruction process [te Dorsthorst et al., 2000].

Transport

The section on the waste hierarchy showed that decomposing a building on a higher building level, means less embodied energy gets lost, implying a higher level of sustainability. Naber argues in her thesis that the 'Component' level (Figure 2.3) is the highest possible level of reuse when the building parts are to be transported after disassembly [Naber, 2012].
However a construction project in the Netherlands has proven otherwise. For

the project 'De Bolder', the entire building was assembled in a factory, while the foundation was made on site. When the building was assembled, it was transported in one piece to the final location (Figure 2.16).

Figure 2.16: 'De Bolder' transported as a whole over to its final location.

This case proves that the maximum component size in which the building needs to be deconstructed, depends on the available transportation. However since this thesis aims to create and show the feasibility of a new reuse concept, the means of transportation chosen need to be widely available and generally accepted.
These widely available means of transport are investigated by Klomp, looking for the most economical way to transport the stadium elements from location A to location B. The conclusion of this study wrote: 'Although permit based transportation via trucks leads to larger possible elements, which lowers assembly costs, the employability of the stadium is limited when only special road transport is possible. Furthermore, transport by road is more expensive compared to transport via containers over water. [Hollandia, 2013]
Thus, road transport for special elements should be minimized and the stadium should be transported with sea containers. Hence, the stadium structural elements ought to be optimized regarding the sea container dimensions.' [Klomp, 2013]
For the disassembly this means that the transportable components are maximised to the dimensions of a 40ft sea container (Figure 2.17).

Klomp also stated that the maximum container weight was governing over the container dimensions. This has effect on the component shapes, since now concessions can be made on the ideal stacking shape of the elements, because the container will not be filled to the top [Klomp, 2013].

Figure 2.17: Container dimensions and maximum weight

Practical Implementation Conclusions

Based on the research topics in this section, the practical implementation for reuse depends on the components being deconstructed in the same order as they were assembled, preferring large components with a lot of repetition.

The connections depend on certain guidelines to be demountable . Besides following these guidelines, also communication between precast manufacturer, engineer and building contractor needs to be monitored, ensuring no permanent solution is applied due to on site improvisation.

Finally to ensure flexibility of the reusable elements, the elements should fit into 40ft containers. Since the element weight will most likely be governing over the volume of the elements, concessions can be made on the ideal element stacking shape.

3 | Application Design

"Design works if it's authentic, inspired, and has a clear point of view. It can't be a collection of input."

- Ron Johnson -

The literary review created a setup for the application, with each section narrowing down the research field. This chapter positions the reuse application, used to reach the thesis goal, inside these literary bounds.
Section 1 goes into where and how the application is implemented in the design process. With this decision set, Section 2 covers the application input based on the information available in the chosen design phase. Section 3 discusses the output, ensuring the products of the application are useful in the design process, before working out the process inside the application. This inside process is described in Section 4 and 5, with Section 4 defining the building blocks of the structural frame, called subsets and Section 5 describing the reuse algorithm, configuring these building blocks to user wishes. The final section is a presentation of the UML model of the application, combining the computational decisions defined in the previous paragraphs into one figure.

3.1 Design Chain Implementation

The literature review Section 2.3 suggests implementing the application in an early design phase to ensure enough flexibility. This section elaborates on how the application is implemented through discussing the options given by the design iteration cycle.

Preliminary Design Implementation

The implementation in the preliminary design phase can be done in multiple stages. Based on the design iteration cycle, the three stages before 'Decision' are discussed as possibilities for implementing the strategy (Figure 3.1).

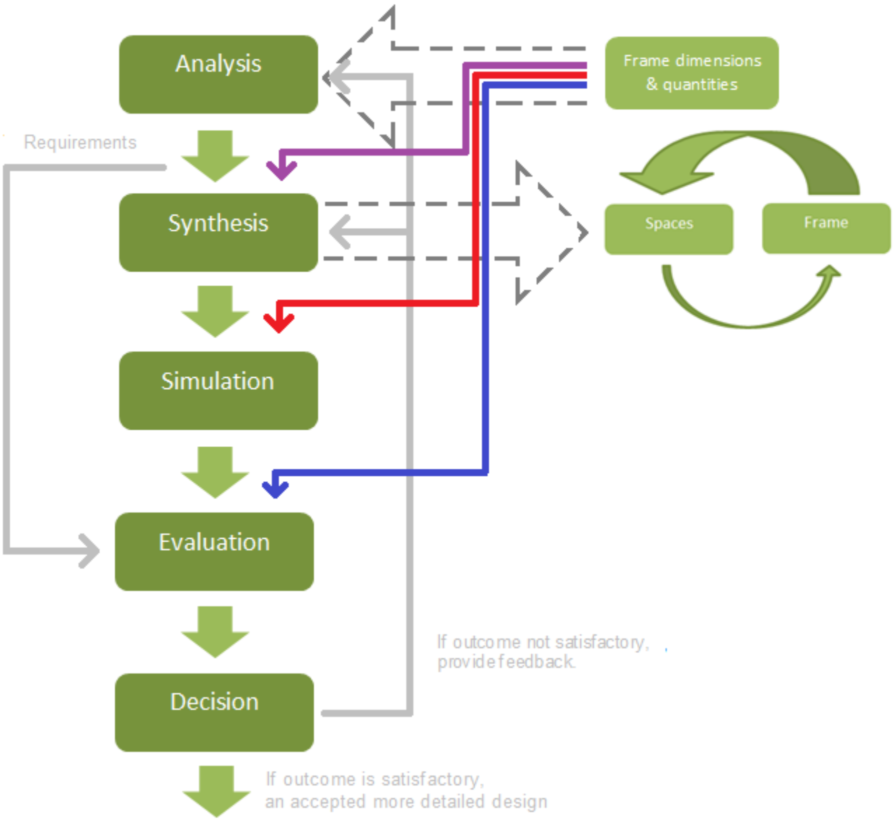


Figure 3.1: Possible applcation implementation moments in the design iteration cycle.

Implementation Options

The first strategy is implementing the elements in the 'Synthesis' step (purple line in Figure 3.1). Using this strategy the constructed frames consist of the available reusable elements and their dimensions, so the frames can only differ in configuration. Leading to a relatively high loss in flexibility compared

to the other strategies, reducing the influence of the 'Spaces' on the 'Frame'. Meanwhile this strategy assures that the designed frame respects the type of elements available, making it more likely that the resulting frame satisfies the reuse constraint created by the fixed set of elements (Section 2.3).
The red strategy imports the available element set in the 'Simulation' step. The designed frame is constructed entirely out of element types available in the reusable element set, but the lengths of the elements are defined in the synthesis phase, to better fit the demands for the 'Spaces'. This means the method gives more design flexibility, but potentially reduces the structural mass reuse percentage and also reducing the structural capacity utilisation because the elements are reduced in length, reducing the internal forces when loads stay the same. Thus in the red strategy the reuse constraint is less leading compared to the purple strategy, but the design flexibility is increased.
The blue strategy implements the reusable set one step later than the red strategy, following the traditional design iteration steps until evaluation. During evaluation the dimensioned element set is compared to the reusable element set. Differences can now not only occur in the amount and length of the elements used, but also on their type. So the blue strategy gives even more design freedom than the red strategy, but again reduces the influence of the reuse constraint in the design cycle.

Frame Configuration versus building shape

Looking at the research goal, creating a frame consisting of elements from the reusable set is one of the key fitness parameters. This parameter possibly conflicts with the next part of the research goal stating that the structural frame should meet the functional requirements for the spaces. Since the spaces are represented by the building mass study and the elements are fixed in size, an exact match on lengths is not likely to occur if both are defined independent from each other.
For this reason a design decision needs to be made whether following the building mass shape or reusing the element set is governing. Cutting the reusable elements into the right dimensions requires energy/increases costs, while aiming to save energy is one of the incentives for reuse. Also making costs during deconstruction is one of the bottlenecks for applying reuse in practice (Section 2.1). For these reasons the design choice is made that reusing the elements without re-dimensioning the elements is leading over following the building mass shape.

Iteration Cycle Step Decision

Due to the design decision that element reuse is governing over shape, the purple strategy in Figure 3.1 is preferred. The other two options discussed are less direct in converging to the available element set. They have a larger search space, without containing higher fitness solutions, therefor needlessly taking more calculation time.

Flexibility versus Information

The reuse application will be build for reusing structural elements, due to this choice and the proposed construction phase separation from the IFD building methods 'Slim Bouwen' and 'Open Bouwen', the choice is made to set the increased design difficulty caused by design aspects being on different detailing levels out of scope [Lichtenberg, 2005] [Zeiler, 2009].
When looking at the detailing level of these structural elements (Figure 2.11), the geometric and material properties of the elements are set, as are the quantities of all the elements available. This means that the remaining design flexibility consists of the frame configuration.
Consequently the remaining design problem is creating the structural configuration that follows the architectural shape, while meeting the floorspace requirements and reusing the highest possible percentage of structural elements.

3.2 Application Input

The aim of this thesis is to follow the quote by Ron Johnson, defining the clear point of view for the project. Now it is known where in the design chain the application is implemented, the next step is defining the input available at this design stage, input forming the boundary conditions and starting information for the application.

Available Input

The design stage where the application is implemented is the 'Conceptual Design' stage (Figure 3.2). During this phase the available data consists of the list of requirements (LoR), the available set of reuseable elements and a building mass study.

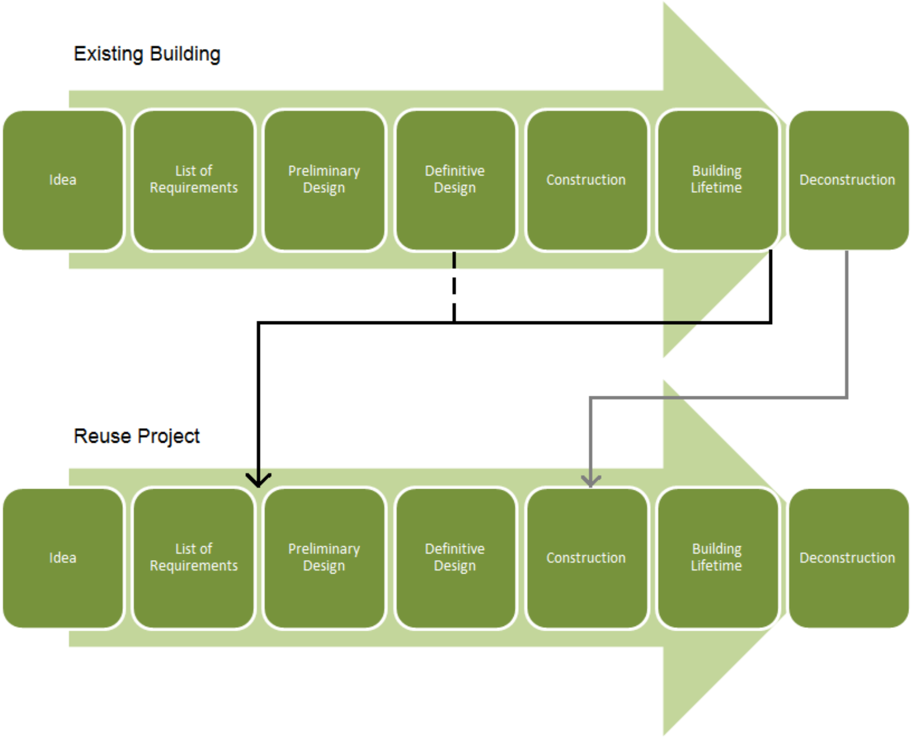


Figure 3.2: Ideal interaction moment between design 1 and design 2, based on required design flexibility.

List of Requirements

The LoR from a building consist of demands and wishes from the project principle and its stakeholders. It contains information on the function of the building and the relation between, and sizes of the spaces inside it. The designing party itself is responsible for proving the structural safety in accordance with the

building codes.
Regarding the relation and sizes between the spaces inside the building, the application will only take into account the required total square meter demand. Ordening the different spaces inside the structure is assumed to be done on the lower element level of 'Non Structural Building Elements' (Section 2.2). This decision sets an upper boundary on the room height and the size of column free spaces. This means the application end product is limited in its solution type, which can reduce the solution feasibility for buildings requiring high rooms or large column free spaces.
Other aspects in the LoR refer to aspects on comfort, logistics and building esthetics. In the reuse application, the comfort and logistic demands, are to be solved on a lower element level, being 'Installations' and 'Non Structural Building Elements' (Section 2.2). The aesthetic demands are accounted for in the application by following the shape of the building mass, other building influences on the aesthetics, like the facade are again covered on the lower element level 'Non Structural Building Elements'.

Reusable Element List

The reusable element list is an inventory containing all the reusable floor plates, columns and beams in the existing building that will be deconstructed. From these elements all the geometric and structural properties are known. This list can be derived by hand or following from a 3D building model containing this element information. For each element type it is important to know the geometric properties, structural properties and quantities [Glias 2013].

Building Mass Study

Due to the influence of the structural frame on the spaces and form of the total building (Figure 3.3), the spaces should not be identified into detail too far, because than the elements are unlikely to fit into the new structure, however some spacial guidance is required to come to a frame suited for the building project.

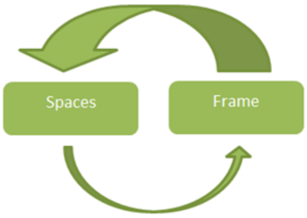


Figure 3.3: Interaction between the spaces and the frame during design with reusable elements.

The aesthetic input ideally would be on the most basic level, giving guidance without limiting the flexibility to fit the available elements, to this end the aesthetic input for the application will consist of a building mass study. Which is a study resulting in the global forms of the building based on the aesthetic demands defined in the list of requirements. The building mass is a raw geometric shape, connecting to the surrounding urban environment. The architect performs a building mass study as a first step in the design process, so this information will be available during the 'Conceptual Design Phase' where the application is implemented.

Building Scaling

In case the square meter requirement for the reusable building and the total amount of available floor plate square meters in the reusable element set differ, the application requires a strategy to handle this. Looking at the research goal, finding a structural configuration reusing as many element from the available set as possible is preferred. For this reason the decision is made to rather add some new elements to finish a building, than to neglect a part of the reusable set because of a misfit in floor space requirements.[1]
To determine the feasible amount of reuse buildings, the application performs a floorspace calculation on the building mass model and divide the available amount of floor plate square meters through the square meter building demand. Rounding of this number upwards, because adding elements is preferred, results in the amount of buildings that will be constructed from the available element set. (Figure 3.4)

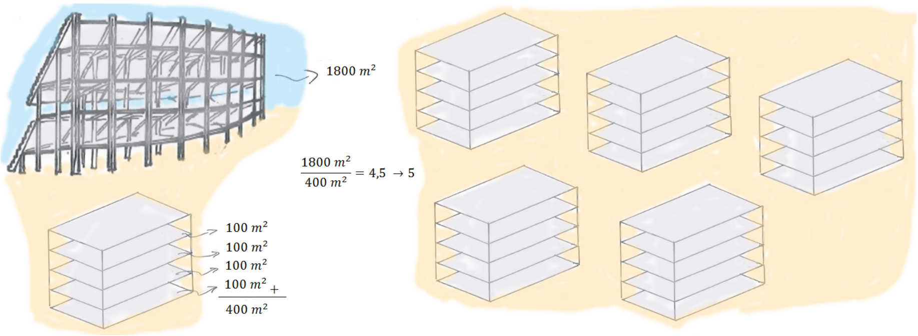


Figure 3.4: Defining the feasible maximum amount of reuse buildings.

During the floorspace calculation based on the building mass, an amount of floors in the building needs to be assumed. This is done by dividing the total height of the building mass through the largest available column, giving an lower boundary estimation of the amount of floors. Ensuring that the output of the application will more likely use more floor plates than that there are available, again requiring extra production rather than leaving elements that are not going to be reused.

---

[1] The decision to rather add elements than to neglect is influenced by company strategy from the Royal BAM Group.

3.3 Application Output

The significance of output, making it mean something so someone, is key to the success of the reuse application. Therefore this section goes into the output of the application, output providing insight on the way the elements are reused and output required to continue the reuse building design until it is ready for the construction phase.

Structural Frame Configuration

The research goal asks for a computational reuse strategy, linking the reusable elements directly to their new function. Doing so based on the functional requirements of the structure, while utilising the structural capacity of the elements.
The functional requirements follow from the LoR and the building mass study, setting a shape for the building, a shape requiring a structural frame. Creating a frame that matches the shape and reuses the available element set while utilising the structural element capacity, is the application goal.

Frame Visualisation

To make the output tangible, a visual output component is required, showing the frame configuration in three dimensions, allowing a check on whether the frame meets the shape requirements set by the building mass study (Figure 3.5)

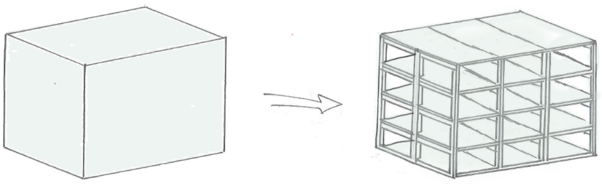


Figure 3.5: Translation of the building mass into the structural frame.

Reuse Percentages

Besides the graphical output, numerical data on reuse is required to inform the user how the frame utilises the available element set. Showing which elements are reused, which are not and how many elements need to be produced to complete the design. This data provides the user with the data required to calculate the feasibility of the reuse project and provides insight in how well the available set is reused, which is part of the application goal.

Next Design Steps

With the output being a structural frame, accompanied by the reuse percentages, floor space per building and reuse coordinates of each element, the design is not yet completed. The next step in design, according to the literature review (Section 2.2), would be to define the (global) stability mechanisms and non structural elements, like interior walls and the facade. While placing the interior walls, the architect and structural engineers need to work in cooperation to ensure the functions are located in such a way that the structural element capacity is utilised as much as possible.
For the structural safety (global) stability and strength calculations will be performed.

3.4 Building Subsets

With the in and output of the application known, the inner workings will be determined next. How to order and arrange the available building elements to come to the structural frame best fitting the research goal.
This section will focus on two aspects, defining the building subsets, which are groups of structural elements and the rules for arranging these subsets inside the building, which together form the structural frame.

Subset Choice

Looking to the element levels (Section 2.2), 'Slim Bouwen' stated that the construction world has been building the same way for years and years. There is a lot of innovation on a component and detail level, however the structural configuration has been roughly the same for thousands of years [Lichtenberg, 2005]. This configuration consists of columns connected to beams and then connected to either a roof or floor plates forming the next building level (Figure 3.6). This basic setup for a structural frame will be used as building block to fill the building mass. These blocks will from now on be referred to as subsets.

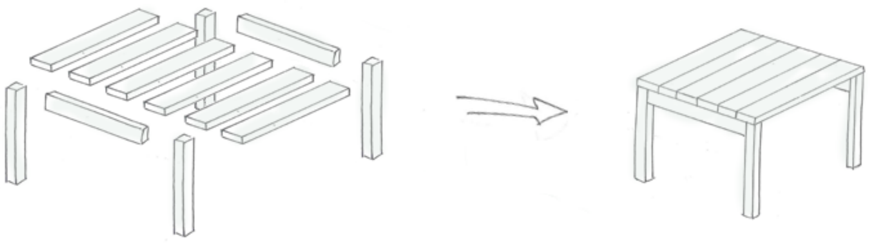


Figure 3.6: Each subset contains floor plates, beams and columns in the configuration shown above.

Using merely this kind of building block influences the possible building shape, every outcome will consist of rectangular shapes. Any other shape would require non rectangular floor plats and/or an irregular supporting structure. These solution types are common practice, however since this thesis is a feasibility study into reuse, these more complex building shapes are set out of scope. The application will merely use the subsets defined in Figure 3.6 and the element types required to create this shape, being floor plates, beams and columns.

Subset to Building Mass Interaction

The application will place these subsets into the shape defined during the building mass study, ordening the subsets in such a way that they best match the building shape. The flexibility the application has to approach this shape is by stacking or placing multiple subsets next to each other, change the type of reusable element used inside the subset or rotate the subsets by 90 degrees. (Figure 3.7)

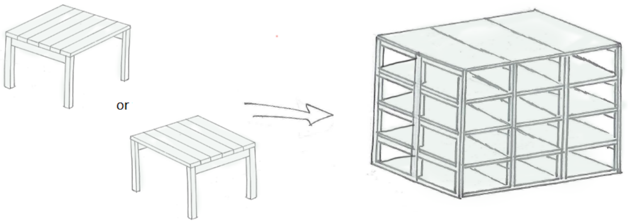


Figure 3.7: Placing the subsets into the building mass to create a form, best following the building mass shape.

Building Layers

The ordening process of the building subsets proposed in the previous subsection comes with some practical implications, all these implication follow from the wish to reduce the amount of actions needed to construct the reuse building. A wish defined in the light of reducing costs and energy use. These practical implications are sketched in this subsection.

Subset Element Types

A subset contains columns, beams and floor plates, to improve the constructability of subset, the structural elements inside a subset will need to consist of the same element type (Figure 3.8). This way no elements need to be adjusted in size in order to fit, reducing the amount of actions required to construct the building. This leads to the application design choice:
Every subset consist of 1 floor plate, column and beam type.

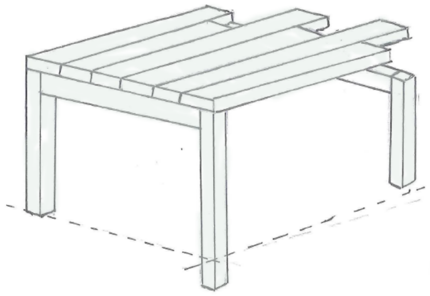


Figure 3.8: Use elements with same type and geometry in one subset.

Floor Plate Implication

In the x-direction sketched in Figure 3.9, every subset needs to consist of the same type of floor plate. Ensuring that no extra columns need to be applied which will make the floor plan less flexible, this repetition will also speed up the building process due to the generic character of the connections that will be applied (Section 2.5).

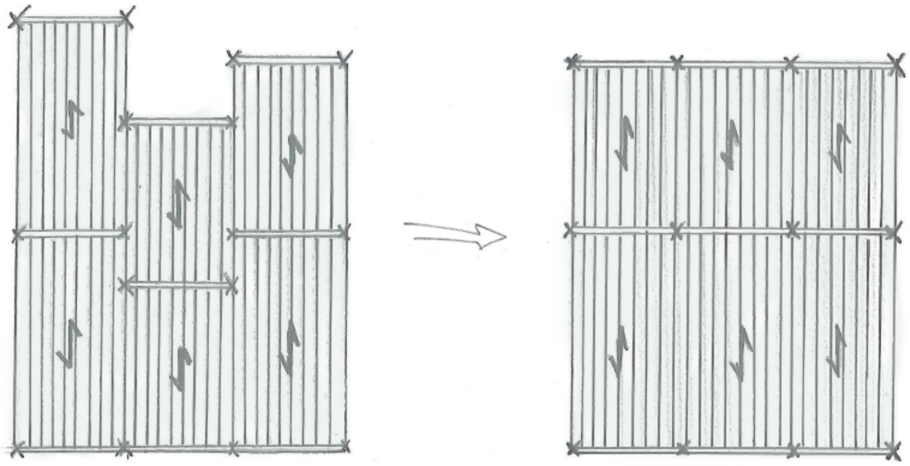


Figure 3.9: Use the same type of floorplates in subsets connected at the length side of the floorplates.

Beam Implication

In the y-direction sketched in Figure 3.10, the same type of implication identified at the floor plates is occurs for the beams.

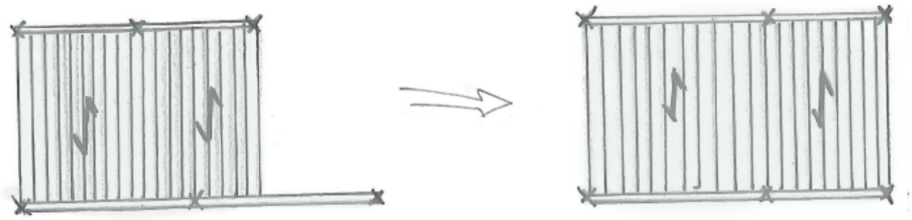


Figure 3.10: Use same type of beams in subsets connected at the width side of the floorplates.

Column Implication

In the z-direction sketched in Figure 3.11, the design assumption is made that every column on a floor is from the same type. Causing no changes to be made to the columns during reuse, ensuring a faster and cheaper building process.

Layer Definition

The above design assumptions cause the building to consists of certain layers. Each layer in a certain direction has subsets with respectively the same type of floor plate, beam or column (Figure 3.12). Changing around the types of elements in each layer and choosing the amounts on subsets per layer inside the building will lead to the structural frame fitting the building shape and

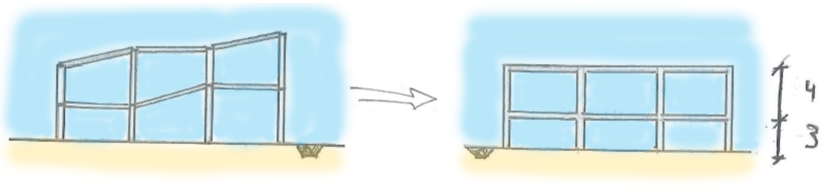


Figure 3.11: Use same type of columns per building level.

using the available element set. Finding the solution best fitting the boundary conditions is done by the genetic reuse algorithm, which is the topic for the next subsection.

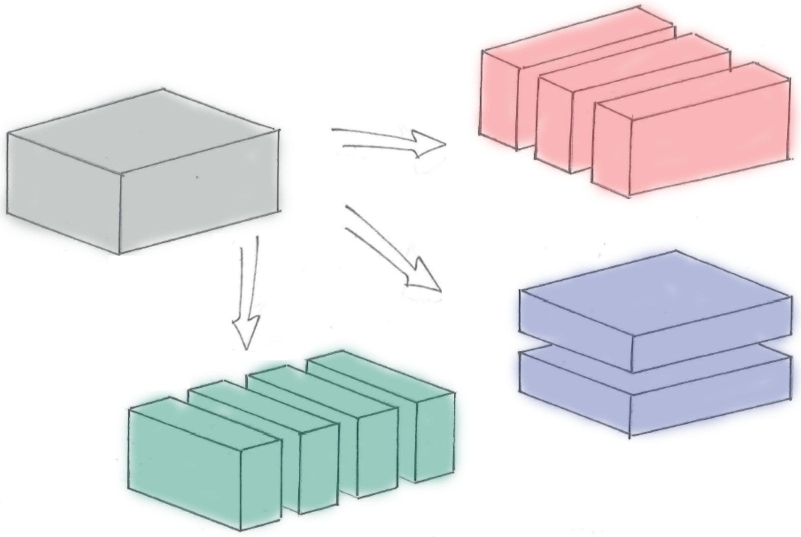


Figure 3.12: Equal element properties in subsets per layer for respectively the floor plates (green), beams (red) or columns (blue).

3.5 Genetic Reuse Algorithm

The process of information storage and transmission in a Genetic Algorithm has been discussed in Section 2.5. This section elaborates on which information will be embedded inside a chromosome for the reuse algorithm, showing how the design flexibility is implemented in the optimisation process.

Layer Genes

The layer system, introduced in the previous section, will be part of the chromosome, with each layer in the three directions representing a gene and the alleles of the gene consisting of the available element types in this direction. (Figure 3.13)

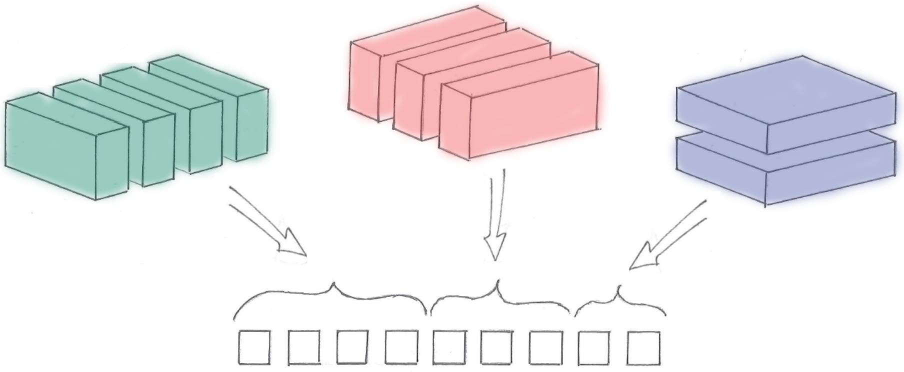


Figure 3.13: The element type inside a layer is represented in the algorithm chromosome by a gene.

Grid Size Genes

The previous section defined that the amount of layers in each direction is also an algorithm variable. Therefor the amount of layers in each direction needs to be part of the chromosome.

Building Layer Amount

The amount of layers in each direction together form a grid of subsets (Figure 3.14). The maximum amount of layers in each direction is based on the minimum length of each element type per direction, this way the upper boundary for the amount of layers is defined, making sure that its possible to reach the required building size with the smallest elements in each subset.

Layers inside Building Shape

The algorithm chromosome contains genes from which the alleles specify how many subsets in each layer are present inside the building, changing around the minimum and maximum grid number of the layers that fall inside the building

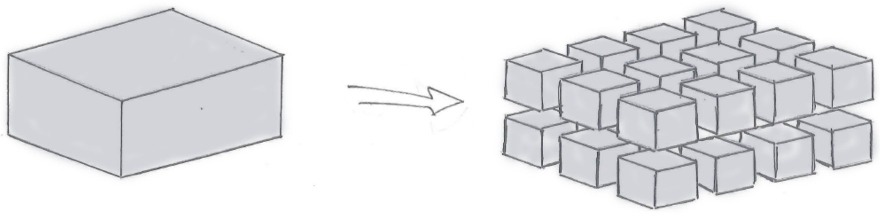


Figure 3.14: Grid representing the maximum amount of subsets in each direction.

shape (Figure 3.15). Because some element types are larger than the ones used to define the amount of layers per direction, the fittest solution might not require the use of every layer.

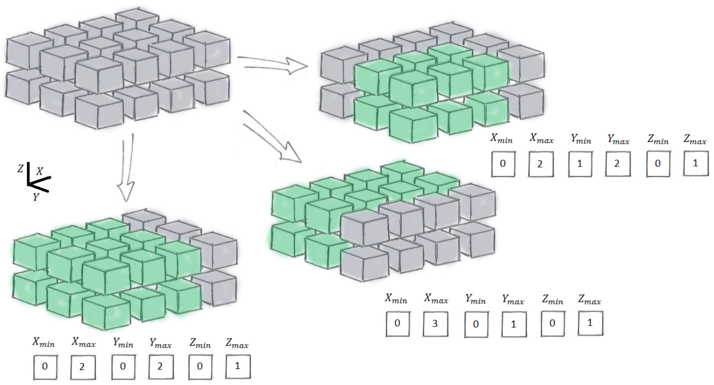


Figure 3.15: Layers that are inside the building shape are shown in green and represented in the chromosome as shown by the numbers inside the genes.

Building Rotation

The next element in the algorithm chromosome is the rotation of the building. Due to the different lengths of the beams and floor plates, rotating them leads to new, and possibly better, solutions. In the chromosome this is represented by a gene, able to change between two options, representing rotation or no rotation (Figure 3.16).

Building Chromosome

Placing the genes representing the building layers, layers inside the shape and the rotation gene together, form the chromosome of one building. This build up is shown in Figure 3.17.

Multiple Buildings

The last aspect incorporated into the algorithm chromosome is the amount of buildings. Due to the fixed set of reusable elements with possible different

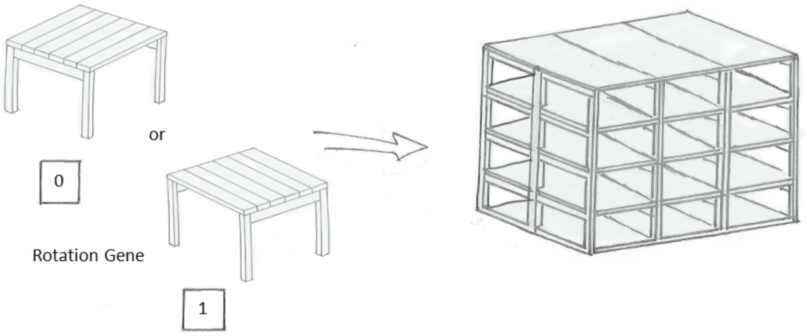


Figure 3.16: Gene inside the chromosome representing in which way the subsets are placed inside the building mass.

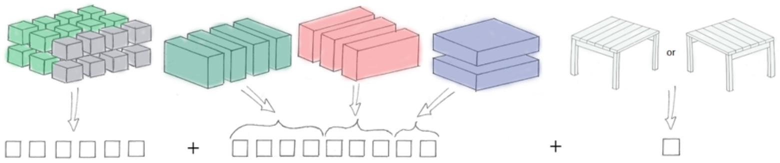


Figure 3.17: The three aspects discussed so far in this section, together form the chromosome of one building.

element types, it is possible that for each building a different configuration of elements is optimal. To accommodate this flexibility, every building will have its own set of genes (Figure 3.18).

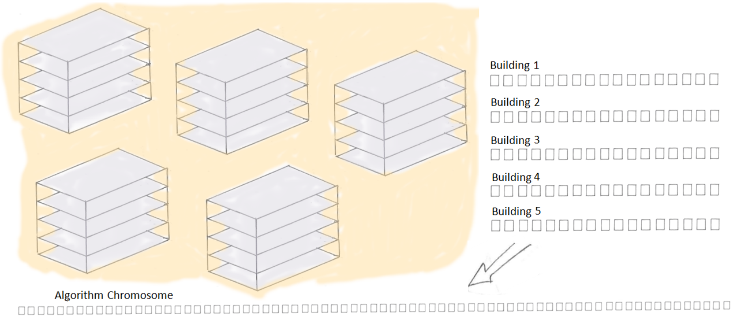


Figure 3.18: Each building has its own set of genes inside the chromosome.

Building Features

The aspects presented in the previous sections are all implemented in the chromosome, however some building features are implemented in the final design but do not require chromosome genes.

Bottom Floor

Each subset consists of columns, with on top of that floor beams and plates. However the first floor is also constructed out of these floor plates and beams. Therefor at ground level, the exact same beam and plate elements will be implemented as on the building levels above. (Figure 3.19)
The elements on the ground level do contribute to the functional floorspace of the building, the elements used as building roof are left out of the area summation.

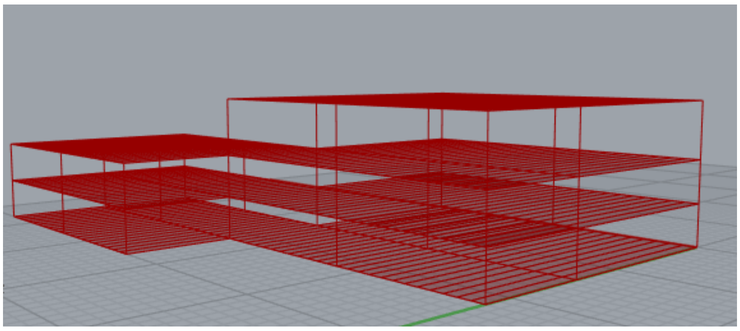


Figure 3.19: On ground level also beams and floor plates are positioned.

Vertical Transport solution

Figure 3.19 also shows that the building solutions do not implement locations for vertical transport. This application design decision is based on the fact that during the creation of the building mass, no data on this aspect is known. It is a decision made later in the design process, vertical transport falls under the building phase of 'Installaties' and 'Inbouw', which are building phases done after the structural frame is determined [Lichtenberg, 2005].
However decisions on vertical transport locations also influence the structural frame, for this reason it is an aspect not clearly assigned to one of the building phases identified by the theory of 'Slimbouwen' whih is used as design method in this thesis. The implementation of this aspect is therefor one left to the application creator. Whom chose not to implement it and left the vertical transport locations and implications for the structural frame to be identified in a later design stage.

The application implementation of every design aspect discussed in this chapter is shown in the next.

4 | Strategy Implementation

"Higher-level tools that actually let you see the structure of the software more clearly will be of tremendous value."

- Guido van Rossum -

Now the application design is set, this chapter translates the design into a model implementing the design. For this reason an UML-model is created based on the application design from chapter 3. Followed by the implementation of this UML-model into a grasshopper application.
The first chapter section goes into the chosen modeling type and presents the UML-model. The other sections in this chapter will start by showing the UML-model part that is being viewed, followed by the Grasshopper components representing these UML classes. After each part of the model is discussed the working of the application will be validated in the chapter 5.

4.1 Application Modeling

This section elaborates on the software structure of the reuse application. Doing so for explaining the application design itself, the application creator point of view, and also how this principle is implemented from the users point of view.

Parametric Associative Design

The implementation of the UML model proposed in the previous chapter (Figure 4.2), is done through the use of 'Parametric Associative Modeling' (PAM). These models generates output, often geometry, based on user defined parameters and relations (associations) between those parameters. The parameters allow adjustment during the design process, offering the ability to quickly compose, adjust and assess different design alternatives [Coenders, 2007]. These adjustment options and visible relations/associations between the components increase the feeling of influence and control the user has on the application. These aspects increase the likelihood that the user will implement the software application in practice. For this reason PAM is used for the implementation of the reuse application.

Figure 4.1: Used software environments for 'Parametric Associative Systems'.

The chosen software environment implementing 'Parametric Associative Modeling' is Grasshopper, which is an algorithm editor integrated with Rhino's 3-D modeling tools. The Grasshopper environment is chosen because it enables designers to implement parametric or associative design without extensive knowledge in scripting/programming, causing the designed reuse application to be better accessible and useable after it is completed. (Figure 4.1)

UML-Model

This paragraph presents the structure of the reuse application. The structure is presented as an UML model. The UML content is based the aspects discussed in the chapter four.

UML Elaboration

This section describes where each aspect defined in chapter 3 is represented in the UML-model, giving a brief elaboration on the application flow.
The input defined in Section 3.2 contained a list of structural elements from an existing building. These elements are represented by the four classes in the left

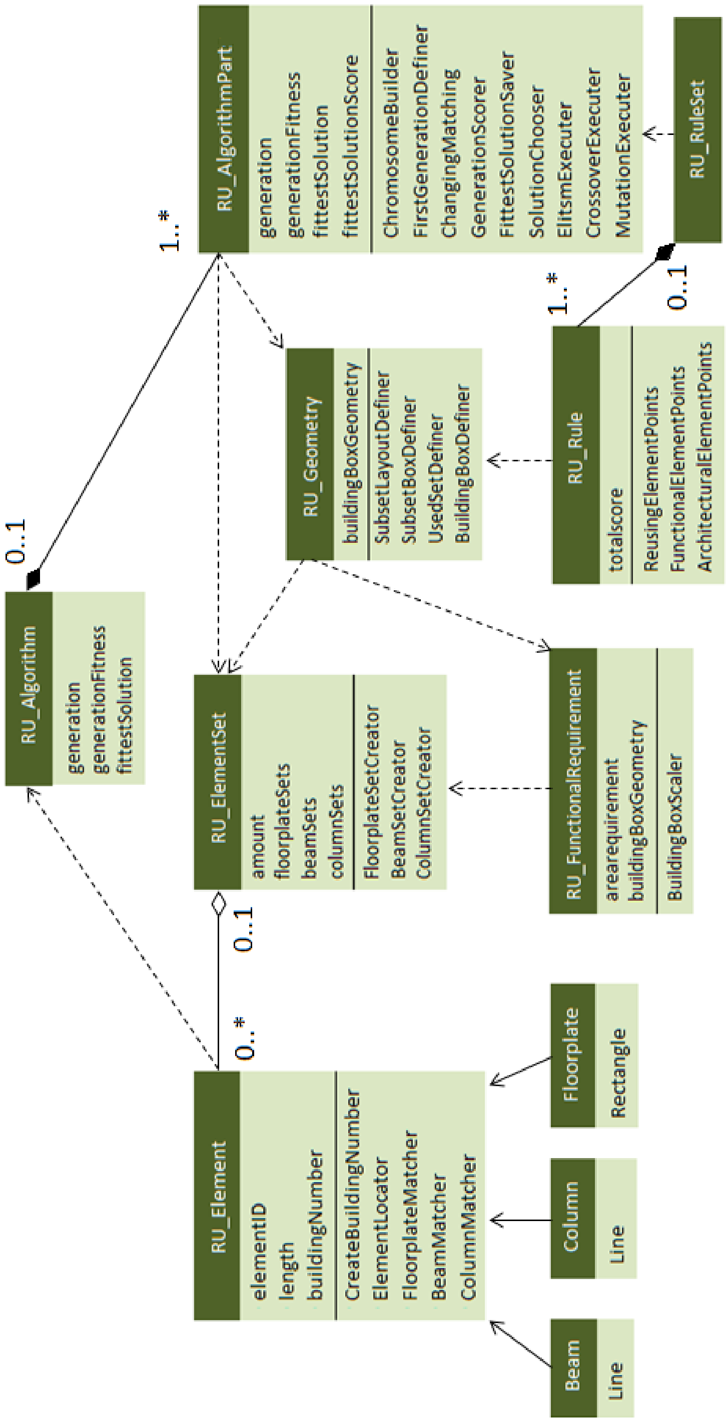


Figure 4.2: UML model for the computational reuse application.

of the UML model, being 'RU_Element', 'Column', 'Beam' and 'Floorplate'. These structural elements and their properties are used in the next class called 'RU_ElementSet' to define the possible contents of the subsets from Section 3.4. Next up are the functional requirements in the class RU_FunctionalRequirements, holding the other input aspects from Section 3.2, being the building mass and the functional floorspace requirement. By using the floor plate data from the 'RU_ElementSet' class, the amount of feasible reuse buildings can be determined.
With all the input from Section 3.2 implemented in the application, one more thing needs to be done before the algorithm can run: Defining the subset grid as mentioned in Section 3.4. This is done in the 'RU_Geometry' class, using the element size data from the 'RU_Element' set class and the scaled building mass from the 'RU_FunctionalRequirement' class.
With the subset grid determined, all the information is present to build the chromosome inside the 'RU_AlgorithmParts' class, defining the possible allele values from the element set and building geometry genes. The last action before the algorithm can run is defining the fitness rules, calculating how well each solution meets the application goals. These rules will be based on the research goal and are defined in Section 4.4.
After the rules are defined, the algorithm can run and the fittest solution is used by the 'RU_Element' class to locate all the elements in the structural frame of the fittest solution.

Because Grasshopper does not contain the definitions required to form the reuse application proposed in the UML model (Figure 4.2). These definitions were programmed by the author, the translation of the UML-model into these grasshopper definitions is discussed in the remainder of this chapter.

4.2 Input Implementation

This section elaborates on how the user input is imported into the system. Explaining the actions a user needs to undertake, and how the imported data is handled by the application. Setting up the data in such a way that it can be used for creating the subset grid and the algorithm, which will be discussed in the Sections 4.3 and 4.4.

UML Input Classes

Figure 4.3 highlights the classes that handle the reuse input. The 'RU_Element' class handles the reuse element data, consisting of 'Column', 'Beam' and 'Floorplate' types, which are modeled as derived classes in the UML model. Each of these classes inherits the members and methods from the 'RU_Element' class and adds its own geometry type. Each element instance in this class has its own geometric members, the class adds location ('buildingNumber') and visualising ('line' or 'rectangle') members which will be assigned later in the process.

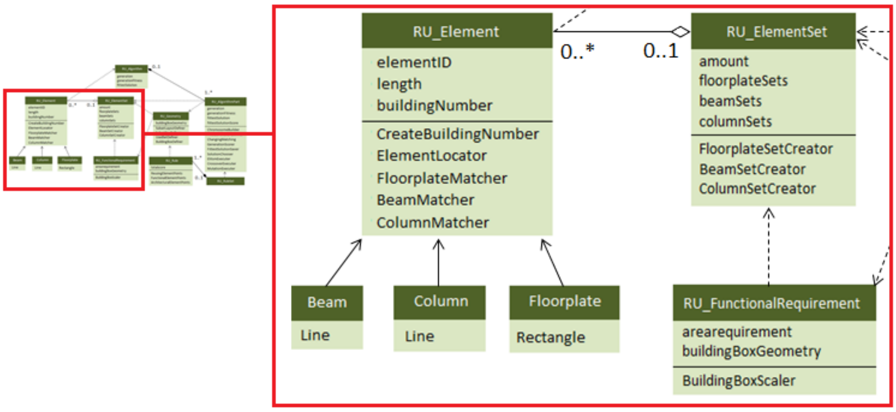


Figure 4.3: Classes involved in handling the input data.

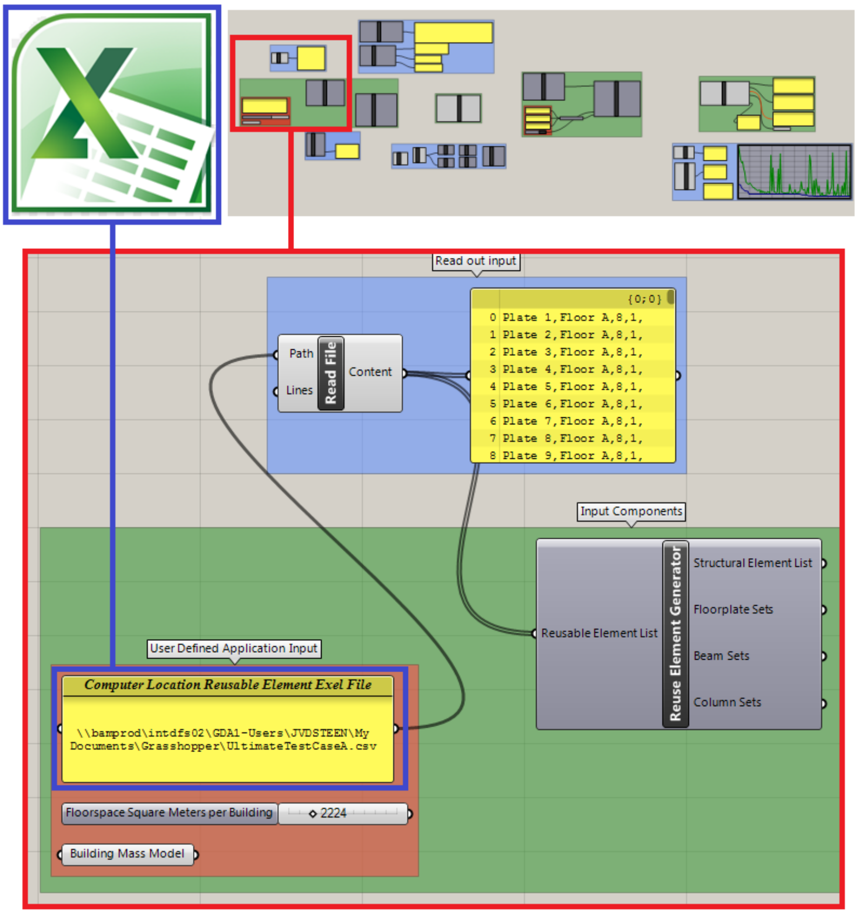
After the element data is set, the elements are divided into element sets with the same geometric specifications in the 'RU_ElementSet' class. This set division is done to identify the geometric options for the subsets when running the algorithm. Also the amount of each element type is calculated to assist in determining the used element set of each solution, which is a solution fitness parameter.

The last class which is part of the input is the 'RU_FunctionalRequirement' class, which handles the last two user defined input data streams: the required functional floorspace and building mass shape. This class scales the introduced building mass shape so that the shape is more likely to hold the required amount of floorspace.

Both this scaled shape and the floorspace requirement are used by the algorithm in determining the solution fitness, how this is done will be explained in Section 4.3.

Grasshopper Input Components

Each of the UML classes from the previous subsection is represented in the Grasshopper application, starting of with the 'RU_Element' class. The input of this class is provided by an excel file containing the geometric data of the reusable elements (Figure 4.4).



```
{0;0}
0 Plate 1,Floor A,8,1,
1 Plate 2,Floor A,8,1,
2 Plate 3,Floor A,8,1,
3 Plate 4,Floor A,8,1,
4 Plate 5,Floor A,8,1,
5 Plate 6,Floor A,8,1,
6 Plate 7,Floor A,8,1,
7 Plate 8,Floor A,8,1,
8 Plate 9,Floor A,8,1,
```

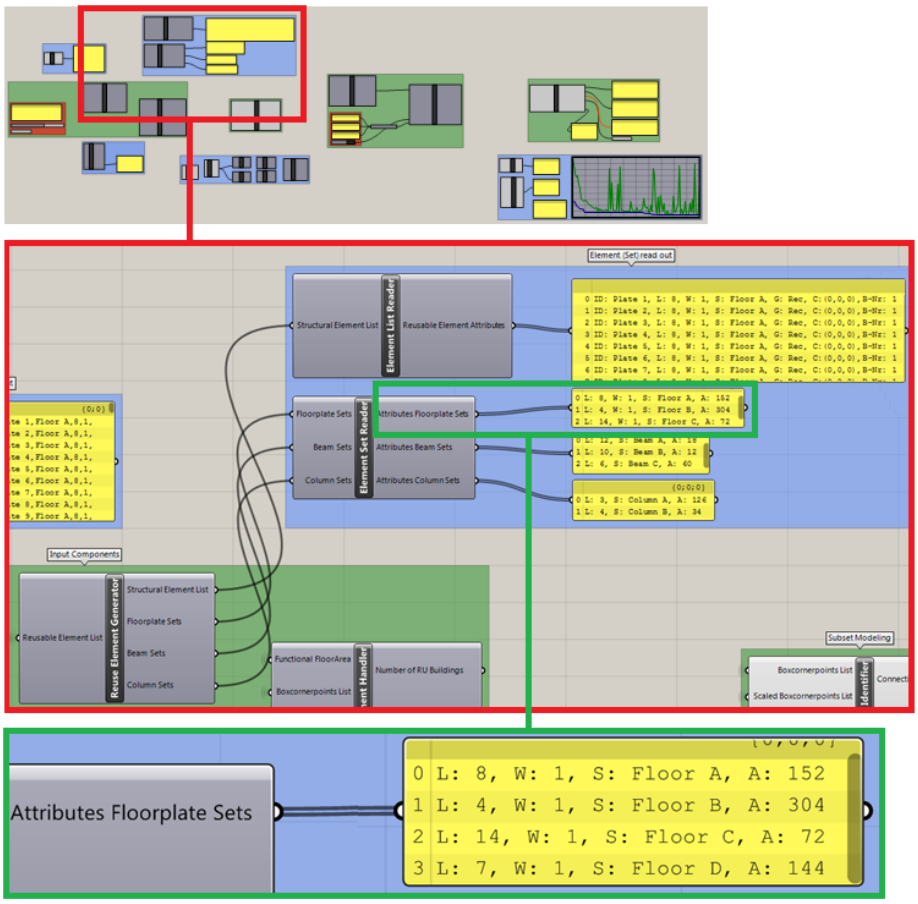
```
\\bamprod\intdfs02\GDA1-Users\JVDSTEEN\My
Documents\Grasshopper\UltimateTestCaseA.csv
```

Floorspace Square Meters per Building: 2224

Figure 4.4: Grasshopper element input implementation

The data is than transferred into the 'Reuse Element Generator' component, which divides the imported elements into sets with the same property, meaning that this component holds the logic representing the 'RU_ElementSet' class (Figure 4.5).

The second input type is the functional floor area, following from the list of requirements. This input together with the 'Building Mass Model', form the input for the 'Functional Requirement Handler' component (Figure 4.6), which is represented in the UML model by the 'RU_FunctionalRequirement' class.



```
{0;0}
te 1,Floor A,8,1,
te 2,Floor A,8,1,
te 3,Floor A,8,1,
te 4,Floor A,8,1,
te 5,Floor A,8,1,
te 6,Floor A,8,1,
te 7,Floor A,8,1,
te 8,Floor A,8,1,
te 9,Floor A,8,1,
```

```
0 ID: Plate 1, L: 8, W: 1, S: Floor A, G: Rec, C:(0,0,0),B-Nr: 1
1 ID: Plate 2, L: 8, W: 1, S: Floor A, G: Rec, C:(0,0,0),B-Nr: 1
2 ID: Plate 3, L: 8, W: 1, S: Floor A, G: Rec, C:(0,0,0),B-Nr: 1
3 ID: Plate 4, L: 8, W: 1, S: Floor A, G: Rec, C:(0,0,0),B-Nr: 1
4 ID: Plate 5, L: 8, W: 1, S: Floor A, G: Rec, C:(0,0,0),B-Nr: 1
5 ID: Plate 6, L: 8, W: 1, S: Floor A, G: Rec, C:(0,0,0),B-Nr: 1
6 ID: Plate 7, L: 8, W: 1, S: Floor A, G: Rec, C:(0,0,0),B-Nr: 1
```

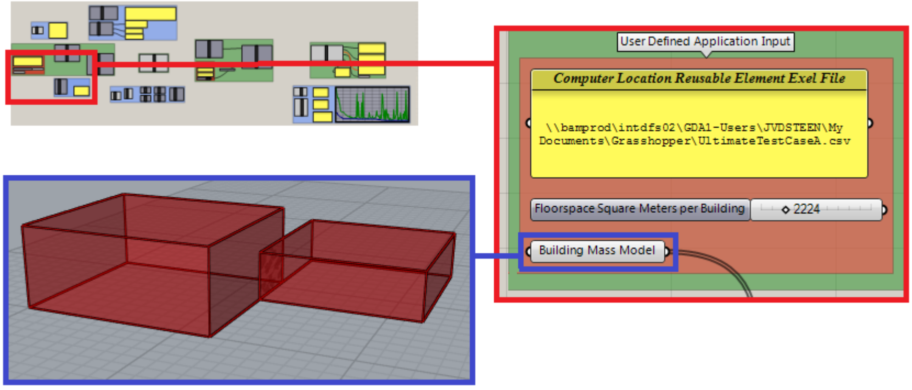
```
0 L: 8, W: 1, S: Floor A, A: 152
1 L: 4, W: 1, S: Floor B, A: 304
2 L: 14, W: 1, S: Floor C, A: 72
```

```
0 L: 12, S: Beam A, A: 18
1 L: 10, S: Beam B, A: 12
2 L: 6, S: Beam C, A: 60
```

```
{0;0;0}
0 L: 3, S: Column A, A: 126
1 L: 4, S: Column B, A: 34
```

```
0 L: 8, W: 1, S: Floor A, A: 152
1 L: 4, W: 1, S: Floor B, A: 304
2 L: 14, W: 1, S: Floor C, A: 72
3 L: 7, W: 1, S: Floor D, A: 144
```

Figure 4.5: Grasshopper element set division, the example shows 4 floor plate sets.



```
\\bamprod\intdfs02\GDA1-Users\JVDSTEEN\My
Documents\Grasshopper\UltimateTestCaseA.csv
```

Floorspace Square Meters per Building: 2224

Figure 4.6: Grasshopper geometry input implementation

This component takes the square meter requirement and building shape, represented as one or two connected box shaped elements and than scales the given building mass to be able to accommodate the required floorspace. (Figure 4.7)

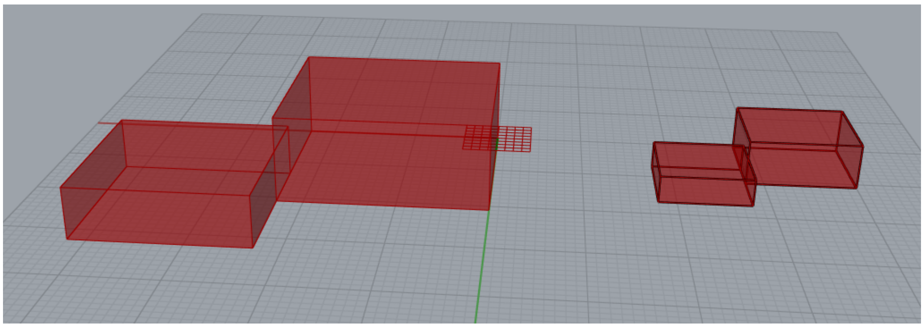


Figure 4.7: The specified building mass (right) is scaled to fit the required amount of floorspace (left).

After scaling the building mass, the component cross-references the required amount of floorspace with the available amount of floor plate square meters in the reusable set. Based on this check, the 'Functional Requirement Handler' determines the feasible amount of reuse buildings. (Figure 4.8)

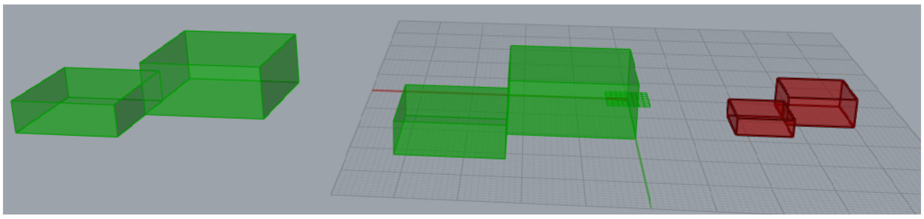


Figure 4.8: Based on the reusable element set, the application determines the feasible amount of reuse buildings (in this case 2).

Now the input is imported into the reuse application, the next step is creating a component which sets up the subset data structure as discussed in Section 3.4.

4.3 Subset Modeling

The main preparation for the reuse algorithm is creating the subset grid. This grid determines the search space of the algorithm, making sure the optimal solution is inside it, while not needlessly enlarging the search space. The method for determining the subsets is explained in Section 3.4, this section goes into the application implementation of this method.

UML Subset Classes

Modeling the subsets is done within the 'RU_Geometry' class (Figure 4.9). The 'RU_Geometry' class uses the building size data from the the scaled building in the 'RU_FunctionalRequirement' class to set an outer shape. The feasible amount of subsets in each direction within this outer shape is determined by deriving the smallest element from each element type in the 'RU_ElementSets' class. The building length is then divided by the smallest element length and rounded upward, resulting in the amount of subsets per direction.

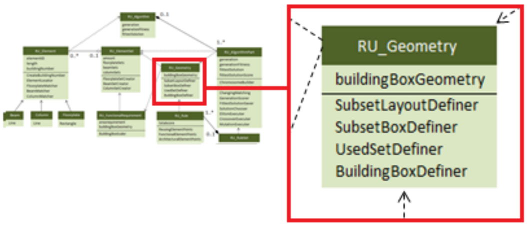


Figure 4.9: Classes involved in setting the Subsets.

Grasshopper Subset Components

On the grasshopper canvas the 'Reuse Geometry Identifier' holds the logic to create the subset grid (Figure 4.10). The input of this component consist of the (scaled) building boxes, which provide data on the sizes of the building mass. The other input for this component consists of the three element sets, used to identify the smallest element in each direction and define the amount of subsets required based on the building mass and this smallest element.

Multiple Building Boxes

As output this component consists of a grid of subsets, set as output in 'Subsets per Direction' (Figure 4.10. The other component output streams are the 'Connection Type' and 'Subsets per Buildingbox'. These are implemented to handle the cases where the architect used more than one building box as building mass shape. A feature implemented in the application to show the potential for handling more complex shapes. This implementation decision is made because the study is a strategy feasibility study and handling more complex shapes will make the application usable for a broader project market, increasing the strategy feasibility. The increased complexity is to this moment limited to accepting

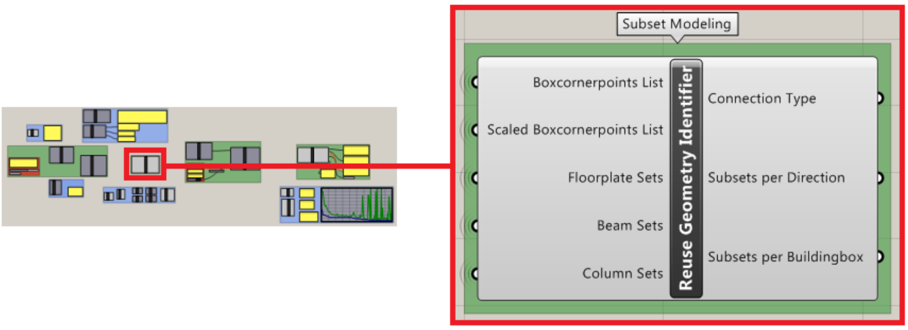


Figure 4.10: Component setting up the subset grid.

two building boxes.
The 'Connection Type' output tells whether there is 'One Box', two boxes connected in the x-plane ('X-connection') or in the y-plane ('Y-connection').
The 'Subsets per Buildingbox' outputs the value ranges of the alleles of the 'subsets in shape' genes in case of multiple building boxes. These ranges are required to ensure that the building boxes wont overlap or not connect inside the subset grid. (Figure 4.11)

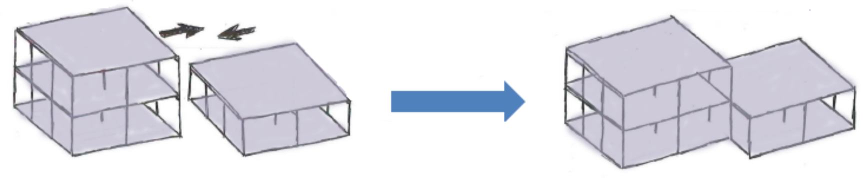


Figure 4.11: Subset ranges ensure that the building boxes wont overlap or leave a gap but connect.

4.4 Reuse Algorithm Implementation

One of the benefits with computational design is, that when performed well, the application performs the hard work, enabling the designer to focus more on implementing design intelligence. The hard work in the reuse application is done by the genetic algorithm, for which the implementation is presented in this section.

UML Algorithm Classes

The genetic reuse algorithm is represented by the UML classes 'RU_Algorithm', 'RU_AlgorithmParts', 'RU_Rule' and 'RU_RuleSet' (Figure 4.12. First up is in the application process is the 'RU_AlgorithmParts' class, which imports the subset grid and possible element types in this subset grid from the 'RU_Geometry' and 'RU_ElementSet' respectively. With this data the reuse chromosome is constructed and the 'RU_Algorithm' class can start up.

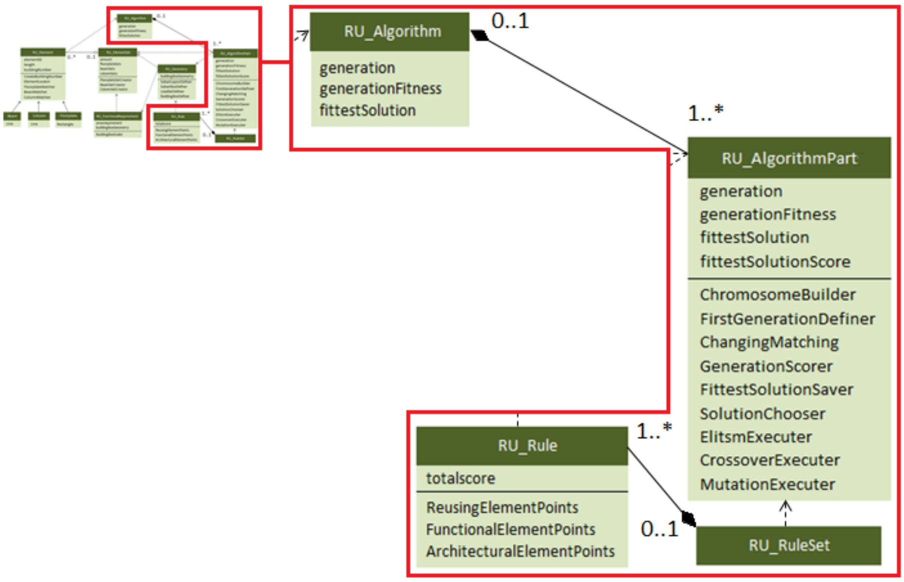


Figure 4.12: Classes involved in running the reuse algorithm.

The 'RU_Algorithm' inherits the chromosome and keeps activating methods in the 'RU_AlgorithmParts' class to execute the algorithm loops. One of these methods is the 'GenerationScorer' method, which activates the 'RU_RuleSet' class, holding a set of 'RU_Rules' which score the fitness of each chromosome in the generation.

During each run these scores are checked to see if a new fittest solution is found, if this is the case its saved by the 'FittestSolutionSaver'. After the score check the algorithm evolves to the next generation by use of selection, crossover and mutation, until a specific amount of runs is achieved or no new fittest solution was found for a set amount of runs. When this happens the 'fittestsolution' is set as output of the 'RU_Algorithm' class so that it can be used to visualise the resulting structural frame.

Grasshopper Algorithm Components

The process sketched in the previous subsection is executed in the grasshopper model by the components shown in Figure 4.13.

Inside Chromosome Builder

The 'InsideChromosomeBuilder' component creates the chromosome, with the chromosome length based on the subset grid defined by the 'Reuse Geometry Definer' component output subsets per direction and subsets per buidlingbox. The value ranges for each gene in the chromosome is defined by passing the max value of each gene in the chromosome. The max value of the genes specifying which element type the subset uses are defined by the amount of element sets and the max values of the subset box grid defined by the subset per direction input. The chromosome is completed by adding the gene for the building orientation and finally this number of genes is copied based on the identified number of feasible reuse buildings.

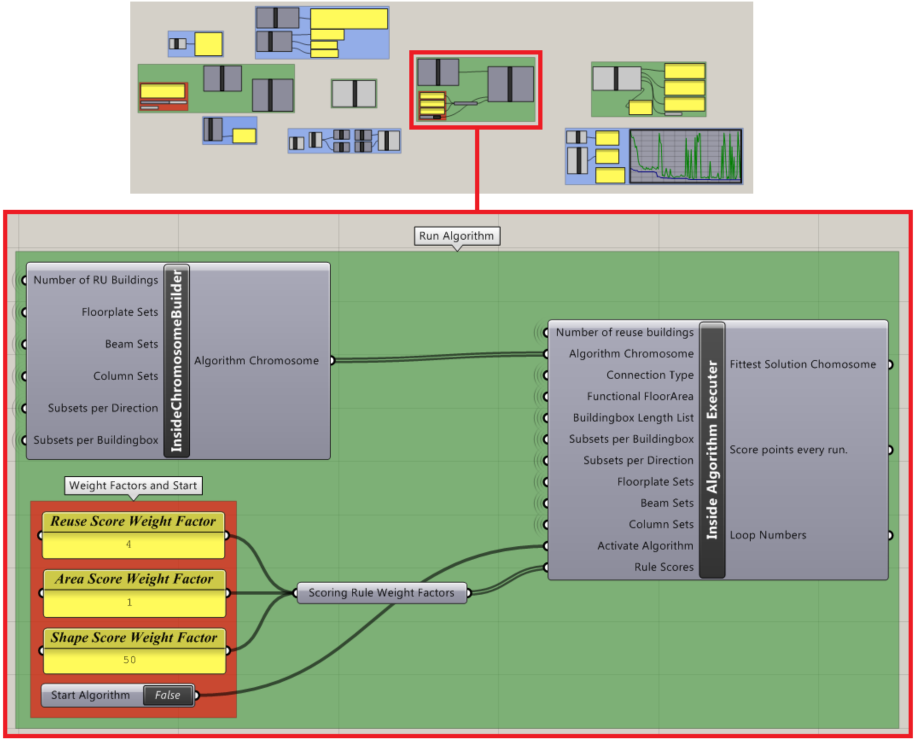


Figure 4.13: Components implementing the algorithm, setting it up, starting it and running through the loops.

Scoring Weight Factors

The scoring weight factors are a set of user defined numbers determining the importance of each algorithm rule. Every rule has one weight factor, with which his score is multiplied. (Figure 4.13)
The reason that these factors are left for the user to define is that the research question does not specify the level of relevance between each of the building aspects. This specification is set as a design choice, allowing the user to specify its own ideal configuration, again increasing the users influence on in the application.

Inside Algorithm Executer

The next component, 'Inside Algorithm Executer', runs the algorithm loops when the 'Start Algorithm' condition is set to true (Figure 4.13). When the looping process is completed the 'Inside Algorithm Executer' returns the chromosome with the highest fitness based on the fitness formula:

$$F = \sum_{n=1}^{B} |A_{sn} - A_r| * W_a + \sum_{i=1}^{ES} |E_{ui} - E_{ai}| * W_e + \sum_{j=1}^{S} |V_{cj} - V_{bj}| * W_s$$

$F =$ Fitness Score
$B =$ Number of Buildings
$A_{sn} =$ Area supplied in Building n
$A_r =$ Area required
$W_a =$ Weight Factor Area
$ES =$ Element Set
$E_{ui} =$ Elements used from Set i
$E_{ai} =$ Elements available in Set i
$W_e =$ Weight Factor Elements
$S =$ Number of building boxes
$V_{cj} =$ Volume created at building box j
$V_{bj} =$ Volume supplied building mass for building box j
$W_s =$ Weight Factor Shape

The solution scoring the lowest value in the fitness formula is used to visualise the building geometry. The application also returns the fitness score development during the looping process, indicating whether the algorithm shows the genetic behaviour (discussed in Chapter 5, Algorithm Validation). Finally the application presents the loop numbers in which a new fittest solution was found. This last piece of information is used to gain insight in how fast the algorithm found the final outcome.

Algorithm Stopping Criteria

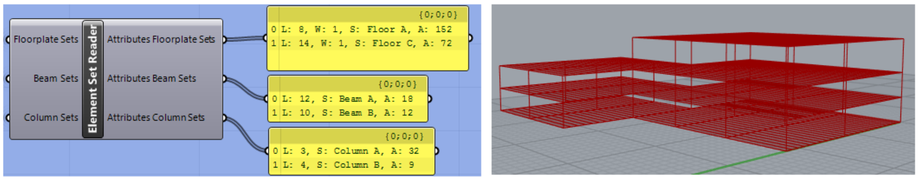
One of the downsides of a genetic algorithm is that there is no absolute certainty the fittest solution has been found. This can only be done after every solution has been investigated. This aspect influences the definition of the algorithm stopping criteria. Because these will be based on a certain likeliness no better solution will be found, instead of a certain statement the fittest solution is found. Based on this information the stopping criteria are set to trigger based on two different events. The first event is when the fitness score reaches the value of zero. Meaning there is a perfect match and there is absolute certainty no fitter solution can be found by the algorithm. The second event is an empiric value obtained during testing of the algorithm. This second event is that the algorithm stops in case that for 100.000 runs no new fittest value has been found. This value is chosen because during algorithm testing the maximum gap between a new fittest solutions was never greater than 82.381.

Optimisation Rules Implementation

When the algorithm is determining the fitness of the solutions in each loop, it does so based on three fitness rules which are based on the research goal. These three rules follow from: reusing as many elements as possible (1), while meeting the functional floorarea (2) and shape (3) of the building.

Reused Element Set

The first rule checks how many elements of the reusable set have been implemented in the solution by crosschecking the found frame with the element sets. Elements which are not reused and element types used more than available, score fitness points. (Figure 4.14)



```
{0;0;0}
0 L: 8, W: 1, S: Floor A, A: 152
1 L: 14, W: 1, S: Floor C, A: 72
```

```
{0;0;0}
0 L: 12, S: Beam A, A: 18
1 L: 10, S: Beam B, A: 12
```

```
{0;0;0}
0 L: 3, S: Column A, A: 32
1 L: 4, S: Column B, A: 9
```

Figure 4.14: How many of the available elements (A values in the sets on the left) are reused in the frame (right).

Functional Floorspace

During the input phase of the algorithm, the user identified the ideal amount of functional floorspace inside the building. This reason floorarea is compared with the floorarea used inside each building the solution defined. Each square meter of floor plates too much or too little scores a fitness point. (Figure 4.15)

Architectural Shape

The third rule scoring the solution is the shape of the building compared to the scaled building mass. For each meter the structural frame differs from the user

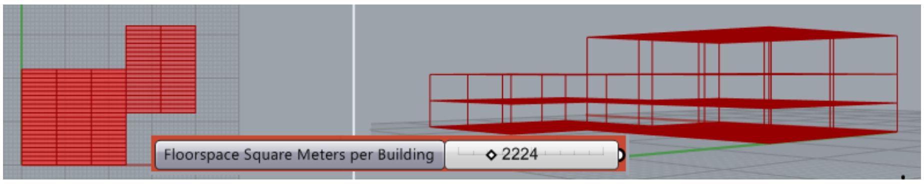


Figure 4.15: Difference between the floor plate area used and the user specified number give the fitness points.

defined building mass the algorithm scores a fitness point. (Figure 4.16)

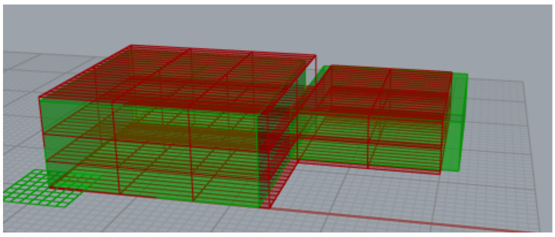


Figure 4.16: Difference between the buidling mass (green) and the structural frame (red), give the fitness points.

Fittest Score

The sum of the fitness score from each rule, multiplied by its weight factor, gives the solution fitness.
Since in each of these rules, the solution scores points on aspects where it does not meet the asked requirements, the fittest solution will be the solution with the lowest score. For this reason the lower scores solutions will have a higher chance of being selected for the next generation during the selection process.

After the algorithm has finished looping, the fittest solution and the scoring milestones will be outputted towards the final component of the algorithm, discussed in the final section of this chapter.

4.5 Application Output

With the algorithm up and running, the next step of the application will be presenting the results. It is important the user feels comfortable and familiar with the system in order to ensure the application will be used. Providing the user with insight and a sense of control are therefor important application success factors. Success factors influencing which output the application shows during and after executing the algorithm, to provide this insight and sense of control. For this reason the implementation of the output is discussed in the following section.

UML Output Classes

The model will provide two types of output, output during the run time of the algorithm and output when the final solution has been found. Looking at the UML-model (Figure 4.17), the classes involved in creating this output are 'RU_Algorithm' and 'RU_Element'.

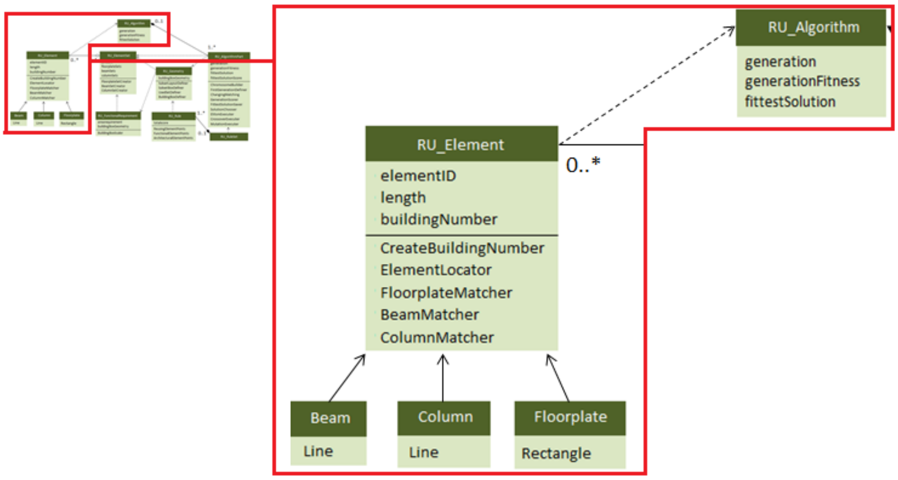


Figure 4.17: Classes involved in the output of the model.

Score Converging Output

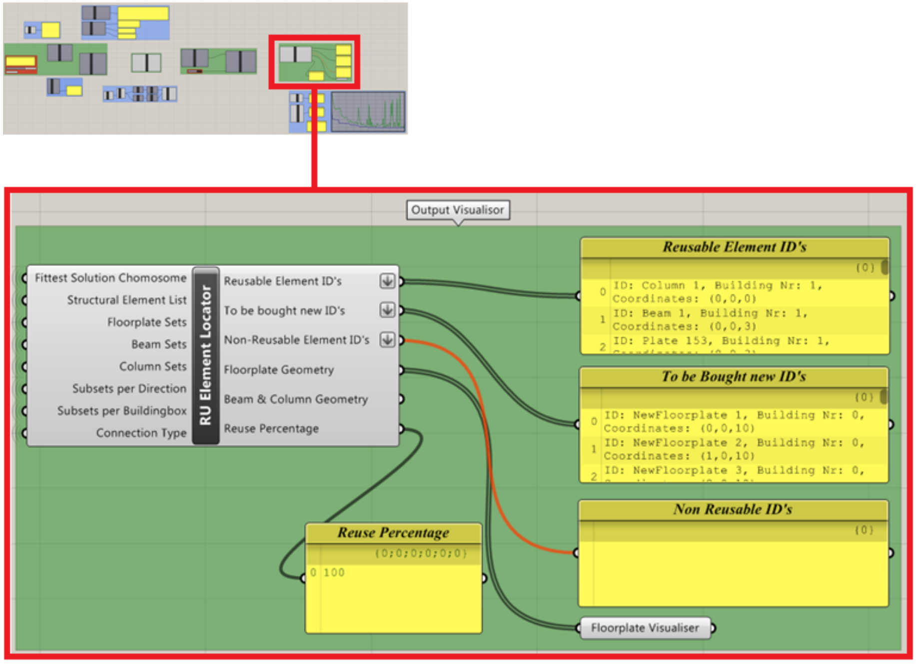
When the algorithm is running the output providing insight in the iteration process is a graph showing the development of the fitness score. This data is calculated and outputted in the 'RU_Algorithm' class.
The second output component is the building configuration, which is shown every time a new fittest solution has been found. The frame is created by the 'RU_Element' class, which imports the fittest solution from the 'RU_Algorithm' class and assigns the location of the elements by using the 'FloorplateMatcher', 'BeamMatcher' and 'ColumnMatcher' methods. These structural elements are then visualised to give a 3D impression of the structural frame, showing how the building develops towards a more optimal shape.

Final Building Output

After the stopping criteria of the algorithm have been met, the final element locations are set in the 'RU_Element' class, this data is again used to visualise the structural frame and output the Element ID's, locations and coordinates of the 'Reusable Elements' and 'Elements to be Bought New' for finishing the frame. From the 'Non Reusable Elements' only the ID's will be printed out, since they wont have a new location and coordinates.
The application also shows the reuse percentage of the available element set, giving the user insight to which extend the available set is used.

Grasshopper Output Components

In the grasshopper application the fittest solution chromosome is imported from the 'Inside Algorithm Executer' component, together with the element sets and subset grid inputs, the 'RU Element Locator' component decodes the chromosome into the frame. (Figure 4.18)



```
{0}
0 ID: Column 1, Building Nr: 1,
  Coordinates: (0,0,0)
1 ID: Beam 1, Building Nr: 1,
  Coordinates: (0,0,3)
2 ID: Plate 153, Building Nr: 1,
```

```
{0}
0 ID: NewFloorplate 1, Building Nr: 0,
  Coordinates: (0,0,10)
1 ID: NewFloorplate 2, Building Nr: 0,
  Coordinates: (1,0,10)
2 ID: NewFloorplate 3, Building Nr: 0,
```

```
{0}
```

```
{0;0;0;0;0;0}
0 100
```

Figure 4.18: Output components, visualising the frame and informing on reuse locations and percentages.

Element ID's and Locations

The elements inside this frame are then matched with the available elements inside the 'Structural Element List'. The elements in this list are assigned a location and coordinate set. If more elements are required to finish the frame, these elements are created and put into the 'To be Bought new ID's' list. And

if elements are not reused they are set in the 'Non Reusable ID's' list.
The elements set in the 'Reusable Element ID's' list and 'To be Bought new ID's' list also get their geometry visualised by the 'Floor plate Geometry' and 'Beam and Column Geometry' outputs.
The output component is completed by showing the 'Reuse Percentage' as mentioned earlier in the 'Final Building Component' section.

Optimisation Behaviour Graph

Besides the frame visualisation, there is also a graph showing the score development during the iteration process (Figure 4.19). The green line in Figure 4.19 shows the average fitness score in the generation and the blue line shows the fittest score inside each generation, the behaviour of these lines will be discussed in the next chapter.

Average Generation Scores

```
 98 1399
 99 7147
100 1008
101 1443
102 1827
```

Fittest Generation Scores

```
 98 835
 99 835
100 835
101 835
102 835
```

New Fittest Score Loop Numbers

```
 6 18
 7 21
 8 163
 9 185
10 252
11 268
```

Figure 4.19: Optimisation process graph, showing the fitness scores during looping on a logarithmic scale.

Besides the average and fittest score, also the loop numbers in which a new score was found is printed out to give insight on how fast the algorithm found the final solution. Looking at Figure 4.19 the fittest solution was found in loop number 268, while the final growth step of the blue line is almost halfway the graph and the algorithm ran until 100.268 loops were done. This is due to the logaritmic scaling of the graph. A logarithmic scale is chosen because the algorithm mostly jumps between fittest solutions during the early loops and less often during the later stages of the iteration process. [1] To give more insight in this early stage of the iteration process, a logaritmic scale was chosen.

---

[1] Why this happens is explained in Chapter 5: 'Application Validation'.

Output Validation

With the entire model translated from UML model into a grasshopper application, the next chapter will focus on the validation of the created application. Does it find the solutions required for the problem stated by the application user.

5 | Algorithm Validation

"However beautiful the strategy, you should occasionally look at the results."

- Winston Churchill -

The application implementation is followed by its validation. For algorithms the validation definition given by Cormen is: "An algorithm is said to be correct if, for every input instance, it halts with the correct output." [Cormen et al., 1990]
The validation will therefore start with a section on the characteristics of genetic algorithm behaviour.
The second section discusses how key components of the genetic algorithm are calibrated to increase application performance.
The final paragraph of this chapter discusses the scoring rule implementation, these are the rules determining the fitness of each solution and their configuration therefor influences how well each solution fits the set design problem. The section first provides insight in the influence on the solution types through test cases and than elaborates on why the freedom of setting the score configuration is left to the user.

5.1 Genetic Behaviour

This section elaborates on the inner workings of the reuse algorithm, sketching the genetic behaviour, followed by validating whether the reuse algorithm shows these behaviour characteristics.
This information is than used in the following sections by looking into how this behaviour can be manipulated to increase algorithm performance in the Section 5.2 and 5.3.

Fitness Landscape

The genetic behaviour will be explained based on a fictional representation of the fitness landscape (Figure 5.1). This landscape represents the entire solution space for a problem the algorithm needs to solve. Because similar solutions are assumed to have roughly the same fitness score, the fitness landscapes forms creates a rather mountainous landscape [Mitchell, 1999].

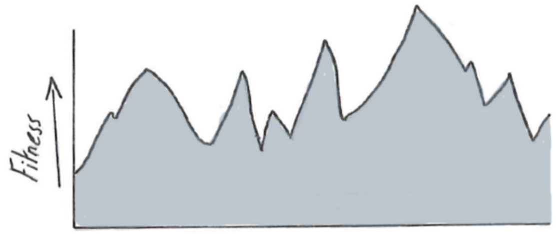


Figure 5.1: Fitnesslandscape, plotting the fitness of every possible solution in the search space.

The fitness landscape of Figure 5.1 is an qualitative approximation. The landscape differs for each problem the reuse algorithm encounters. However mapping this landscape is not the goal of the algorithm and this section, the focus is on how the algorithm moves along this landscape in search of the highest fitness peak.

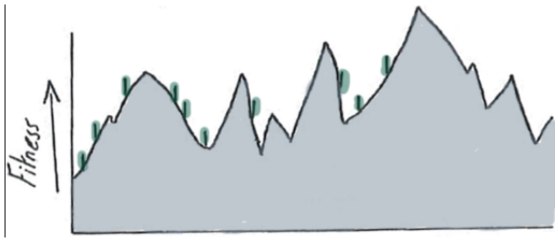


Figure 5.2: Each green spot represents a solution in the landscape.

Each iteration the algorithm works with a generation of solutions, the amount of solutions in a generation will be determined during calibration in Section 5.2.

Due to their random starting values the first generation solutions are scattered along the fitness landscape (Figure 5.2).

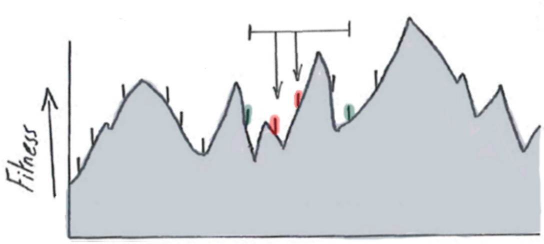


Figure 5.3: The green parent solutions, form the boundaries for the red solutions forming from the crossover method.

Selection

From each of these solutions the fitness value is determined and based on these values a weighted selection procedure is used to select the chromosomes for the next generation. The weight factor for each solution is its fitness score.

Crossover

The selected solutions will form pairs for the crossover method. During crossover the solutions (called parent solutions) exchange genetic material at a random gene number. The new solutions will therefor lie at a random location between the two parents (Figure 5.3). This process allows solutions to be found in the landscape between the outer solutions and because the fitter solutions have a higher chance to be selected, the solutions of the next generation are likely to have shifted towards the peaks of the solution landscape.

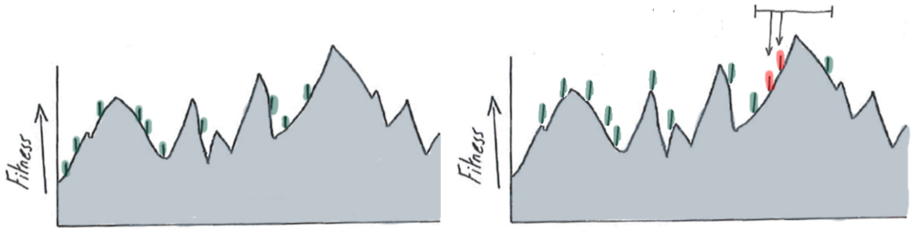


Figure 5.4: Both graphs above can lead to exclusion of the highest fitness solution.

Mutation

Due to the randomly chosen landscape location for the first generation and random location of the children solutions between their parent boundaries, it can occur that the fittest solution falls outside the generation scope and can not be found by means of crossover and selection (Figure 5.4). To ensure the fittest solution is in reach of the generation, the Mutation method is introduced.

Mutation will randomly change genetic material (gene(s)) in the solution chromosome to introduce new genetic material. This process can be visualised in the fitness landscape by setting a location randomly in the landscape, possibly including the fittest solution within the generation boundaries. (Figure 5.5)

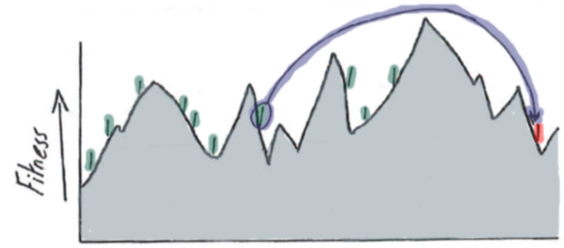


Figure 5.5: The circled solution is mutated, creating a random new solution somewhere on the landscape.

Elitsm

The last method involved in the search process is Elitsm, also described in the literature study Section 2.4. Elitsm can be visualised in the fitness landscape by always making sure the best scoring solution is selected (Figure 5.6).

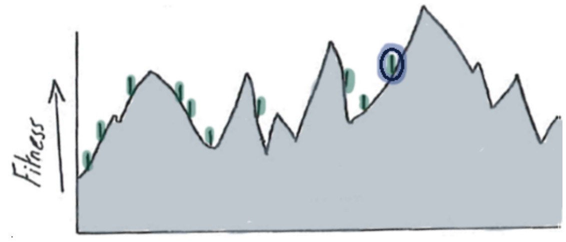


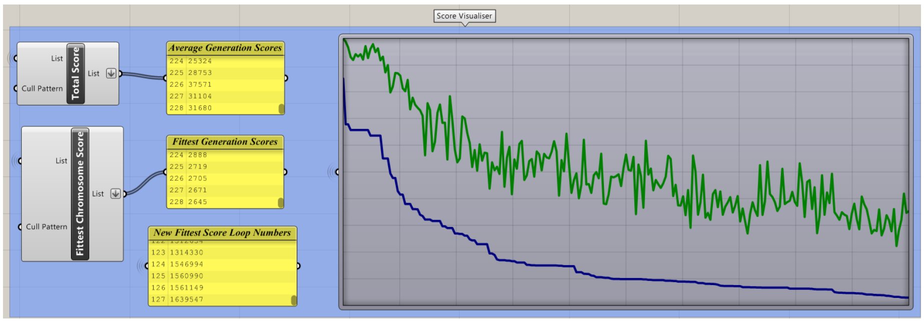
Figure 5.6: The solution inside the circle is the fittest solution and has a guaranteed selection.

By means of Selection, Crossover, Mutation and Elitsm, the generation moves through the landscape in search of the highest peak. The efficiency of the algorithm in finding the fittest solution will be discussed in Section 5.2. The next subsection will first discuss whether the reuse algorithm output shows the characteristics belonging to the search process described above.

Output Graph

The output graph shown in Figure 5.7 contains 2 datastreams, the green line represents the average fitness value of the generation and the blue line shows the fittest (lowest) value. The plotted iteration numbers are based on a logarithmic sequence, meaning the values from the first 10 iterations are plotted, than every 10th value until iteration number 100, every 100th value until iteration number 1000, etc. This is done because during the first iterations there will be a lot of

shifting between fittest solutions due to the amount of local peaks in generation range. Later on in the iteration process, fitter solutions become scarcer and the jump to a fitter solution can occur after an amount numerous amount of iterations, causing the visual quality of the early behaviour to diminish in case every value is plotted.
Besides this sequence also the values are plotted each iteration a new fittest solution is found, this is done to not lose any convergence information.



```
Average Generation Scores
224 25324
225 28753
226 37571
227 31104
228 31680
```

```
Fittest Generation Scores
224 2888
225 2719
226 2705
227 2671
228 2645
```

```
New Fittest Score Loop Numbers
123 1314330
124 1546994
125 1560990
126 1561149
127 1639547
```

Figure 5.7: Blue line represents the fittest score and the green line the average generation score.

Fittest Solution Behaviour

The fittest solution development (blue line Figure 5.7) represents the fittest value in the generation, due to the setup of the scoring system this means the fittest solution is always the lowest scoring solution. Due to Elitsm this value will either decrease if a new fittest solution is found or remain equal to the previous generation value in case no new fittest solution has been found. The value will never increase because the fittest solution was not chosen during the selection process, Elitsm makes sure it always gets picked at least once for the next generation.
The decreases in value are caused by either Crossover or Mutation finding a new fittest solution. Big isolated scoring jumps are often caused by Mutation planting a solution at a new higher peak in the fitness landscape and small jumps in clusters are often a signal of the solution 'Crossovering' towards the (local) peak in between the parent boundaries. However because it is a chance based process this is not necessarily the case, its also possible that multiple small jumps are caused by Mutation, however it is more likely this indicates Crossover behaviour.
For practice cases the fitness landscape is not known (if it was, there is no need to run the algorithm), therefore it cannot be said with 100% certainty that the fittest solution has been reached (unless the fitness score of 0 is reached) or that the found fittest score only occurs for that particular solution. These claims can only be made after the entire search space is investigated. However this search method is implemented to prevent the need of searching the entire search space. For this reason the stopping criteria mentioned in Section 4.3 are introduced.
Important to realise is that when the algorithm stops this does not directly mean that the fittest solution inside the search space is found. It is up to the

user to decide whether the found solution sufficiently meets the requirements.

Average Score Behaviour

The average score represented in by the green line in Figure 5.7 shows a less stable behaviour compared to the blue line. This is caused by Crossover and Mutation not necessarily resulting in fitter solutions (Figure 5.8), allowing the average generation score to increase, while this is not the case for the fittest solution. This freedom results in a greater score variation compared to the fittest solution.

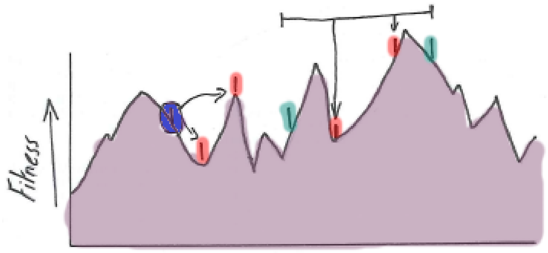
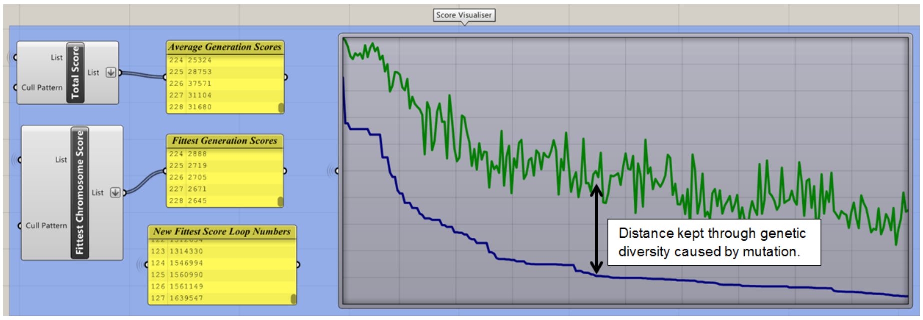


Figure 5.8: Mutation and Crossover are random operations so there is not control on finding only higher fitnesses.

During selection fitter solutions have a larger chance to get picked, causing the generation to converge towards (local) fitness peaks. When Mutation is excluded from the process this would mean the parent boundaries move closer together each generation, making the genetic material to become more and more alike and causing the algorithm to get stuck at a (local) peak. For this reason Mutation is used to introduce new genetic material, which is possibly better but can consist of worse scoring solutions, causing the average value to move away from the blue line.
The ideal combination between using Crossover to reach the (local) peak and introducing new material to keep diversity high enough to scout for other peaks, differs per configuration of the search space. If there is a relatively low amount of peaks a low Mutation rate is preferred to allow Crossover moving towards the peak. However when the landscape consists of a high amount of peaks the algorithm can get stuck at a local peak and requires Mutation to find new ones, so in this case a higher Mutation rate is advised.
Balancing out these parameters is done during calibration in Section 5.2, making sure there is enough Crossover behaviour while keeping enough genetic diversity. In the graph, keeping this genetic diversity is indicated by the distance between the two lines. (Figure 5.9)



Average Generation Scores
```
224 25324
225 28753
226 37571
227 31104
228 31680
```

Fittest Generation Scores
```
224 2888
225 2719
226 2705
227 2671
228 2645
```

New Fittest Score Loop Numbers
```
123 1314330
124 1546994
125 1560990
126 1561149
127 1639547
```

Distance kept through genetic diversity caused by mutation.

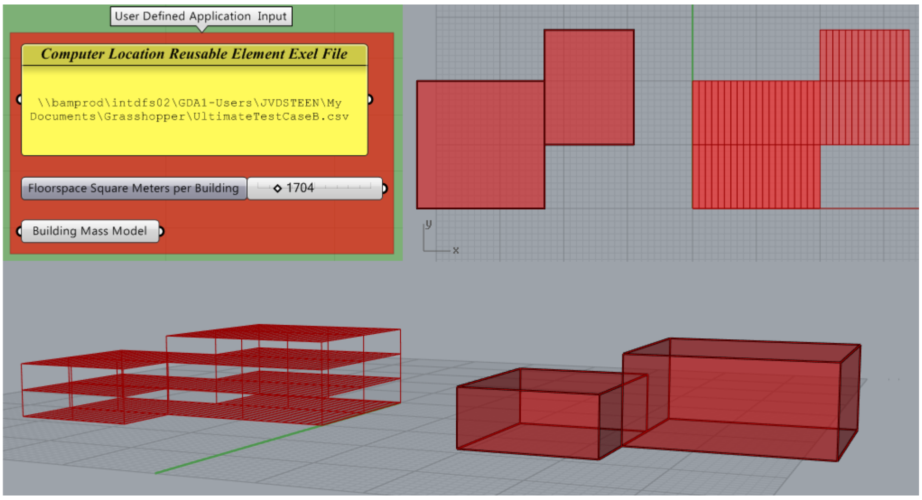
Figure 5.9: The distance between the fittest and the average score indicates genetic diversity, required to find new (fitter) solutions.

5.2 Mutation and Crossover Rates

This section elaborates on the application testing, checking if the algorithm finds the optimal solution for a given testcase. The algorithm performance in this testcase is used to calibrate the Mutation rate, Crossover rate and generation size. The section concludes by checking the performance on another testcase to see whether the calibration holds for multiple solution landscapes. [1]
Each combination of parameters was tested for 50 times and the average value results will be used to find the optimal combination.

Rate Testing

The parameter calibration will be done for the testcase shown in Figure 5.10. For this testcase the Excel document of available elements is setup in such a way that there is one known fittest solution. Meaning that during testing it was known when the algorithm converged to the optimal solution.



```
\\bamprod\intdfs02\GDA1-Users\JVDSTEEN\My
Documents\Grasshopper\UltimateTestCaseB.csv
```

Floorspace Square Meters per Building ◇ 1704

Figure 5.10: Casestudy to calibrate the algorithm, the available set has one perfect fit for the given shape.

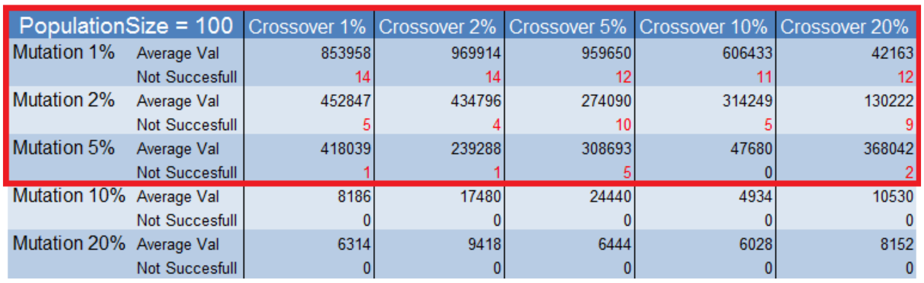
Mutation Rate Exclusion

The first test runs were done with a generation size of 100, reducing the change that a lack of genetic diversity occurs. While the generation size was set, the Crossover and Mutation rates were tested for the values presented in Figure 5.1.

The average value is the average amount of solutions tested before the optimal solution was found (example: 231 runs with an generation of 100 = 23.100 solutions were tested). Figure 5.1 shows that when the Mutation rate drops the

---

[1] Besides the testcases described in this section, exploratory research is done to different kinds of Mutation, these results did however not improve performance. Their specifications and results are posted in Appendix C.



PopulationSize = 100		Crossover 1%	Crossover 2%	Crossover 5%	Crossover 10%	Crossover 20%
Mutation 1%	Average Val	853958	969914	959650	606433	42163
	Not Succesfull	14	14	12	11	12
Mutation 2%	Average Val	452847	434796	274090	314249	130222
	Not Succesfull	5	4	10	5	9
Mutation 5%	Average Val	418039	239288	308693	47680	368042
	Not Succesfull	1	1	5	0	2
Mutation 10%	Average Val	8186	17480	24440	4934	10530
	Not Succesfull	0	0	0	0	0
Mutation 20%	Average Val	6314	9418	6444	6028	8152
	Not Succesfull	0	0	0	0	0

Table 5.1: The Mutation rates of 1%, 2% and 5% (inside red box) show higher average values and failed convergence attempts compared to the Mutation rates of 10% and 20 %.

average amount of solutions tested increases. Also the numbers in red indicate the number of times the algorithm fails to reach the optimal solution, which increased for low Mutation rates as well.
This can be explained by the low Mutation rate causing the algorithm to get stuck at an local optimum. Therefore the Mutation rates of 1%, 2% and 5% were excluded from further investigation.

Crossover Rate Determination

The Crossover rate is determined by checking the average amount of solutions required to find the optimal solution with the Mutation rate of 10% or 20% and a generation size of 24, 50 and 100. This gives six combinations and for each combination the Crossover rates 1%, 2%, 5%, 10% and 20% are tested. The average amount of tested solutions is shown in Figure 5.11.

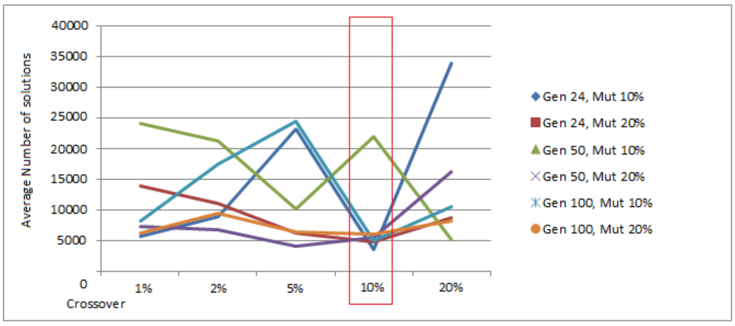


Figure 5.11: Crossover values most stable at a Crossover rate of 10%.

This graph shows that for a Crossover rate of 10%, the average value shows the most stable and fast behaviour of finding the fittest solution. With the exception of a generation of 50 solutions and a Mutation rate of 10%, which is caused by one really high value in this data set, excluding this value test will bring the average of the 50 runs also around 5000 solutions.
For this reason the Crossover rate is set at 10%.

Generation Size Influence

The size of the generation also influences the amount of test values the algorithm uses before finding the optimal solution. A small generation has a higher chance to consist of the same genetic data, which reduces algorithm performance. However when the generation becomes relatively large, a needless amount of chromosomes is tested, slowing down the algorithm.
Testing for different generation sizes is done for 10, 24, 50 and 100 solutions per iteration, while the Crossover rate is set at 10% and the Mutation can take on 10% and 20%.

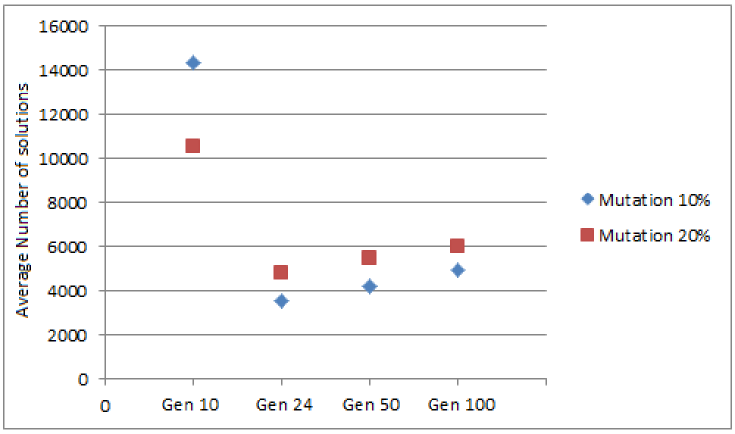


Figure 5.12: Performance increases until a generation size of 24 chromosomes.

The testing outcomes are presented in Figure 5.12, showing that the amount of solutions tested before the optimum is found decreases until a generation size of 24, when the generation becomes smaller, the amount of genetic diversity decreases too much and it takes longer to find the optimal solution.
Based on the graph from Figure 5.12 the choice is made to set the generation size to 24 and the Mutation rate to 10%.

Calibrated Rates

Concluding from the previous subsections, the algorithm is calibrated as followed:

- Mutation: 10%
- Crossover: 10%
- Generation Size: 24 solutions per iteration

To check whether this calibration works for different solution landscapes, the values are also tested on another testcase.

Validation changing Landscape

The new testcase is shown in Figure 5.13, for this testcase also the optimal solution is known and 50 testruns are performed to determine the average number of solutions tested before the optimal solution was found. This average value is 9219 tested solutions. The searchspace for this testcase is $2,66 * 10^{13}$, meaning on average $0,3 * 10^{-7}$ % of the seachspace is tested before the optimum is found.

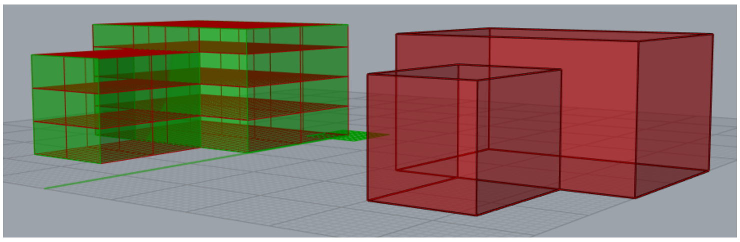


Figure 5.13: The user defined building mass is shown on the right and the calculated frame on the left.

For the calibration testcase shown in Figure 5.10, the average value was 3574 and the searchspace size $5,90 * 10^{9}$, leading to a percentage of $0,6 * 10^{-4}$ %. Meaning that for the testcase with the larger searchspace, the algorithm performed better when measured to the amount of possible solutions over solutions tested.
Indicating that when the searchspace increases, the algorithm increases in efficiency.

5.3 Scoring Rules

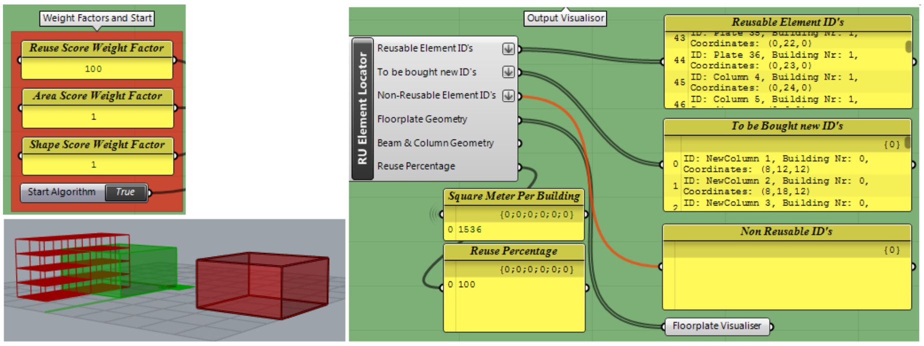
Every user can have a different opinion and for this reason the weight factors for each rule are user determined. How these weight factors affect the final solution is discussed in this section. To this end a testcase is created providing different results for different weight factor configurations, showing the influence each rule has on the structural frame.

Rule Influence Testcase

The scoring rule testcase is run for three different weight factor configurations. In this testcase half of the element set required to fill the top floor of the building mass is removed from the element set, so there is no solution where all three algorithm parameters can reach 0 points, causing the fittest solution to be based on the weight factor configuration.

Reuse Governing

Figure 5.14 shows the solution found when running the algorithm when the rule for reusing the available element set is governing. In this case the algorithm finds the fittest solution when all the elements are used (Non-Reusable Elements ID is empty), but the shape does not necessarily meet the building mass (Figure 5.14 building figures) and the functional square meters does not always meet the given square meter per building (1782 m^2 asked and 1538m^2 supplied in this case).



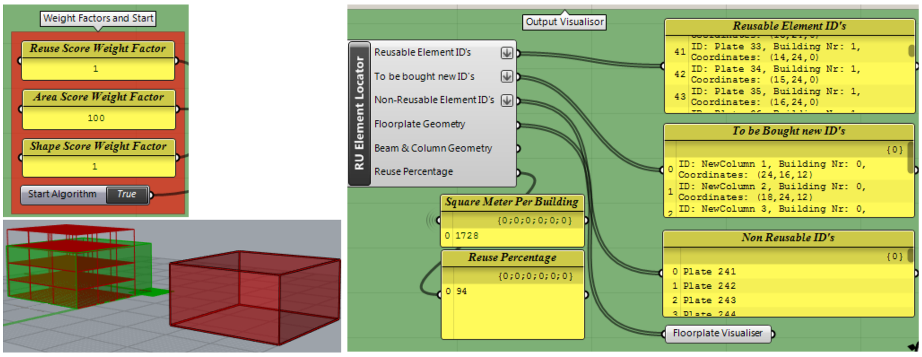
Reuse Score Weight Factor: 100
Area Score Weight Factor: 1
Shape Score Weight Factor: 1
Start Algorithm: True

```
Square Meter Per Building
{0;0;0;0;0;0}
0 1536

Reuse Percentage
{0;0;0;0;0;0}
0 100

Reusable Element ID's
43 ID: Plate 35, Building Nr: 1,
   Coordinates: (0,22,0)
44 ID: Plate 36, Building Nr: 1,
   Coordinates: (0,23,0)
45 ID: Column 4, Building Nr: 1,
   Coordinates: (0,24,0)
46 ID: Column 5, Building Nr: 1,

To be Bought new ID's
{0}
0 ID: NewColumn 1, Building Nr: 0,
  Coordinates: (8,12,12)
1 ID: NewColumn 2, Building Nr: 0,
  Coordinates: (8,18,12)
2 ID: NewColumn 3, Building Nr: 0,

Non Reusable ID's
{0}
```

Figure 5.14: Output when reusing the avialable set is governing.

Floorspace Governing

Figure 5.15 shows the solution found when running the algorithm when the rule for meeting the functional area is governing. In this case the algorithm finds the fittest solution as close as possible to the demanded square meter amount (1782m^2 asked and 1728m^2 supplied), but the shape does not necessarily meet the building mass (Figure 5.15 building figures) and not all the elements are reused (Non-Reuseable element list is filled).



```
41 ID: Plate 33, Building Nr: 1,
   Coordinates: (14,24,0)
42 ID: Plate 34, Building Nr: 1,
   Coordinates: (15,24,0)
43 ID: Plate 35, Building Nr: 1,
   Coordinates: (16,24,0)
```

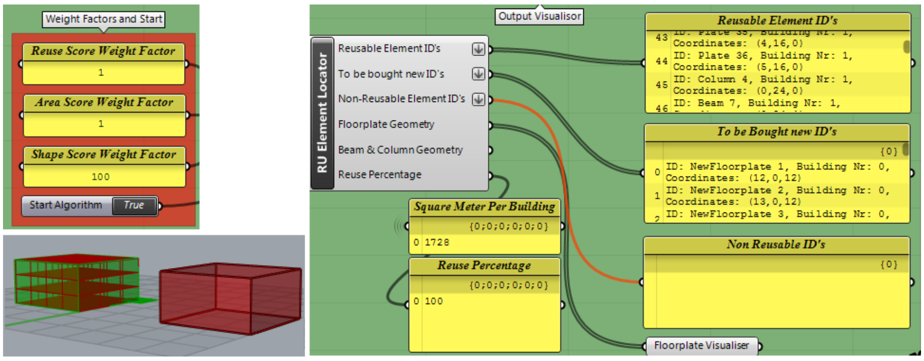
```
                                      {0}
0 ID: NewColumn 1, Building Nr: 0,
  Coordinates: (24,16,12)
1 ID: NewColumn 2, Building Nr: 0,
  Coordinates: (18,24,12)
2 ID: NewColumn 3, Building Nr: 0,
```

```
                                      {0}
0 Plate 241
1 Plate 242
2 Plate 243
```

Figure 5.15: Output when meeting the asked floorspace requirements are governing.

Shape Governing

Figure 5.16 shows the solution when the rule for meeting the building mass shape is governing. In this case the algorithm finds the fittest solution as close as possible to the provided shape, paying less attention to reusing the available elements and approaching the required amount of square meters.
Making the shape rule weight factor governing does provide solutions often close to the optimal solution. Caused by the building mass being scaled to fit the other two rules. Therefore it is preferable to give the shape score an higher weight factor compared to the other two rules.



```
43 Coordinates: (4,16,0)
44 ID: Plate 36, Building Nr: 1,
   Coordinates: (5,16,0)
45 ID: Column 4, Building Nr: 1,
   Coordinates: (0,24,0)
46 ID: Beam 7, Building Nr: 1,
```

```
                                          {0}
0 ID: NewFloorplate 1, Building Nr: 0,
  Coordinates: (12,0,12)
1 ID: NewFloorplate 2, Building Nr: 0,
  Coordinates: (13,0,12)
2 ID: NewFloorplate 3, Building Nr: 0,
```

Figure 5.16: Output when meeting the asked floorspace requirements are governing.

Stadium Testcase

This chapter confirmed that the scoring process shows the characteristics of a genetic algorithm with elitism. Furthermore the algorithm parameters, being Mutation, Crossover and the generation size, have been calibrated to improve algorithm performance. And finally the influence of the scoring rules is checked to see how the weight factors influence the results.
With the algorithm up and running the final chapter will present the stadium reuse project, for which the application was designed.

6 | Al Wakrah Stadium Reuse

"A good idea is 10 percent; implementation, hard work and luck are 90 percent."

- Guy Kawasaki -

The last chapter showcases the implementation of the reuse algorithm on the Al Wakrah (Qatar) stadium project. The first section will provide a case description and includes a look into the stadium (de)construction process. The second section elaborates on the implementation of the showcase data into the application and the results following from the algorithm. The final section discusses further steps required before the reuse project can be implemented in practice.

6.1 Project Specifications

The calibrated reuse application is now ready to be applied on a project from practice. This first section will cover the case facts, setting up the application input.

Case Facts

The showcase project is the Al Wakrah soccer stadium in Qatar, which will be constructed for the 2022 World Cup in Qatar (Figure 6.1).

Figure 6.1: Overview of the Al Wakrah stadium (Zaha Hadid Website).

After the world cup the stadium will be used by Al-Wakrah Sport Club. Due to the fanbase size of this club, the second ring of this stadium is to be removed after the 2022 FIFA World Cup to make sure the Montreal scenario from the introduction does not occur. (Figure 6.2)

Figure 6.2: Total stadium during World Cup (left), removal of 2nd ring (middle) and stadium in final configuration (right).

The plans for the soccer stadium are still in a preliminary phase, meaning the (de)construction strategy is yet to be determined and no construction drawings for the stadium are available. For this reason the a construction design is made by the author.
The construction design is based on the second ring of the Amsterdam Arena which, based on the Al Wakrah renders, both have the same amount of rows in the second ring (about 24 rows). Leading to the 2nd ring design shown in Figure 6.3.

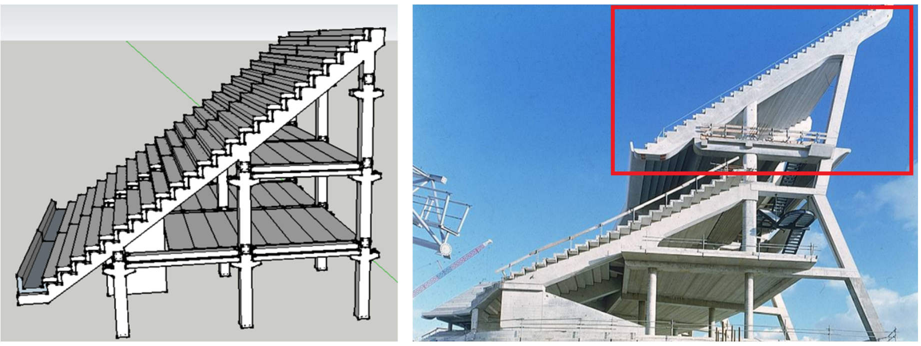


Figure 6.3: Applied structural configuration for Al Wakrah (left) based on 2nd ring Amsterdam Arena (right, red square).

(De)Construction Process

To implement the reuse strategy, the building requires a deconstruction strategy. The Al Wakrah strategy will be one suited for concrete elements, because an important aspect argued by Klomp is that the mass from concrete elements is required in stadium stand elements due to their the damping effect [Klomp, 2013]. This introduces a conflict of requirements. For transportation, the weight is governing and should be minimalised, however for vibrations the use of concrete elements is required to damp the vibrations. Futhermore use of composite systems is discouraged because these elements can often not be retracted from the building without some form of demolition.
This leads to the choice to use prefrabicated concrete elements, suited for container transport.

DEMU Connections

Section 2.5 concluded that the deconstruct-ability of a structure is mainly influenced by its connections.
For this reason the Al Wakrah stadium will make use of the DEMU system (Figure 6.4). This system ensures that during deconstruction, the elements can be retracted from the stadium without the use of demolition equipment. Ensuring a faster, lower cost/energy using deconstruction process compared to retrieving in-situ elements from a building.

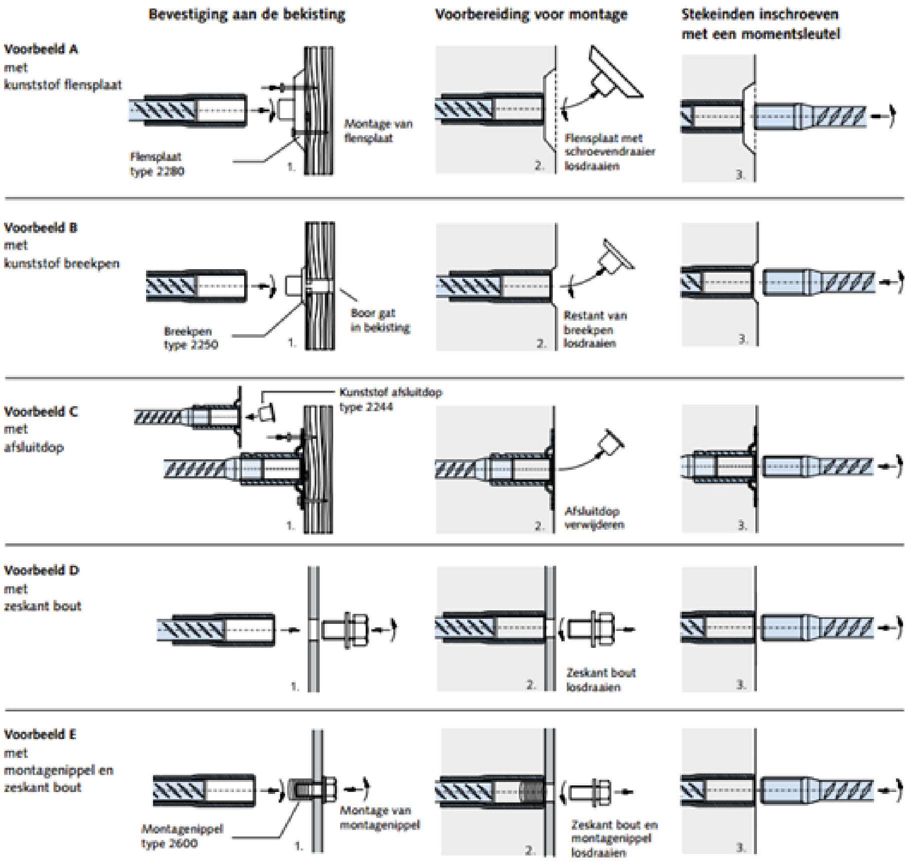


Figure 6.4: Bolted reusable connections in concrete elements by use of the DEMU system [Halfen, 2013].

(De)Construction Procedure

Using the DEMU system the (de)construction steps will be shown for three connection systems. Beam to Column, Column to Column and Stand Element to Raking Beam.

- Beam - Column

  Figure 6.5 (top left) shows bolts, placed inside a PVC tube, which are screwed into the cast in DEMU holes of the columns. The bolts are used to guide the beam to its place (Figure 6.5, top right). When the beam is located accordingly, the remaining cavity between PVC tube and concrete element is filled with grout (Figure 6.5, bottom left), after hardening of the grout the beam is connected to the column.
  During deconstruction the bolts are removed from within the PVC tube and the element can be moved from the building (Figure 6.5, bottom right). Before the element is reused, the PVC tube needs to be drilled out of the concrete to ensure enough space to cope with measurement errors in the reuse structure.

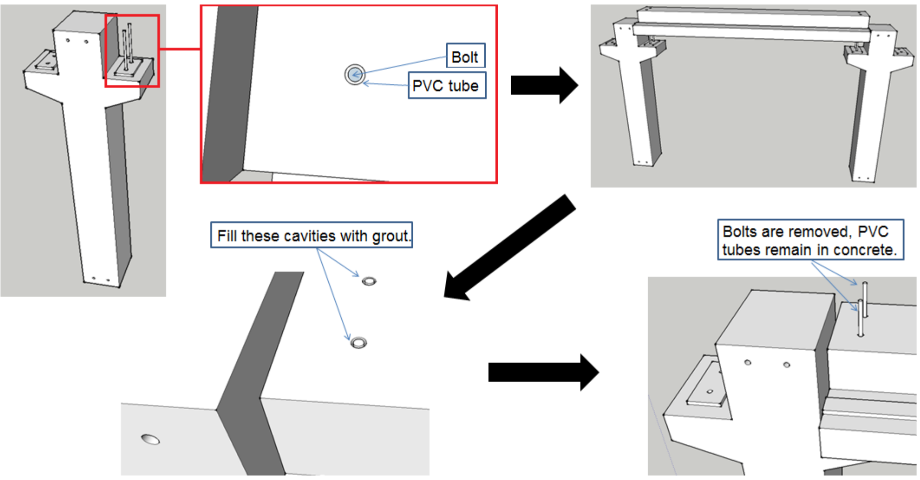


Figure 6.5: Four steps for connecting and disconnected a beam to a column using the DEMU system.

- Column - Column

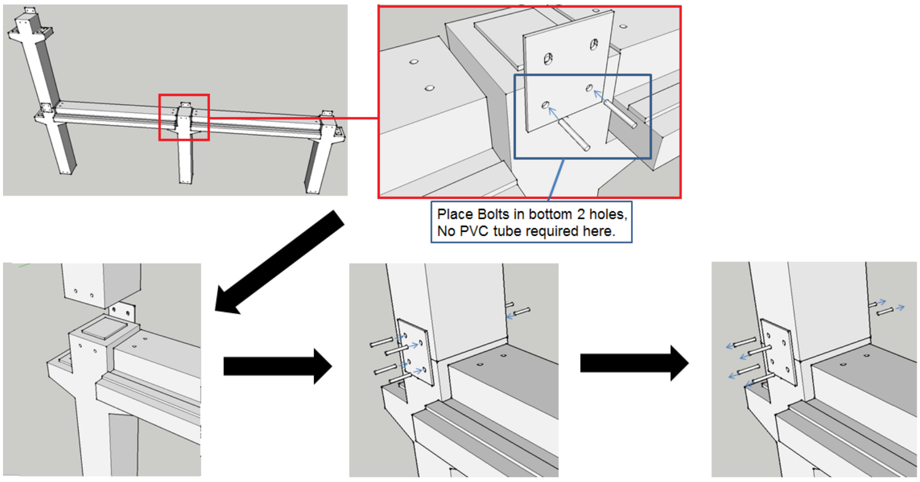


Figure 6.6: Four steps for connecting and disconnecting a column to a column using the DEMU system.

The DEMU connection slots are cast into the concrete columns. During connection, steel connection plates are bolted to the column (Figure 6.6, top middle), next step is hoisting in the column (Figure 6.6, bottom left). To deal with measurement differences cause by manufacturing, a rubber filling plate is placed on top of each column.
This step is followed by the placement of the connection plate on the other side and the remaining bolts (Figure 6.6, bottom middle). The measurement differences in this step are taken into account by using longer grooves to place the bolts.

During deconstruction the bolts and connection plates are removed and the columns can be removed. (Figure 6.6, bottom right)
For connecting the raking beam to the column the same procedure is used.

- Stand Element - Raking Beam

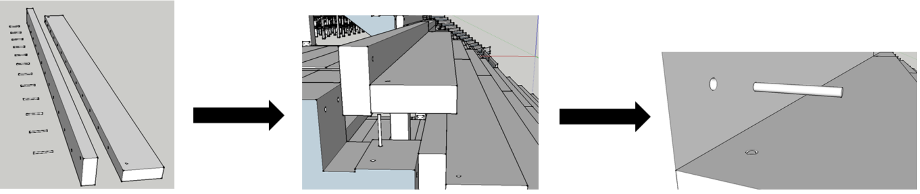


Figure 6.7: Assembling the stand element, placing it and fastening it, deconstruction happens in the reverse order.

Figure 6.7 (left) shows that first the stand element is assembled on the ground, after assembly it is hosted in and grouted following a similar procedure as connecting the beam to the column, except now also a bolt in the vertical direction needs to be applied. (Figure 6.7, middle and right)
During deconstruction the reverse order is applied to regain the vertical and horizontal part of the stand elements to be reused as floorplates in a new building.

Repetition Measures

Besides increasing the deconstruction speed through the connections, also repetition influences the speed and reusability of the elements according to Section 2.5.
A building feature increasing repetition are the indentations in the raking beams. These indentations are required due to the changing row height due to sight line requirements. To make sure the stadium has C-values meeting FIFA standards, while also ensuring repetition, the raking beam has indentations which lower the vertical stand elements when vertical step size reduces when getting closer to the pitch. (Figure 6.8)

Another measure to increase repetition is that the stand elements, when deconstructed into floor plate elements, have the same width as the other floorplates used inside the stadium (Figure 6.9). This ensures larger element sets, increasing the usability of these sets.

The final repetition feature is located in the corner sections of the stadium. Due to the cornering the elements have angled sides at one element side. For this reason the elements need to be sewn straight. Multiple rows are sewn to the same length to increase the element set size and thus increase reusability. (Figure 6.10)

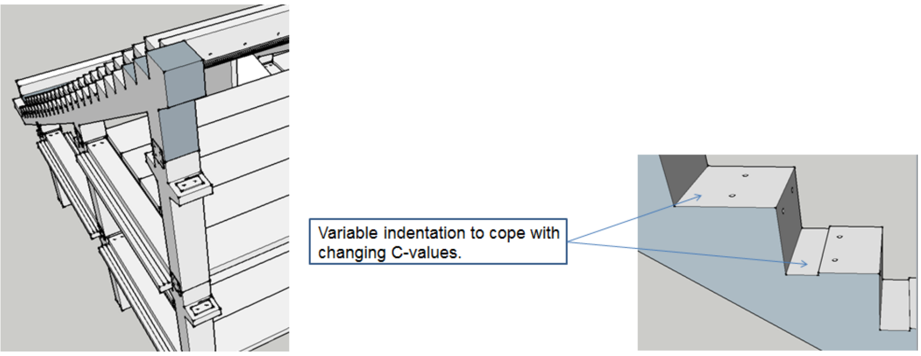


Figure 6.8: Due to the required curvature of the raking beams, indentations are made to ensure the vertical stand element can stay equal in height.

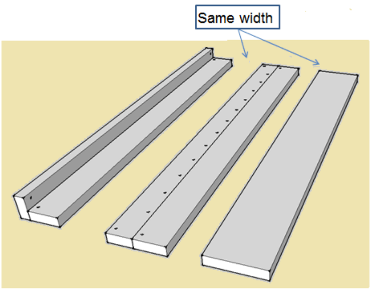


Figure 6.9: The sum of the horizontal and vertical section of the floorplates equal the width of a floorplate.

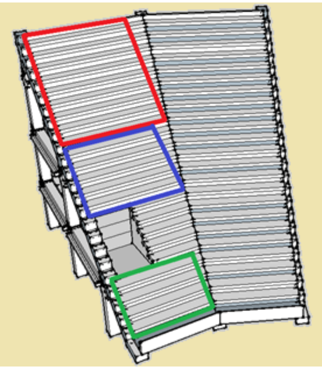


Figure 6.10: Multiple rows are sewn to the same length to increase repetition.

Reusable Elements

Based on the cross section presented in Figure 6.3 the total second ring consists of the elements shown in Figure 6.11. From this set the raking beams, U-shaped bottom elements, ramps to enter the stands and stabilising walls are marked as non reusable elements, because of their custom shape.

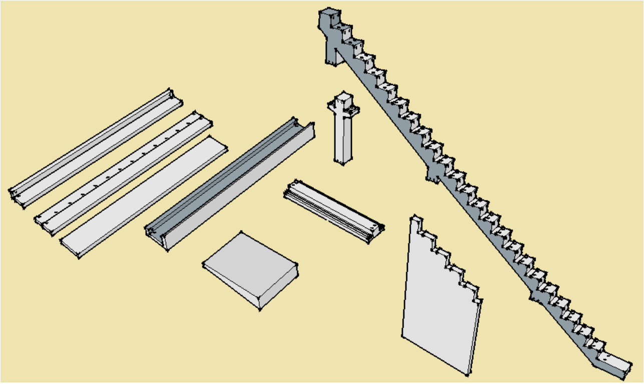


Figure 6.11: Element types used in the second ring of the stadium.

This means that the reusable element database consists of the element types shown in Figure 6.12. Included in this set are the stand elements, which typically will be stair-shaped elements comprising about 70-80% of the stadium elements. To be able to include these elements in the reusable set, a deconstructable concept is created as shown on the right in Figure 6.12.

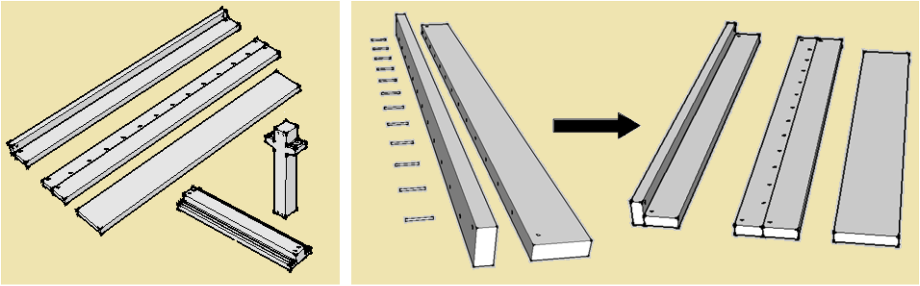
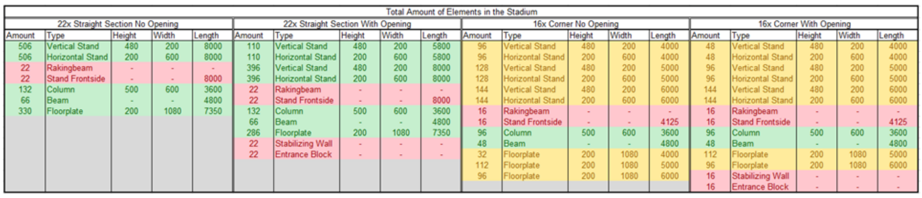


Figure 6.12: Reusable element types (left), where the stand elements are deconstructed into floorplates as shown on the right.

Reusing the elements from Figure 6.12, the element database from Figure 6.13 is compiled. This database will serve as input for the application.



Total Amount of Elements in the Stadium																			
22x Straight Section No Opening					22x Straight Section With Opening					16x Corner No Opening					16x Corner With Opening				
Amount	Type	Height	Width	Length	Amount	Type	Height	Width	Length	Amount	Type	Height	Width	Length	Amount	Type	Height	Width	Length
506	Vertical Stand	480	200	8000	110	Vertical Stand	480	200	5800	96	Vertical Stand	480	200	4000	48	Vertical Stand	480	200	4000
506	Horizontal Stand	200	600	8000	110	Horizontal Stand	200	600	5800	96	Horizontal Stand	200	600	4000	48	Horizontal Stand	200	600	4000
22	Rakingbeam	-	-	-	396	Vertical Stand	480	200	8000	128	Vertical Stand	480	200	5000	96	Vertical Stand	480	200	5000
22	Stand Frontside	-	-	8000	396	Horizontal Stand	200	600	8000	128	Horizontal Stand	200	600	5000	96	Horizontal Stand	200	600	5000
132	Column	500	600	3600	22	Rakingbeam	-	-	-	144	Vertical Stand	480	200	6000	144	Vertical Stand	480	200	6000
66	Beam	-	-	4800	22	Stand Frontside	-	-	8000	144	Horizontal Stand	200	600	6000	144	Horizontal Stand	200	600	6000
330	Floorplate	200	1080	7350	132	Column	500	600	3600	16	Rakingbeam	-	-	-	16	Rakingbeam	-	-	-
					66	Beam	-	-	4800	16	Stand Frontside	-	-	4125	16	Stand Frontside	-	-	4125
					286	Floorplate	200	1080	7350	96	Column	500	600	3600	96	Column	500	600	3600
					22	Stabilizing Wall	-	-	-	48	Beam	-	-	4800	48	Beam	-	-	4800
					22	Entrance Block	-	-	-	32	Floorplate	200	1080	4000	112	Floorplate	200	1080	5000
										112	Floorplate	200	1080	5000	96	Floorplate	200	1080	6000
										96	Floorplate	200	1080	6000	16	Stabilizing Wall	-	-	-
															16	Entrance Block	-	-	-

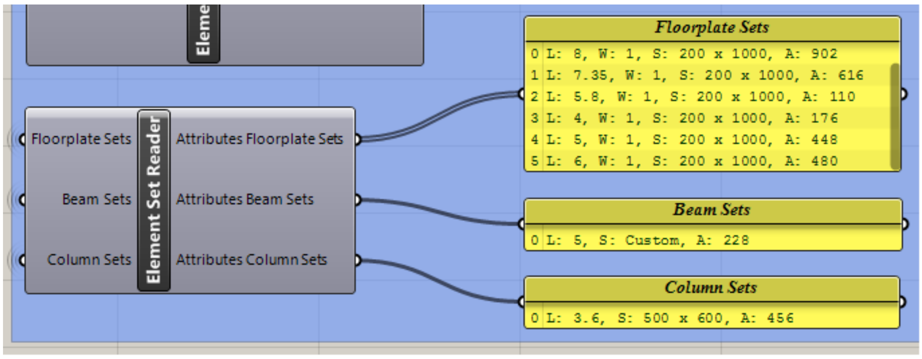
Figure 6.13: Green elements can be directly reused, orange elements can be reused after decomposition as shown in Figure 6.12 and red elements will not be reused.

6.2 Reuse Results

With the input data set, the algorithm is put to work in this section. The section starts by presenting the model input, followed by an oversight of the output data, showing the reuse structures build up from the stadium elements.

Application Input

The element input consists of the list of elements shown in Figure 6.13, when imported into the model these elements are divided into sets as shown in Figure 6.14.



Floorplate Sets

```
0 L: 8, W: 1, S: 200 x 1000, A: 902
1 L: 7.35, W: 1, S: 200 x 1000, A: 616
2 L: 5.8, W: 1, S: 200 x 1000, A: 110
3 L: 4, W: 1, S: 200 x 1000, A: 176
4 L: 5, W: 1, S: 200 x 1000, A: 448
5 L: 6, W: 1, S: 200 x 1000, A: 480
```

Beam Sets

```
0 L: 5, S: Custom, A: 228
```

Column Sets

```
0 L: 3.6, S: 500 x 600, A: 456
```

Figure 6.14: The element sets used for the township project, there are 7 types of floorplates and 1 type of column and beam available.

The reuse project is a housing project in the townships of South Africa. The building mass study performed by the architect resulted in the form presented in Figure 6.15. The project LoR resulted in a floorspace demand of 4352 m^2.

Calculated from the reusable floor plate square meters and the required floor space per building, a total of four buildings will be produced from the stadium elements. (Figure 6.16)

Finally the weight factors are set as presented in Figure 6.17.

Reuse Output

Running the algorithm on the current application setup, 4 structural frames are created (Figure 6.18). With an overall reuse percentage of 100% and with each building approaching the required building floor area as shown in Figure 6.19. Besides all of the structural elements being reused, also 1556 new elements need to be acquired to create the structural frames, which is equal to 30% of the reusable element amount.

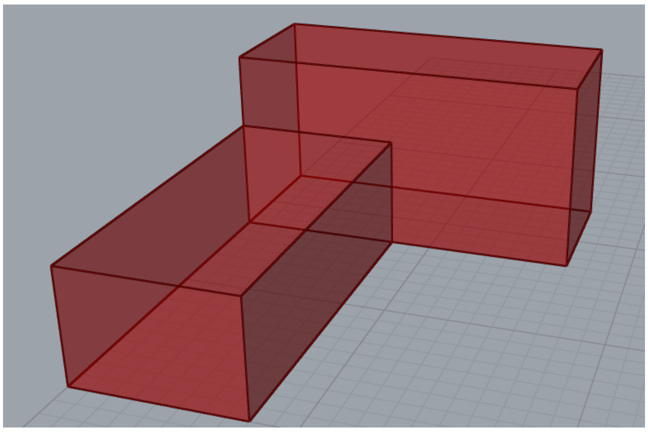


Figure 6.15: The form for the township structures, consists of two rectangular boxes with equal floorplans and different heigths.

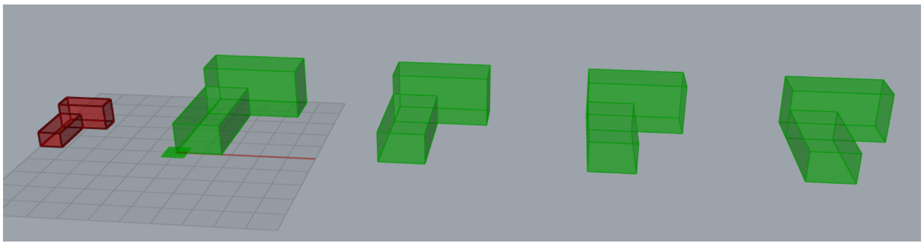
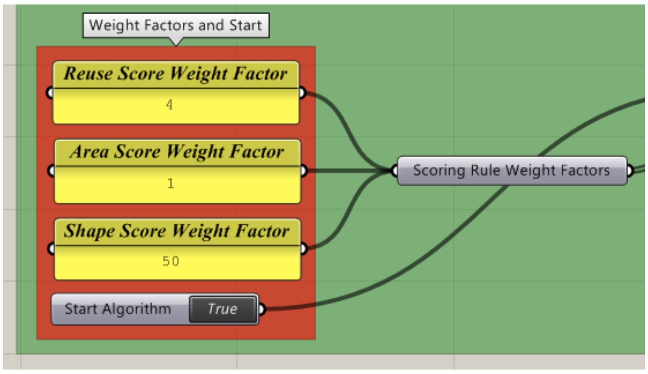


Figure 6.16: The available plates and required floorspace per building indicated that 4 reuse buildings is feasible.



Weight Factors and Start

Reuse Score Weight Factor: 4

Area Score Weight Factor: 1

Shape Score Weight Factor: 50

Start Algorithm: True

Figure 6.17: The weight factors used for the township project.

Reducing Amount of Reuse Buildings

The amount of elements required to finish the design, raised suspicion that it might be beneficial to create only three buildings. Since this removes around 25% of the elements required while 30% of used elements were to be bought new.

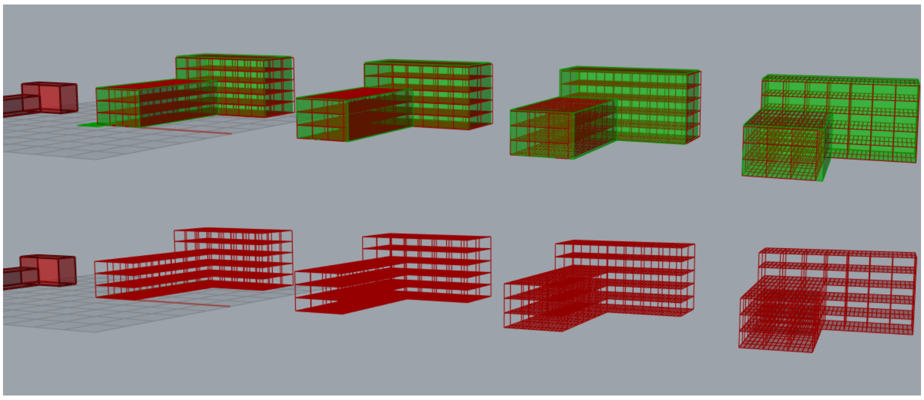
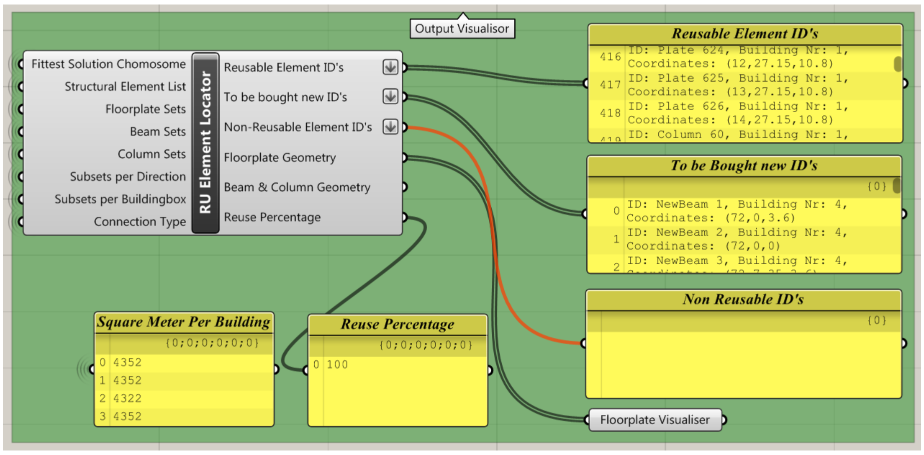


Figure 6.18: The constructed frames, shown inside the asked shape (top) and without the shape (bottom).



Reusable Element ID's

```
416 ID: Plate 624, Building Nr: 1,
    Coordinates: (12,27.15,10.8)
417 ID: Plate 625, Building Nr: 1,
    Coordinates: (13,27.15,10.8)
418 ID: Plate 626, Building Nr: 1,
    Coordinates: (14,27.15,10.8)
    ID: Column 60, Building Nr: 1,
```

To be Bought new ID's

```
                                   {0}
0 ID: NewBeam 1, Building Nr: 4,
  Coordinates: (72,0,3.6)
1 ID: NewBeam 2, Building Nr: 4,
  Coordinates: (72,0,0)
2 ID: NewBeam 3, Building Nr: 4,
```

Non Reusable ID's

```
                                   {0}
```

Square Meter Per Building

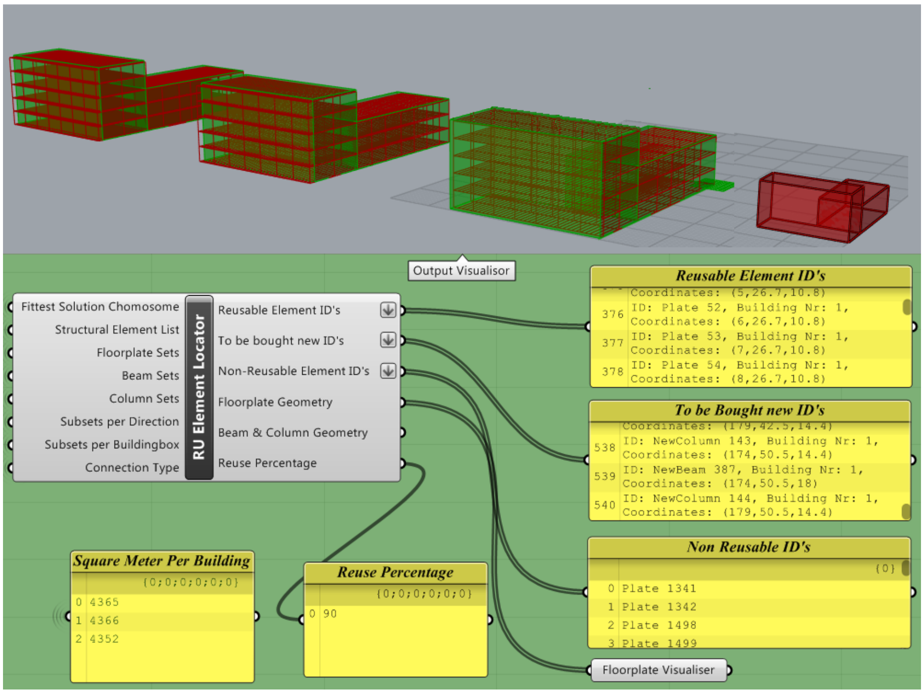
```
  {0;0;0;0;0;0}
0 4352
1 4352
2 4322
3 4352
```

Reuse Percentage

```
  {0;0;0;0;0;0}
0 100
```

Figure 6.19: The ouput data showing the location of every element, the square meter per building and the reuse percentage.

The outcome of this case is shown in Figure 6.20. The results indicate the reuse percentage has dropped with 10% compared to the four buildings case. Looking into the Non-Reuseable element list, it shows that 332 elements are not reused. With all the Non-Reusable elements being floor plates.
The 'To be bought new ID' list contains 541 elements, consisting of columns and floor beams. The decision to choose for the four building option or the three building option depends on user preferences. If the user wishes a high as possible reuse percentage, rounding the number of buildings upward is prefered (3 building option in this case). Else if the user wants to minimise the investment required to finish the design, the user want to reduce the amount of elements to be bought new and goes for the 3 building option.



Reusable Element ID's

```
    Coordinates: (5,26.7,10.8)
376 ID: Plate 52, Building Nr: 1,
    Coordinates: (6,26.7,10.8)
377 ID: Plate 53, Building Nr: 1,
    Coordinates: (7,26.7,10.8)
378 ID: Plate 54, Building Nr: 1,
    Coordinates: (8,26.7,10.8)
```

To be Bought new ID's

```
    Coordinates: (179,42.5,14.4)
538 ID: NewColumn 143, Building Nr: 1,
    Coordinates: (174,50.5,14.4)
539 ID: NewBeam 387, Building Nr: 1,
    Coordinates: (174,50.5,18)
540 ID: NewColumn 144, Building Nr: 1,
    Coordinates: (179,50.5,14.4)
```

Non Reusable ID's

```
                {0}
0 Plate 1341
1 Plate 1342
2 Plate 1498
3 Plate 1499
```

Square Meter Per Building

```
  {0;0;0;0;0;0}
0 4365
1 4366
2 4352
```

Reuse Percentage

```
  {0;0;0;0;0;0}
0 90
```

Figure 6.20: Output data when allowing only three reuse buildings, showing only plates in the 'Non Reusable ID's' list, while the 'To be Bought new ID's' list contains only beams and columns.

6.3 Reuse Project Construction

The final section discusses the follow up steps, required evolve this model from a wire frame into an build-able structure.

Structural Frame Development

The structural frame will be build according to the locations assigned to the elements in the reuse application (Figure 6.21).

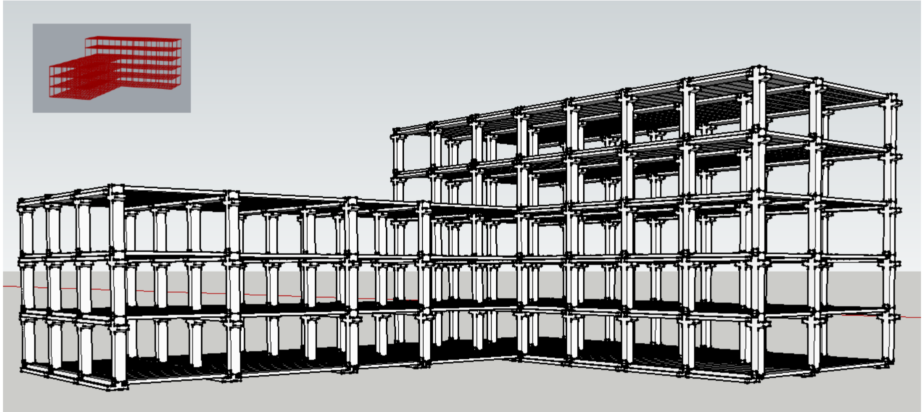


Figure 6.21: Transformation of the grasshopper wire frame to a frame showing the structural elements.

During this design phase the functions will be located into the building, accompanied by the required spacings for vertical transport and installations (Figure 6.22).

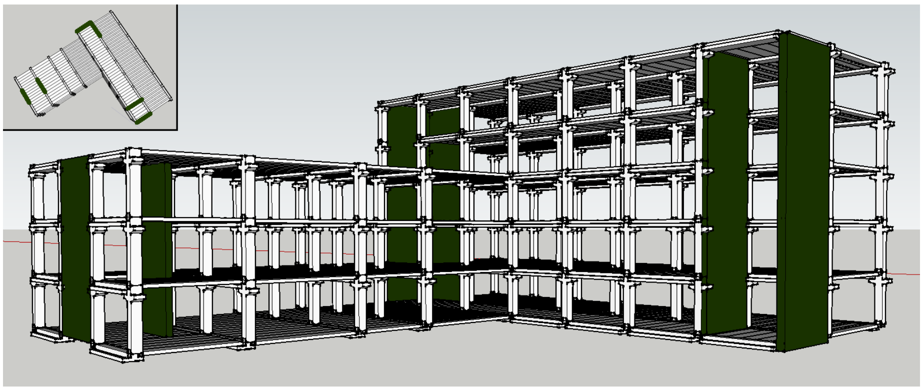


Figure 6.22: Creating spaces for vertical transport and stabilising (green) elements.

The placement of these locations influence the elements at these locations, for instance elements located at a vertical transport zone will be reduced in length or removed in total. Leading to a reduction of the reuse percentage given in the application. The size of this reduction is dependent on the choices made by the architect and building codes.

Frame with stability Elements

After the functions (interior walls) and installations are located the building the loads on the structure are known, indicating smart locations for the stability elements(Figure 6.22). After placement of the stability elements the building will be checked according to the building codes from the reuse location. (Hand Calculations are placed in Appendix D).

Frame with Finishing Elements

Finally the facades are placed, finishing up the building so it can be delivered to the client. (Figure 6.23)

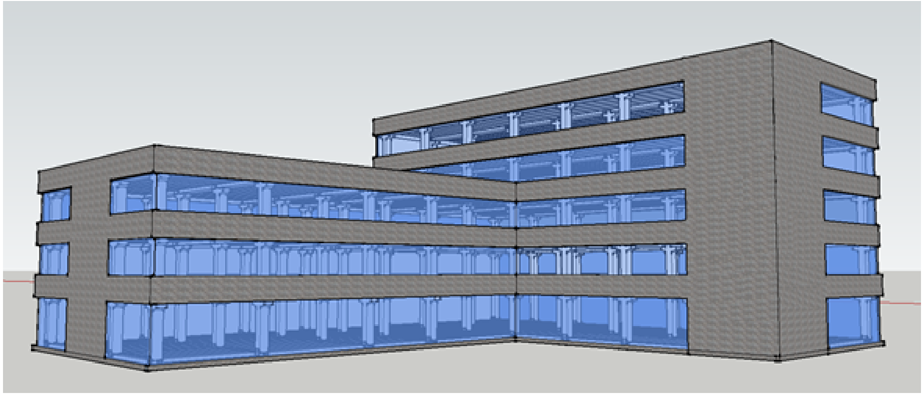


Figure 6.23: Placing the facade so that the building can be delivered to the client.

7 | Discussion

Through the use of parametric and associative design the research focused on the development of a generic reuse application, designed to find the ideal configuration of the structural frame constructed of reused elements. The ideal configuration depended on three parameters:

1. Reusing the available element set
2. Meeting the floorspace requirements
3. Following the architectural building shape

This chapter reflects on the these research objectives and discusses the application advantages and limitations.

7.1 Objective Reflection

To find the ideal frame configuration, a subset configuration method is designed. These subsets serve as building blocks consisting of element types available in the reusable set. The ideal configuration of the subset grid is based on the provided building mass and required floorspace. This configuration is pursued by the implementation of a genetic algorithm.
Using the subset method, the designed frame in the stadium case showed a 100% reuse of the available element set, while approaching the floorspace requirement with a 1% deviation and approaching the set building shape with a maximum deviation of 4%.
However the application output does not hold all the information required for a definitive design of a structural frame. For instance the structural capacity of the structure is not calculated. When more design knowledge and parameters can be translated into algorithm rules, the model results will increase in usefulness.
The end product can therefor not be one to one translated into practice. Requiring more work to be done before the structure can be realised.

7.2 Application Advantages

- Configuration Speed
  In the introduction (Chapter 1) a problem identified in design for reuse

is the time consuming process of finding the structural configuration for the reuse frame due to the strained design process. Therefore requiring intensive collaboration between the architect and engineer.
By use of the reuse application this process is automated, finding the desired frame by use of a genetic algorithm, which scouts the the solution space in a more efficient way that mere trial and error and it does so without requiring the manhours. Giving the design team the chance to focus on implementing more design intelligence.

- Flexibility
  Due to the generic setup of the application and use of Parametric and Associative design, the application is useable for multiple reuse projects. And the user is able to steer the solution by changing the user input, while getting nearly immediate feedback on the outcomes.

- Process Transparency
  Because the application gives immediate feedback on the results, the user gains understanding of the impact of design choices earlier in the process. Allowing the implementation of more design knowledge which can lead to a higher quality design.

- Result Usability
  The results include the coordinates of each element, allowing the immediate allocation of the elements into the new project based on the frame output. Due to the high detail of element allocation, more accurate estimations for costs and environmental impact can be made earlier in the design process.

7.3 Application Limitations

- Strategy based on 'Slimbouwen' and 'Openbouwen'
  These strategies divide the design process in clear phases, which is used to define structural elements as reusable parts of a building. However some interaction exists between the proposed levels, for instance the influence of the building interior (vertical transportation) on the structural frame. This interaction is not taken into account in the application, potentially reducing the usability of the application results.

- Finishing Steps
  The frame output from the application is not a complete definitive design, information on the vertical transport areas and global stability elements is required. The application also does not include connection designs or checks the structural capacity of the elements.
  These aspects need to be determined before the construction of the reuse project can start.

- Non-Structural Element Reuse
  The strategy focuses on the reuse of structural elements, the non-structural elements are not assessed. Lost investments and environmental impact of these elements are set out of scope.
  However when a structure is deconstructed based on different incentives, for instance in the stadium case, also the non-structural elements might not have served their total economical and or technical lifetime. Finding a reuse function for these elements is not included in the strategy.

- Loss percentages
  In all estimations and reuse calculations, losing elements due to damaging the elements during (de)construction and transport is not taking into account.
  When these loss percentages increase the feasibility of the reuse strategy reduces, which could influence the decision to whether or not reuse should be implemented.

- Logistic process
  Efficiency in the (de)construction process improves when elements, which are retracted from one structure, can immediately be applied in the reuse project.
  This feature is not explicitly applied in the application. The order in which elements are selected for the reuse project is currently dependent on the order in which these elements are placed in the structural element list.

- Complex Shapes
  The subset configuration allows the user to specify rectangular buildings, however in practice there is also demand for angled and curved facades. The current state of the application does not allow the implementation of these more complex shapes.

- Structural Calculations
  The designed frame does not calculate the structural safety of the frame during the iteration process. Therefor it is possible that the frame output can be structurally unfeasible or has a large amount of overcapacity.

8 | Conclusions

The thesis focused on creating a generic reuse application, which creates a structural frame for a reuse project. The aim for this structural frame is to utilise the available element set, while meeting the functional floorspace requirements of the new building and respecting the defined building mass shape.
This chapter presents the conclusions which can be drawn based on the research performed to reach this goal.

- Reuse Incentives
  The literature study concluded that environmental incentives for pursuing reuse are caused by a higher grade of element lifetime utilisation and that the level of embodied energy retained inside the elements is higher compared to the current recycling processes applied in practice.
  On economical incentives the literature study concluded that theoretical studies claim it to be beneficial however in practice this has not been proven yet. The main uncertainties for investors are relatively high risks on return on investments. And the negative influence caused by the time value of money.

- Connection Costs
  One of the main influences on extra costs for implementing DfD are the connections. For this reason it can be concluded that creating deconstructable connections at the same costs as non-deconstructable connections, can significantly lower the bar to apply DfD.

- Structural Element Reuse
  Based on lifetime differences between element groups and the wish to fully utilise these lifetimes, as well as the low amount of expected change in user wishes for structural elements, the literature study concluded that this is the element group suited for reuse.
  Furthermore the types of structural elements that will be reused are floor plates, beams and columns due to their generic character.

- Design Cycle Changes
  The literary study indicated that the design process is strained when designing with reusable elements. Strained due to the reduced design freedom and increased amount of design constraints, which increases the time

intensity of the design process. By implementing the genetic algorithm this strain is overcome as can be concluded from the 1 hour calculation time required for finding the reuse options in the 'Al Wakrah' stadium project.

- Subset Configuration Method
  From the 'Al Wakrah' results it can be concluded that, due to the discrete steps provided by the subset configuration method, no perfect fit is likely to be found. The method approaches the given conditions by an error margin on the building mass shape of 3% and deviating 1% from the required floorspace, while using all the available elements.

- Fittest Solution
  Because the genetic algorithm is a holistic algorithm, it cannot be concluded with 100% certainty that the fittest solution in the search space has been found. The validation results do show algorithm converges to (local) fitness peaks in the solution space. The decision whether these solutions are of sufficient quality is up to the designer.

- Practical Implementation Results
  Based on the information supplied by the application output, it can be concluded that the reuse algorithm does not provide a solution which is directly implementable in practice. Calculation on structural safety and design decisions on stability system, installations, facade and non structural elements still need to be done.
  The application setup does allow the addition of more design intelligence rules, possibly allowing these steps to be implemented in the algorithm in the future.

- Stadium to Township Translation
  From the three building test in Chapter 6 it can be concluded that, due to the non reusability of the raking beams and the stadium frame configuration, the reuse designs requires relatively more new floorbeams and columns compared to the amount of new floorplates.
- Variable Mutation Rate
  Exploratory tests to a variable mutation rate did not increase the algorithm performance. A possible explanation is that applying elitsm allows a mutation rate of 10%, ensuring enough genetic variation inside the generation, without the chance of losing the optimal solution because of too much mutation.

- Total Chromosome Mutation
  Exploring total chromosome mutation did not improve algorithm performance. Reasons for why this did not work cannot surpass mere speculation without further testing.

9 | Recommendations

Based on the discussion and conclusion, two future directions of the reuse application will be recommended, being development of the application as a solo application or integrating it into already existing applications.
The chapter concludes by presenting general recommendations independent from the future development direction of the application.

9.1 Stand alone Development

So far the application has been setup as a stand alone project, it does not import data or use methods from other applications.
The first step to increase the application usability as a stand alone application could be introducing an algorithm rule calculating the structural capacity. Doing so ensures that the designed frame is in line with the building codes and that the given solution is feasible, avoiding the chance that the found frame cannot be constructed. The other advantage of this rule is that the variable load that the frame can handle, can be matched to a user specified required variable load. In doing so the algorithm can converge to a frame which is feasible for the given load, while seeking to reduce the amount of overcapacity of the frame.
The second step in increasing the usability is introducing a second layer algorithm. The goal of this second layer algorithm would be to cope with the interaction of the structural frame with the lower building levels. After the first structural frame is defined, implications on the designed frame due to room divisions, vertical transport and stability elements are introduced. When these implications are implemented, the second layer algorithm is run, checking whether a fitter solution can be found that respects the lower building level implications.
The final recommendation for the stand alone application is to enable the introduction of more complex building geometry. Looking at the build environment, there is a demand for angled and curved facades, facades with cannot be handled in the current state of the application. Enabling for instance the outer building boxes to accept these more complex forms can increase the project market for the application and thus the usability of the tool in practice.

9.2 Application Integration

The recommendations made in the previous section can be developed to a final state where the entire design process is run by the application, while at set moments in the design chain there is a possibility for user interaction to steer

the design direction.
However in doing so a lot of work would be doubled, because design processes are already available in other applications. This section will therefore scout the integration options of this applications with others.
The reuse algorithm develops a building with the structural frame as starting point, introduction required stability systems later on in the process. This can be a viable approach for relatively simple structures, however when more complex shapes are demanded, the given solutions can have low significance due to the high impact of the required stability systems.
For these kinds of projects the start point should be the development of the stability system and than place the reusable elements inside this frame after. This most likely will lead to a reduction in the amount of elements reused, however the feasibility of the solution is guaranteed. An available integration option with a tool that starts from the stability system would be merging the reuse application with 'Structural Components' [van den Weerd et al. 2012].
A second integration option would be on the sustainability aspect of the design. Based on the toolbox created by Heidegger [Heidegger, 2013], it is possible to integrate sustainability aspects of a building in an early design phase, the design phase where this application starts as well. This could lead to a building frame configured to a building mass respecting sustainable aspects, which is followed up by a facade design with respect to sustainability. Due to the environmental incentives for the reuse strategy, integrating the reuse algorithm with the sustainability toolbox, could strengthen the design results.
Finally an integration with the sustainability assessment tool from Kokkos [Kokkos, 2014], can increase the transparency of the reuse project. Producing more precise numbers on the sustainability front to communicate the efficiency and strength of the reuse strategy and if positive results are produced, increasing the incentives for applying reuse.

9.3 Other Recommendations

- Costs and Environmental impact
  A cost calculator could be introduced, which can, based on the section type, lengths and reused amounts, calculate the difference in costs between reusing and designing with new elements.

- Loss percentages
  Researching the loss percentages per material and connection type, can lead to taking into account a margin in the amount of elements placed in the reuse frame, to make sure the frame can be finished in spite of element failure during (de)construction or transport.

- Logistic process
  When assigning the element locations in the reuse projects, the application considers the order in which it encounters the elements in the structural element list. This does not necessarily mean these elements are clustered in a manner that increases the availability of the element set required to construct the reuse project. However implementation of a logistics rule could speed up the construction speed of the reuse project and trough this aspect also increase the feasibility of the reuse strategy.

Bibliography

Adams, S. (1989) *Practical Buildability*, London: Construction Industry Research and Information Association.

Alexander, C. (1964) *Notes on the Synthesis of Form*, Cambridge (Massachusetts): Harvard University Press.

Ballast Nedam (2013) *Ballast Nedam ontwikkelt modulair stadionconcept: Plug and Play*, [Online], Available: www.ballast-nedam.nl [28 August 2013].

Blake, A. (2005) *The Economic Impact of the London 2012 Olympics*, Nottingham: UTSePress.

Hobbs, G. (2011) *Construction Waste Reduction around the World*, Rotterdam: CIB.

Construction Industry Research and Information Association (1983) *Buildability: an Assessment*, [Online], Available: http://www.opengrey.eu/item/display/10068/658015 [05 Februari 2014].

Coenders, J.L. (2007) 'Interfacing between parametric associative and structural software', *Innovations in Structural Engineering and Construction*, Taylor and Francis, pages 63-68.

Coenders, J.L. (2011) *NetworkedDesign: next generation infrastructure for computational design*, PhD Dissertation, Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Building and Structural Engineering, Structural Design Lab and BEM-Next Laboratory, and Arup Amsterdam.

Cormen, T.H., Leiserson, C.E., Rivest, R.L. & Stein, C. (1990) *Introduction to Algorithms*, Cambridge (Massachusetts): The MIT Press.

Crowther, P. (1999) 'Design for Disassembly', *Environmental Design Guide* [Electronic], Available: http://eprints.qut.edu.au/2882/ [31 November 2013].

te Dorsthorst, B., Kowalczyk, T., van Dijk, K., and Boedianto, P. (2000) 'State of the art deconstruction in the Netherlands', *State of the art deconstruction in selected countries*, publication 252, August, Chapter 6.

Durmisevic, E. (2006) *Transformable Building Structures*, PhD Dissertation, Delft University of Technology, Faculty of Architecture.

Edmonds, J. & Gorgolewski, M. (2000) *Steel Component Design for Deconstruction*, [Online] Available: http://www.reuse-steel.org/files/Information%20papers/Deconstruction%20IP%2010-5.pdf [24 Aug 2013].

Fransen, R. and Vermeulen, K.D. (2012) *Innovative Stadium Design*, Master's Thesis, Delft University of Technology, Faculty of Architecture, Department of Building Technology.

Glias, A. (2013) *Designing with reused structural concrete elements*, Master's Thesis, Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Structural Engineering.

Halfen (2013) *DEMU Doorkoppelsystemen*, [Online], Available: www.Halfen.com [04-06-2013].

Heidegger, V. (2013) *EnergyFacade: operational energy optimizing tool for conceptual facade design*, Master's Thesis, Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Structural Engineering, BEM-Next Laboratory.

Hendriks, F. and Janssen, G.M.T (2003) *Use of recycled materials in construction*, [Online], Available: http://www.epa.gov/osw/conserve/imr/pdfs/recy-bldg.pdf [07-10-2013].

Hertogh, M.J.C.M. & Soons, F.A.M. (2012) *Introduction to Integral Design*, Delft, the Netherlands, TU Delft Lecture Notes

Den Hollander, A.H. (2010) *Structural feasibility study and design for a Portable stadium*, Master's Thesis, Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Design and Construction.

Hon, S. L., Gairns, D. A. & Wilson, O. D. (1988) 'Buildability: A Review of Research Practice', *Australian Institute of Building Papers*, vol. 3, 1988, p. 106.

Huijbrechts, P. (2010) *Conceptueel Bouwen*, Boxtel, the Netherlands, AEneas

Jeffrey, C. (2011) *Construction and Demolition Waste Recycling, A Literature Review*, [Online], Available: http://www.dal.ca/content/dam/dalhousie/pdf/sustainability/Final%20C%26D%20literature%20review.pdf [10-10-2013].

de Jong, K.A. (1975) *Analysis of the behavior of a class of genetic adaptive systems*, Technical Report, The University of Michigan, College of Literature, Science and the Arts, Computer and Communication Sciences Department.

Kleinsman, M.S. (2006) *Understanding Collaborative Design* PhD Dissertation, Delft University of Technology, Faculty of Industrial Design.

Klomp, M.C. (2013) *Preliminary structural design and financial feasibility study of a transportable multifunctional stadium*, Master's Thesis, Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Structural Engineering.

Kokkos, A. (2014) *Computational Modelling tools for the promotion of Design for Deconstruction*, Master's Thesis, Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Structural Engineering, BEM-Next Laboratory.

Lichtenberg, J. (2005) *Slim Bouwen*, Boxtel: AEneas.

Loosjes, M. (2011) *Preliminary structural design of a demountable stadium*, Master's Thesis, Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Structural and Building Engineering.

Mitchell, M. (1999) *An Introduction to Genetic Algorithms*, Cambridge (Massachusetts): MIT Press.

Naber, N.R. (2012) *Reuse of hollow core slabs*, Master's Thesis, Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Structural and Building Engineering.

De Ridder, H.A.J. (2006) *Living Building Concept*, Gouda: PSIBouw.

De Ridder, H.A.J. (2011) *Legolisering van de Bouw*, Amersfoort: MGMC.

Rijkswaterstaat (2013) *Nederlands afval in cijfers, 2006-2010*, Utrecht: Rijkswaterstaat Leefomgeving.

Roders, M.J. (2003) 'IFD Bouwen in Japan, America and Europe', *IFD Construction - Congress Proceedings*, ISARC2003 congress, Technische Universiteit Eindhoven.

Terwel, K.C. et al. (2009) *Design of Structures*, Dutch reader, Delft University of Technology.

Zeiler, W. (2009) *Integral Open Building Design Methodology*, Dutch reader, Delft University of Technology.

Webster, M. & Costello, D. (2005) 'Designing Structural Systems for Deconstruction', *Greenbuild Conference Proceedings*, Atlanta.

van den Weerd, B. (2012) *StructuralComponents - a software system for conceptual structural design*, Master's Thesis, Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Structural Engineering.

Appendices

A | List of abbreviations

- IFD: Industrial Flexible Design
- LBC: Living Building Concept
- WFD: Waste Framework Directive
- C&D waste: Construction and Demolition waste
- DfD: Design for Deconstruction
- LoR: List of Requirements
- GA: Genetic Algorithm
- UML: Unified Modeling Language
- PAM: Parametric Associative Modeling
- PVC: Polyvinyl Chloride

B | Glossary

- Dry connection
  A connection between building components which is constructed without applying liquid materials as adhesive agent. A good example of a dry connection is the bolted connection.
- Wet connections
  A connection between building components which is constructed by adding a liquid material as adhesive agent. A good example of a wet connection is a grouted connection
- Monolithic Structures
  A structure consisting of mainly wet connections.
- C-values Value defining the quality of the sight lines from the stadium stands to the field.

C | Mutation Experimentation

The previous paragraph introduced a variable mutation rate, a mutation rate dependent on the ratio between the fittest score and the average generation score. This section will elaborate on the testing of two aspects of the mutation process, one being on how the mutation occurs in a chromosome and the other on the influence of the variable mutation rate. The tests check how these aspects influence the amount of loops required to find the final solution.

Chromosome Mutation

The first mutation aspect that is tested is how the mutation occurs in the chromosome. Two options are researched, the first option is checking for each gene individually if mutation occurs and if it does assign a random value to that gene. The second option applies mutation per chromosome, if mutation occurs the entire chromosome is mutated, assigning random values to each gene in the chromosome (Figure C.1). The difference between both mutation methods can be visualised by the fitness landscape, when only 1 or a few genes in the chromosome change as in option 1, the newly found location is close to the old one in the search space, while mutating the entire chromosome as in option two, the newly found solution can be anywhere in the solution space, more likely leading to the inclusion of new genetic material.

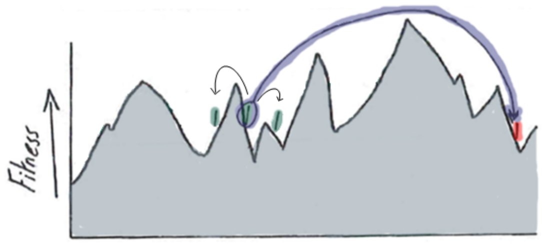


Figure C.1: Single gene mutation often leads to solutions close by, while total chromosome mutation leads to placing anywhere in the search space.

The advantage of mutation per gene is that is more likely to deliver a relatively fit solution because fit solutions are thought to be clustered together, making it more likely that the solution is selected for the next generation and be used. However this type of mutation delivers only a small bit of new genetic

material, making the chance that the algorithm stays stuck on the same local peak larger.
Mutation per chromosome means a higher chance on new genetic material, however because of the random location in the search space, the chance on a very poor fitness score increases meaning that the chromosome will not be selected and the new material wont be used by the algorithm.
To this end a 100 tests have been run with the gene mutation set at a 5% rate and 100 tests with entire chromosome mutation set at a 1% rate as shown in Table insert table here.

Variable Mutation Rate

The second test done on the mutation implementation is the variable mutation rate. The variable mutation is implemented as a inversely proportional correlation between the ration of average/fittest score and the mutation rate (Figure C.3). Meaning that when the difference between the average and fittest score is great, the mutation rate will be low and vica versa.

Average Generation Scores

```
 98 1399
 99 7147
100 1008
101 1443
102 1827
```

Fittest Generation Scores

```
 98 835
 99 835
100 835
101 835
102 835
```

New Fittest Score Loop Numbers

```
 7 21
 8 163
 9 185
10 252
11 268
```

Figure C.2: Blue line represents the fittest score and the green line the average generation score.

This means crossover is dominant when diverse genetic material is in the generation and mutation is dominant when this diversity is low, presumably allowing the algorithm to crossover towards (local) peaks while reducing the chance of the algorithm getting stuck. To this end 100 tests have been run with the variable mutation rate and 100 tests with a fixed mutation rate set at 1%. With mutation implemented as total chromosome mutation, because this won in the previous mutation test.
The test results are presented in Table insert table, showing that the hypothesis of the variable mutation rate is correct. Allowing the crossover process to work when enough genetic diversity is available and introducing more new material

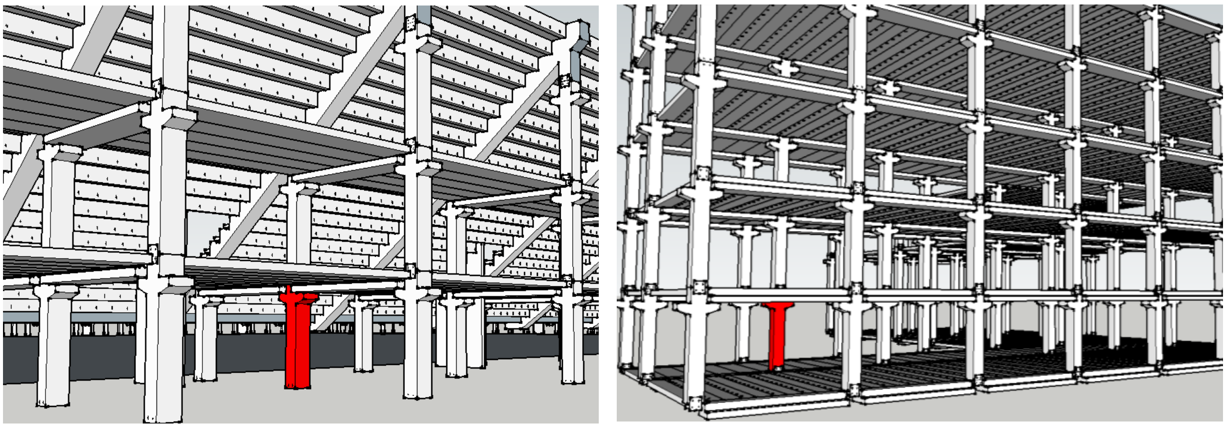
Figure C.3: The correlation between the ratio of the average and fittest score to the mutation rate.

when genetic diversity is low works for this kind of solution landscape. However testing on different types of solution landscapes is required before more general conclusions can be drawn.

D | Frame Calculations

Column load Comparison

The difference in structural configuration between the stadium and the residential reuse function lead to a different maximum column loading. Because these elements are in a preliminary design phase the required reinforcement has not yet been determined, however based on the geometry in concrete the loads on these columns can be compared to check the feasibility of the column reuse.



The red column on the left is the maximum loaded column in the stadium and the red column on the right is the maximum loaded column in the residential building.

Loads

Variable load stadium: $5{,}0\ \frac{kN}{m^2}\ and\ \psi = 0{,}25$ → Column loaded by stadium bowl and 1,5 floors

Variable load residential: $1{,}75\frac{kN}{m^2}\ and\ \psi = 0{,}4$ → Column loaded by roof and 4 floors

Stadium Load

Permanent load (U.L.S.)

$$Total\ Permanent\ Load = Bowl\ Elements + Raking\ Beam + Floorplates + Columns + Floorbeams \rightarrow 1{,}2\left((0{,}6m * 0{,}2m + 0{,}48m * 0{,}2m) * \frac{1000mm}{600mm} * 8m + 0{,}775m * 0{,}6m * 6m + 1{,}5 * 8m * 5m * 0{,}2m + 2 * 0{,}5m * 0{,}6m * 3{,}6m + 1{,}5(0{,}5m * 0{,}25m + 1m * 0{,}2m) * 5m\right) * 25\frac{kN}{m^3} \approx \boldsymbol{670kN}$$

Variable Load (U.L.S.)

$$Total\ Variable\ Load = Stadium\ Bowl + 1{,}5\ Floor \rightarrow 1{,}5\left(8m * 5m * 5{,}0\ \frac{kN}{m^2} + 1{,}5 * 8m * 5m * 5{,}0\ \frac{kN}{m^2} * 0{,}25\right) \approx \boldsymbol{415\ kN}$$

Total Column Load (U.L.S.)

$$Total\ Load = Permanent\ Load + Variable\ Load = 670kN + 445kN = \boldsymbol{1115kN}$$

Residential Load

Permanent load (U.L.S.)

$Total\ Permanent\ Load = 1{,}2 * (Floorplates + Columns + Floorbeams) \rightarrow 1{,}2 * (5 * 8m * 5m * 0{,}2m + 4 * 0{,}5m * 0{,}6m * 3{,}6m + 5 * (0{,}5m * 0{,}25m + 1m * 0{,}2m) * 5m) * 25\frac{kN}{m^3} \approx \mathbf{1575}\boldsymbol{kN}$

Variable Load (U.L.S.)

$Total\ Variable\ Load = 1max\ Floor + 3\ Floor + roof \rightarrow 1{,}5\left(8m * 5m * 1{,}75\frac{kN}{m^2} + 3 * 8m * 5m * 1{,}75\frac{kN}{m^2} * 0{,}4 + 8m * 5m * 1\frac{kN}{m^2}\right) \approx \mathbf{290}\boldsymbol{kN}$

Total Column Load (U.L.S.)

$Total\ Load = Permanent\ Load + Variable\ Load = 1575kN + 290kN = \mathbf{1865}\boldsymbol{kN}$

Load Comparison

$\frac{1865kN}{1115kN} \approx 1{,}67 \rightarrow$ For this reason the columns in the stadium need to be over-dimensioned to be able to handle the loads from the reuse function. Or if this is unwanted the residential building height should be restricted to three levels as shown in the calculation below.

Permanent load (U.L.S.)

$Total\ Permanent\ Load = 1{,}2 * (Floorplates + Columns + Floorbeams) \rightarrow 1{,}2 * (3 * 8m * 5m * 0{,}2m + 2 * 0{,}5m * 0{,}6m * 3{,}6m + 3 * (0{,}5m * 0{,}25m + 1m * 0{,}2m) * 5m) * 25\frac{kN}{m^3} \approx \mathbf{930}\boldsymbol{kN}$

Variable Load (U.L.S.)

$Total\ Variable\ Load = 1max\ Floor + 1\ Floor + roof \rightarrow 1{,}5\left(8m * 5m * 1{,}75\frac{kN}{m^2} + 1 * 8m * 5m * 1{,}75\frac{kN}{m^2} * 0{,}4 + 8m * 5m * 1\frac{kN}{m^2}\right) \approx \mathbf{205}\boldsymbol{kN}$

Total Column Load (U.L.S.)

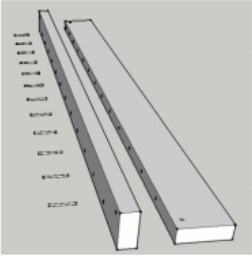
$Total\ Load = Permanent\ Load + Variable\ Load = 930kN + 205kN = \mathbf{1135}\boldsymbol{kN}$

Leading to a nearly equal load between stadium function and residential function.

Hand Calculation Coupled Stand Elements

The calculation is done based on a strength check of the coupling elements. Besides this first check, recommendations are done for further calculations.

The figure on the right shows the coupling elements, which are assumed to be the smallest bolts produced for the DEMU coupling system (Ø12). Which have a design strength of 49,2 kN.



Load Distribution Vertical and Horizontal Element

Due to differences in stiffness the load exerted on the stand element is not equally distributed among the vertical and horizontal element. The ratio for this load distribution is calculated here, assuming that the coupling is a full strength connection.

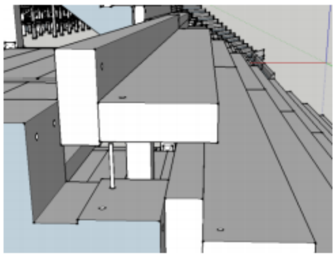
$\delta_{vert} = \frac{5}{384} * \frac{q_{vert} * l_{vert}{}^4}{E_{vert} * I_{vert}}$ and $\delta_{hor} = \frac{5}{384} * \frac{q_{hor} * l_{hor}{}^4}{E_{hor} * I_{hor}}$ with $\delta_{vert} = \delta_{hor},\ l_{vert} = l_{hor}$ and $E_{vert} = E_{hor} \rightarrow$

$q_{vert} * I_{hor} = q_{hor} * I_{vert} \rightarrow q_{vert} * \frac{1}{12} * b_{hor} * h_{hor}{}^3 = q_{hor} * \frac{1}{12} * b_{vert} * h_{vert}{}^3 \rightarrow$

$q_{vert} * \frac{1}{12} * 600mm * (200mm)^3 = q_{hor} * \frac{1}{12} * 200mm * (480mm)^3 \rightarrow q_{vert} \approx q_{hor} * 4{,}6 \rightarrow \frac{4{,}6}{1+4{,}6} \approx 0{,}82 \rightarrow$

$$\boldsymbol{q_{vert}\ takes\ 18\%\ of\ the\ total\ load\ and\ q_{hor}\ takes\ 82\%\ of\ the\ total\ load.}$$

Load Exerted on Stand Element



Loads on Horizontal part:

Variable load: $5{,}0 \frac{kN}{m^2} * 0{,}6m * 1{,}5 = 4{,}5\ \frac{kN}{m}$

Permanent load: $0{,}6m * 0{,}2m * 25 \frac{kN}{m^3} * 1{,}2 = 3{,}6\ \frac{kN}{m}$

Loads on Vertical part:

Variable load: 0

Permanent load: $0{,}2m * 0{,}48m * 25 \frac{kN}{m^3} * 1{,}2 = 2{,}9\ \frac{kN}{m}$

Total load (ULS)

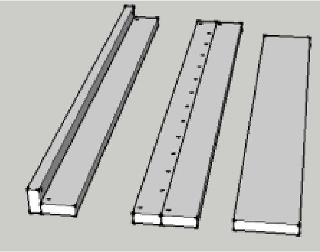
$4{,}5\ \frac{kN}{m} + 3{,}6\ \frac{kN}{m} + 2{,}9\ \frac{kN}{m} = 11 \frac{kN}{m}$ from which $11 \frac{kN}{m} * 0{,}18 = 2 \frac{kN}{m}$ is taken by the horizontal part and $11 \frac{kN}{m} * 0{,}82 = 9 \frac{kN}{m}$ by the vertical part.

Load per DEMU connection

The load transferred per connection is: $\frac{(load\ exerted\ on\ horizontal\ element - load\ taken\ by\ horizontal\ element)}{2\ (since\ the\ load\ can\ be\ transferred\ to\ vertical\ elements\ at\ both\ sides)} \rightarrow$ $\frac{\left(4{,}5\frac{kN}{m}+3{,}6\frac{kN}{m}\right)-2\frac{kN}{m}}{2} = 3{,}2\frac{kN}{m}$. With the bolts spaced 600mm this gives: $3{,}2\frac{kN}{m} * 0{,}6m = 1{,}92\ kN$ per DEMU connection.

With the design strength of the connection with Ø12 bolts is $49{,}2\ kN$. The connections have no problem taking the load. They are over dimensioned by a factor 25. However there will also be a vibration load on this connection, for this load case the stiffness of the vertical element has a positive influence and therefor in this design stage a full strength connection is required and the bolts are not yet spaced further from each other.

Strength in Reuse Design



When the reuse function is a housing project the variable load is $2{,}0\frac{kN}{m^2}$ which is lower than the $5{,}0\frac{kN}{m^2}$ variable load in the stadium. However when the vertical beam element is reused as a horizontal plate, it stiffness along the loaded axis will be lower. Due to this reduced stiffness this loadcase could be governing. A quick check on deflection gives:

Variable load: $2{,}0\frac{kN}{m^2} * 0{,}6m = 1{,}2\ \frac{kN}{m}$

Permanent load: $0{,}2m * 0{,}48m * 25\frac{kN}{m^3}\ = 2{,}4\ \frac{kN}{m}$

Total load = $1{,}2\ \frac{kN}{m} + 2{,}4\ \frac{kN}{m} = 3{,}6\ \frac{kN}{m}$

Deflection: $\delta = \frac{5}{384} * \frac{ql^4}{EI} = \frac{5}{384} * \frac{3{,}6\frac{N}{mm}*(8000mm)^4}{38*10^3\frac{N}{mm^2}*\left(\frac{1}{12}*480*200^3\right)} \rightarrow \delta \approx 14\ mm$

Maximum allowable deflection: $\delta_{max} = 0{,}004\ l = 0{,}004 * 8000mm = 32\ mm > 14\ so\ its\ oke.$ However over dimensioned by a factor 2. Even though it checks out, the changes of element properties when reusing them should always be checked.

Further calculations

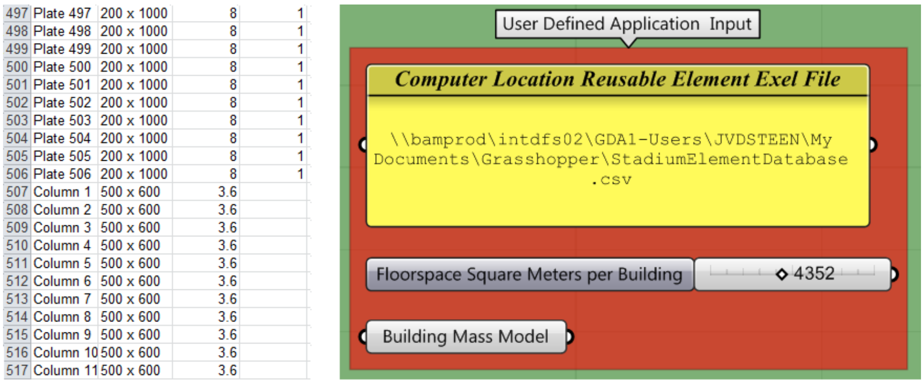
The connection type used for the stand element checks out on strength and is over dimensioned by a factor 25. Before final conclusions can be made on the implementation of this connection, in a further design stage a vibration calculation needs to be done.
This hand calculation only shows the feasibility of the connection needed for the preliminary design.

E | Application Manual

This appendix forms the user guide to work with the reuse application. The user is guided trough the steps by means of short descriptions and screen shots.

Reusable Element Input

The reusable elements are setup as an excel file, the setup of the elements in this excel file is shown in Figure E.1 (left). In the model the code path has to be provided, this code path specifies the location of the excel file (saved as CSV file) on the computer. (Figure E.1, right)



497	Plate 497	200 x 1000	8	1
498	Plate 498	200 x 1000	8	1
499	Plate 499	200 x 1000	8	1
500	Plate 500	200 x 1000	8	1
501	Plate 501	200 x 1000	8	1
502	Plate 502	200 x 1000	8	1
503	Plate 503	200 x 1000	8	1
504	Plate 504	200 x 1000	8	1
505	Plate 505	200 x 1000	8	1
506	Plate 506	200 x 1000	8	1
507	Column 1	500 x 600	3.6	
508	Column 2	500 x 600	3.6	
509	Column 3	500 x 600	3.6	
510	Column 4	500 x 600	3.6	
511	Column 5	500 x 600	3.6	
512	Column 6	500 x 600	3.6	
513	Column 7	500 x 600	3.6	
514	Column 8	500 x 600	3.6	
515	Column 9	500 x 600	3.6	
516	Column 10	500 x 600	3.6	
517	Column 11	500 x 600	3.6	

User Defined Application Input

Computer Location Reusable Element Exel File

```
\\bamprod\intdfs02\GDA1-Users\JVDSTEEN\My
Documents\Grasshopper\StadiumElementDatabase
.csv
```

Floorspace Square Meters per Building ◇ 4352

Building Mass Model

Figure E.1: The excel specifies 1 element per row, for each element the ID, Section and Length are notated (for plates also the width).

Building mass & Floorspace

Next up is defining the building mass, which can be located anywhere in the rhino viewport. The buiding mass may consist of 1 or 2 connected rectangular boxes. Do not worry about the scaling of the mass, the application will do this after the amount of floorspace is specified in the input (Figure E.1, right figure 'Floorspace Square meters per Building').
After the building mass is defined in the rhino viewport, the two boxes need to be assigned to the 'Building Mass Model' input (Figure E.1, right). When this is done correctly, the rhino viewport will now show the amount of scaled

buildings that can be created. (Figure E.2)

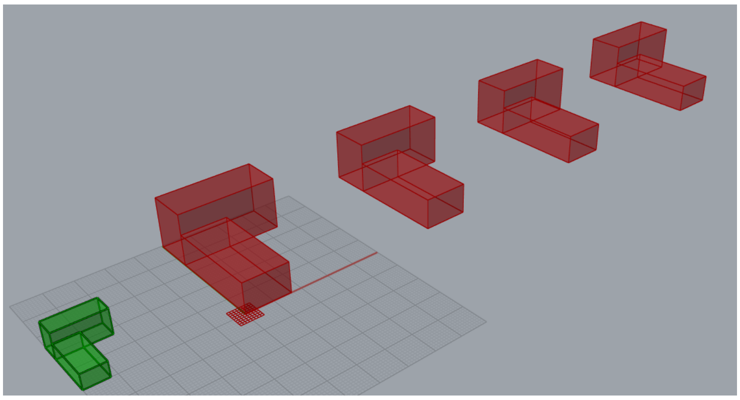
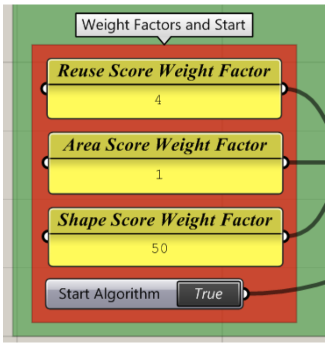


Figure E.2: The green building is the mass specified by the user and assigned to 'Building Mass model', the red figures are the scaled building masses.

Weight factors and Start

The final input that has to be provided are the Scoring Rule Weight factors (Figure E.3). Setting these weight factors influences the outcome of the algorithm, based on user priorities. After these weight factors are set, the user can start the algorithm by setting the 'Start Algorithm' boolean to 'True' (Figure E.3).



Weight Factors and Start

Reuse Score Weight Factor: 4

Area Score Weight Factor: 1

Shape Score Weight Factor: 50

Start Algorithm: True

Figure E.3: By changing these integers the algorithm outcome can be influenced based on user priorities.

Now the algorithm will start running and after it is done iterating the final outcomes are presented in the Rhino viewport by means of a frame visualisation and the grasshopper application will show the reuse percentage, the floorarea used in each building and the location of every element in the structural frames.

List of Figures

List of Tables