Parametric approach for designing affordable & high density housing

A study towards improving affordable and high density housing, using the advantages of computerization in architectural practice.

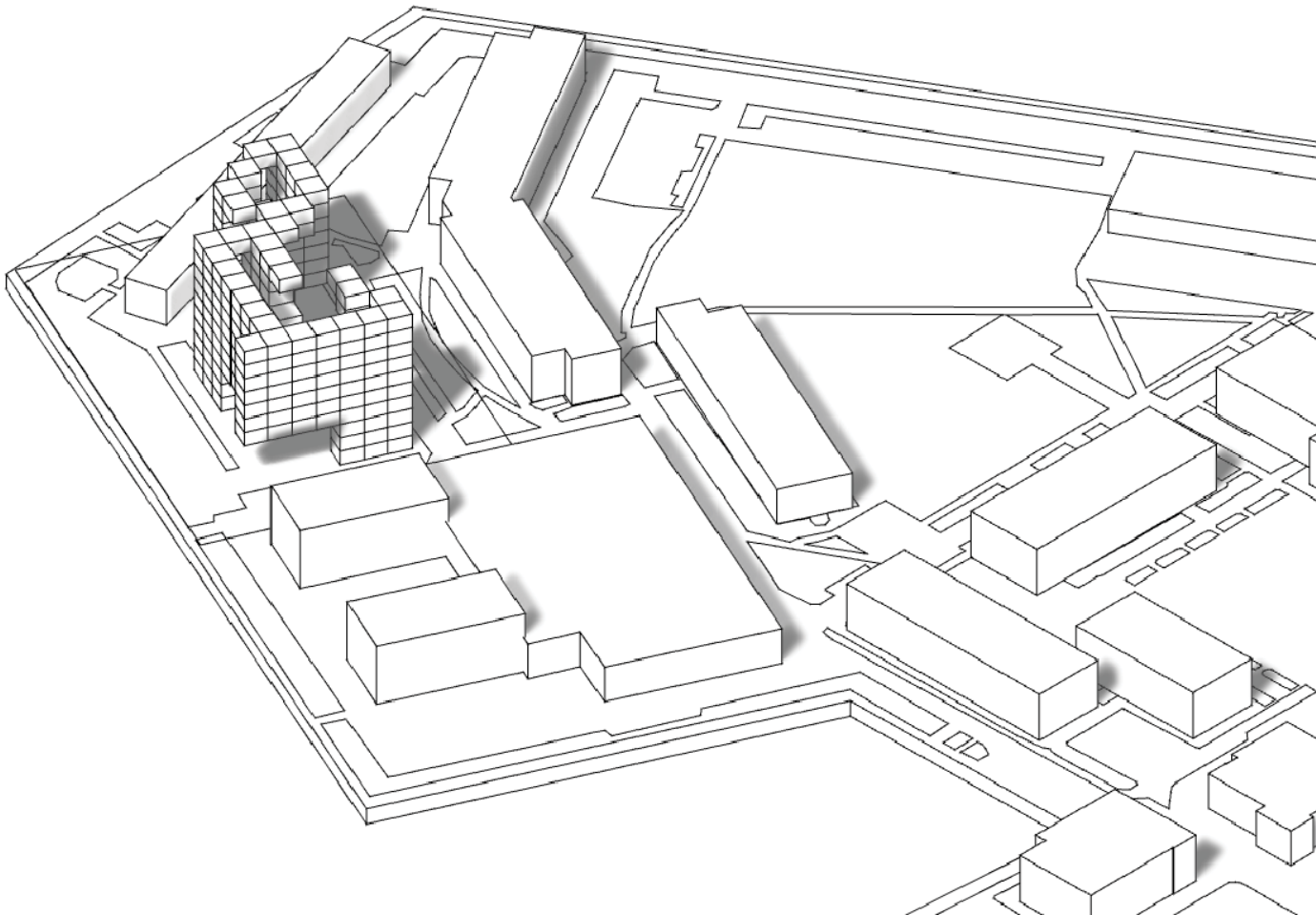
Thematic Research Paper

Name & student nr.
Sjoerd Poelman, 4142993

Studio
Architectural Engineering / Intecture

Tutors
Mo Smit & Pieter Stoutjesdijk

Date
27-02-2017



Abstract

The subject of this paper looks into designing high density residential architecture that enables certain qualities, in order to improve the livability and perception of the design. A parametric design approach was chosen for this project in order to enable these qualities. First, several important conditions and criteria for this project were formed. These form guiding principles for the parametric models. This is important, since parametric modeling software can often lead to reflexive design criteria and unnecessary complexity. In order to avoid this, relation to context and social relevance become important guiding criteria. View, daylight, orientation and privacy are all guiding criteria that contribute to this social relevance.

The first Grasshopper model was then created. By switching between different domains (or design aspects) within a design process, and combining their results, a qualitative design can eventually be created. The first Grasshopper model, however, forms a more linear process, in which every step within the created model is placed within one continuous code. A second Grasshopper model was then created. This model showed promise by allowing better differentiation between de different design domains. However, by basing the input parameters of the second Grasshopper model on a gradient image, the input parameters became in a way fixed within this model. With the help of parametric *form-finding*, the architect can create a certain kind of differentiation of elements within his design, giving him more freedom in the composition of the design within the design process. However, parametric codes are created with a certain algorithmic logic. This can be limiting the creative freedom of the designer and a design could then possibly lose its *human-touch.* Therefore, the designer should always be in control over the parametric code he created, changing the role of the designer from a physical designer to a more virtual designer, a programmer, who is in control of all the aspects of parametric design (input, algorithmic process and output).

By reflecting on each Grasshopper model, in order to understand the strengths and weaknesses, the complexity of the design process of designing high density housing itself could be made more understandable and rationalized. However, according to certain Literature (Simon 1973; Lawson 1980) the design process can never be fully rationalized. This is because the design process can also be described as ill-structured. Understanding the advantages of the computer in well-structured problems and its disadvantages in ill-structured problems, is also important for understanding the need for heuristics within the parametric design approach. Rowe (1982) states that the architectural design process is inherent heuristic. This means that the further a design process advances, more conditions become clear and therefore, the better the design process could be rationalized by the designer or architect. Within the design and development of the parametric tool itself, this would mean that perhaps a parametric code / program could be designed in such a way, that it can rewrite itself according to what it has learned. In languages that support *Object Oriented Programming* (OOP) an *object* can be seen as a structure that contains both data and procedures. Understanding these procedures will enable the designer to reflect on the input and output data, making a cooperative and iterative trial-and-error process possible between the computer and the designer.

Keywords

Design Process, Computer-Aided Design, Intuition, Subjective Judgement, Rationalization, Parametric Approach, Grasshopper, High Density, Affordable Housing. Spatial Perception, Spatial Optimization, Integration.

Index

1. Introduction

The process of developing an idea or architectural vision towards creating an actual design or creating architecture is a very complex design process that could be described in many ways. Architects and designers are able to always surprise us with finding new and innovative solutions for problems concerning their design. They are able to do this through a complex thinking process of which its exact nature and operation is not clear. Since the rise of computer technology and the use of the computer in architectural practice, the design process has gained a new interest again (Cross, 2007). New computerized tools are becoming available of which their purpose is to support the architects in creating their design or even stimulate the design process itself in order to find innovative new solutions for problems concerning their design and architectural practice as a whole.

In the earlier years of computer technology, Steven Coons (1963**)** already describes *Computer-Aided design* (CAD) as a cooperative system between the computer and the designer. In this cooperative system the creativity of the architect could be combined with the analytical power of the computer. This would then result in a more efficient way of designing.

The potential of Computer-Aided Design has been extended in various ways during the last decades. This increased potential resulted in a worldwide adaptation of CAD tools within the domain of architecture, engineering and construction, making a CAD system an essential tool within these disciplines (Lee et al., 2006). However, research has shown that recent developments in CAD systems, such as *Building Information Modeling* (BIM), mainly influence the later stages of the design process (Krish, 2010). In these later stages, the concept of the designer is already fixed and important design choices are already made. The influence of CAD tools is then limited to mostly building performance related domains. However, there is an increasing demand for computerized design tools that allow the designer to explore essential geometrical design variations in a flexible way within the early stages of the design process (Krish, 2010).

By implementing the possibilities of the computer in the early stages of the design process of the architect, it would result in a more cooperative design process. The computer and its tools would then be able to support or even stimulate the iterative trial-and-error design process of creating architecture. This cooperation between the computer and the architect needs to be embraced, since it will only help us to better understand the choices we make during the architectural design process.

The introduction of parametric modeling, allows the designer to explore these geometrical design variations within early stages of the design process. This technique, functions by defining various algorithms, constraints and parameters. By then fluctuating the different parameters, different design solutions can be found. Most important within this design exploration process is the notion of constraints. Constraints form the boundaries in between which a design solution can be found (Kilian, 2006).

Some now iconic architectural buildings and structures could not have been created without the extensive use of parametric tools within the design process. The use of these parametric design tools allow the designer to model very complex geometry in a considerably more efficient manner. However, this efficiency is not without its dangers. It is difficult to ignore the fact that parametric modeling software often leads to reflexive design criteria and unnecessary complexity. The architectural design could then be reduced to a meaningless cast without any social relevance. This is often referred to as *computational decoration* (Shea et al., 2005).

This problem of computational decoration is subject to many previously done research projects. These projects mostly propose methods focusing on finding useful constraints, which accommodate the problem of complexity and social relevance in order to refine the space in which to find the design solutions (Caldas, 2007). This paper adds upon this problem, but instead of focusing on finding the necessary design constraints, this paper shifts its focus more to the evaluation of a parametric approach within the early stages of the design process.

In a way this paper focusses on the use of Computer-Aided Design as a tool in the design process; more specifically, the possibilities of a parametric design approach within the early stages of the design process. By understanding the possibilities of a parametric approach within this design process, it would hopefully give more insight into the strengths, but also the limits of a parametric approach in architectural practice.

1.1 Problem statement

With computerization in architecture, parametric design tools and techniques, as well as digital fabrication techniques (like laser cutting, computerized milling and 3d printing) slowly but inevitably become viable and affordable solutions for solving existing and new problems in architecture.

One of these problems (the problem this paper shifts its focus towards) concerns high density architecture. Urbanization and population density are increasing in the Netherlands (especially in the Randstad). The result of this is that the big cities in the Randstad (Amsterdam, Rotterdam, Den Haag, Utrecht) are becoming much more dense and the population within these cities is growing (Rijksoverheid, 2015). In order to accommodate housing needs for the population rise in these cities, high density housing can be considered a viable solution.

When we design high density architecture, one of the most important aspects is how we perceive the design and how it affects its users and its environment (Lawson, 2010). However, when looking at a lot of high density architecture that exists, not only in the Netherlands, but also in other countries, this is not reflected in the designs. A lot of high density housing projects are designed with a certain modernistic view still in mind (Hulsman, 2009). This is not necessarily a bad thing. However, there are certain somewhat outdated dangers luring within this modernistic dogma.

Because of the Dutch *Woningwet* of 1901 and the emphasis on social housing in the 20th century, certain housing types

(the *rijtjeshuis*, *portieketagewoning* and the *galerijflat*) have been better developed compared to others in this time, resulting in designers and developers always falling back on these types of housing (Hulsman, 2009). Besides this, also the rigid regulations and the power of project developers in the mid-20th century, made city developers detest mixed use architecture. Living, working, shopping and recreating had to be strictly separated, holding in mind examples like the Bijlmer or Le Corbusiers' Plan Voisin (Kanteldenker, 2015). These modernistic ideas can lead to unattractive and unresponsive buildings placed in bare landscapes, in which the exchange between public and private domains is overshadowed by function separation and strict boundaries, with the result that spatial perception / experience came under pressure.

Illustration 1,2 – The Bijlmer, Plan Voisin (Kanteldenker, 2015)

We now know that the Bijlmer had its flaws. However, not all of these modernistic ideas were bad. For example, one of the main focus points in Plan Voisin by Le Corbusier was to create quality in recreative and green spaces, within the *unhealthy* dense city, where there is already so little space available (Uytenhaak, 2008). Le Corbusier focusses on designing buildings with a low contextual volume impact by creating compact buildings and stacking housing units into large residential towers, allowing large (recreative) spaces to exist in between these residential towers.

During the era of Modernism, the industrial revolution also played its role. New mass fabrication and replication techniques were invented in this time. Using techniques like these, high density buildings could now be constructed in an affordable and time saving manner, whilst providing housing for many people. However, these affordable fabrication and replication techniques could also lead to unattractive and unresponsive design. Affordable techniques like these would allow for similar housing units, which only accommodate a small variety of user needs.

Besides this, we know that the built environment directly affects its users. Spatial configuration of a design is a very important factor in creating user satisfaction (Lawson, 2010). Also, the experience of living in a high density environment is much more complex than a low density environment. Therefore, Personal relations and neigbourhood relations in a high density environment are directly affected by the spatial design of high density architecture. It is important to understand this relationship between the built environment and user satisfaction when designing affordable and high density residential architecture.

Aspects like; spatial perception and experience, contextual volume impact, relations between public and private domains and user quality in flexibility and customization all need to be considered during the design of high density housing in order to create a qualitative final design. This makes the design process of high density architecture to be very complex. In order for the design to be created, a balance between these different design aspects needs to be found. Rudy Uytenhaak (2008) describes this like:

“het vinden van een balans tussen de soms conflicterende parameters kan vergeleken worden met het bedienen van een schuifpaneel.”

Therefore, this research paper looks into the use of a parametric approach and its supporting role in the design process, in order to better understand its possibilities with designing affordable and high density housing

1.2 Research Question

Digitalization and computerization is becoming more used in this profession. Therefore, it is only logical to research this computerization and the tools that come with it. This paper looks into these tools, mainly parametric design tools, in order to make affordable and high density housing possible. The research question of this paper therefore is:

To what extent could a parametric approach in the design of high density & affordable housing help to enable certain qualities in: spatial optimization, spatial perception, low contextual volume impact, user quality in flexibility and customization?

For this paper it is important to note, that this paper does not focus on the exact parameters and boundary conditions needed for the parametric approach and the exact data and information that is the output of this approach. For this paper this is also very important, in order to be able to make progression within this paper. However, it is more important for this paper to focus on the theory behind the parametric approach used in this project. In order to learn more about *how* this parametric approach can be applied within this project, this paper therefore also focuses on the theory of the design process itself.

Understanding this process, could be the key in understanding the many possibilities of parametric design in architectural practice. In order to answer the this question, different sub-questions are formed within this research paper:

1. Which design constraints and conditions need to be considered in order to use a parametric approach as a helping tool in the design process for creating qualitative, affordable & high density housing.
   See chapter 3.1

2. What should be the structure of the parametric design model, created with these design constraints and conditions, and how does each step in this parametric model function.
   See chapters 3.2 & 3.3

3. What could be the ideal relation between the parametric tools and the designer, as both actors within the design process.
   See chapters 3.3.5, 3.4 & 3.5

4. How does this parametric approach relate to the structure of an overall design process and what are the advantages and disadvantages of a parametric approach compared to a non-parametric approach, for designing qualitative, affordable and high density housing.
See chapters 3.2.1, 3.3.5, 3.4 & 3.5

By answering these sub-questions within this research paper, more insight should be given into the main research question of this paper.

2. Methods

For the structure of this paper a strategy scheme was created. This scheme illustrates the process of the progression of this paper, but also the process of the parametric models themselves. The scheme consists of three different stages, all focusing on a different aspect of the research and the associated parametric design process. These stages are named: the FINDING STAGE, the MODELING STAGE and the EVALUATING STAGE. Each stage will be further elaborated within the next paragraphs of this chapter. Also, the associated methods needed, in order to be able to research each stage, will be described within the following paragraphs of this chapter.

2.1 The Finding Stage

The strategy scheme created for this research paper will start with the FINDING STAGE. This stage will focus mostly on the first sub-question of this paper. Since the focus of this paper is not on the exact data from the parametric models, but more on the design process of this parametric approach and its relation to the design context; the conditions / constraints and criteria for the parametric design models will be analyzed in a less concrete manner. The focus here is not on the exact starting

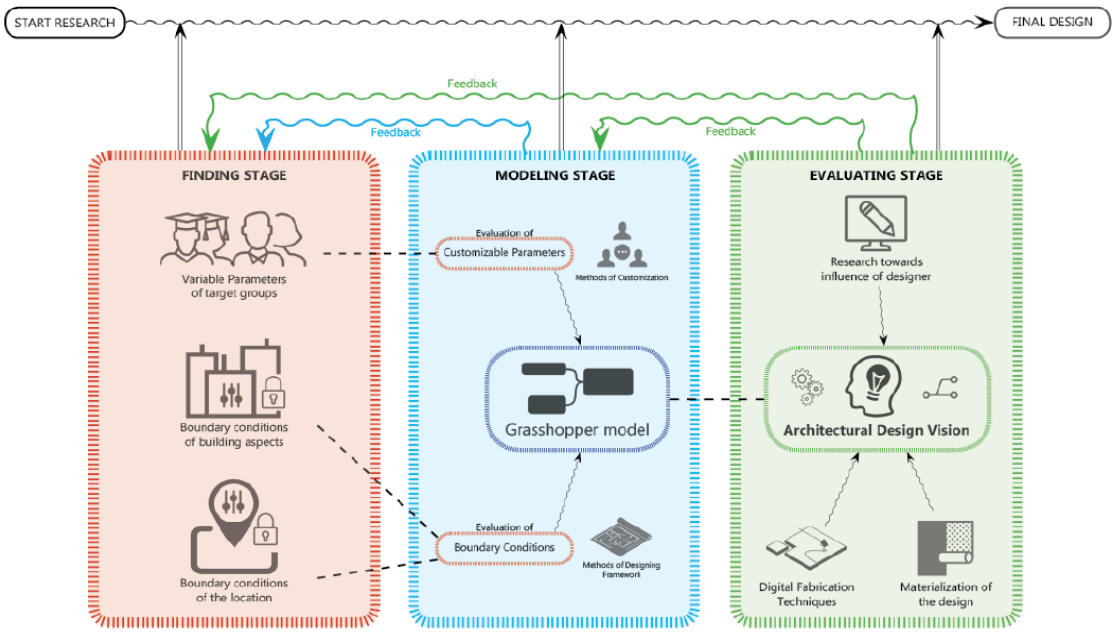


Illustration 3 – Overall Strategy Scheme

parameters, but on the necessary social and design aspects needed as starting conditions in order to enable the various qualities mentioned in the overall research question. These conditions will then function as an architectural guiding theme for the parametric models, in order to avoid unwanted reflexive design problems and unnecessary complexity.

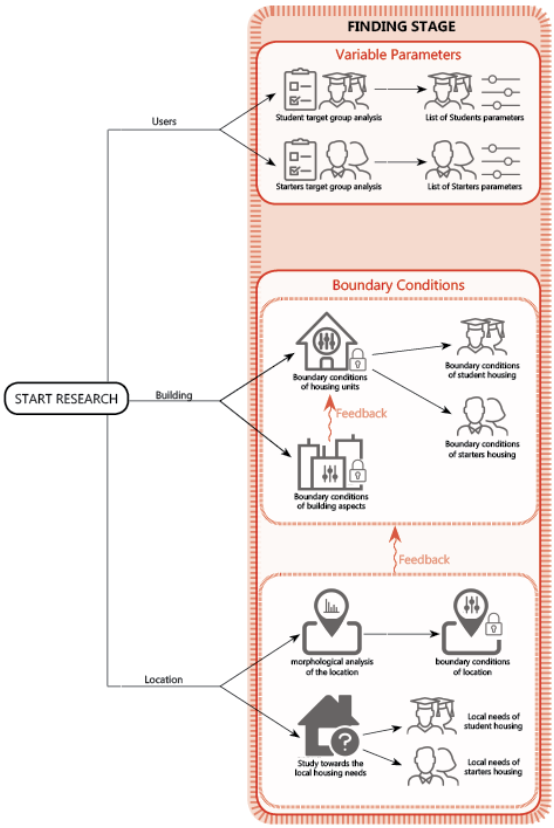


Illustration 4 – The Finding Stage

In order to find these conditions and criteria, literature studies will be done towards the different design aspects that are associate with the design of high density architecture, for example the work of Rudy Uytenhaak (2008): *Steden vol Ruimte. Kwaliteiten van Dichtheid*. Since the design of high density housing can become very complex very fast, this paper focusses only on the larger scale (location scale / building scale) of the design project in designing affordable and high density housing.

2.2 The Modeling Stage

After the necessary design aspects, conditions and criteria are described within the first stage of this paper, we move to the second stage, the MODELING STAGE. This stage focusses on creating the parametric design models (in a way these can be considered as case studies) for designing affordable and high density housing. Understanding the structure and working of these models is important for researching the possibilities of using a parametric approach. Therefore, this stage will mostly focus on answering the second sub-question of this research paper, by using *research by designing* as underlying methodology for creating the needed parametric models.

In order to create these parametric models, different design parametric methods / tools can be considered for this stage. In the end *Grasshopper* was chosen (along with Rhino, and certain plugins that are used within Grasshopper) as parametric design tool for the continuation of this research paper. Within this stage of the paper two Grasshopper models will be discussed, where the second Grasshopper model is a first attempt on improving the first Grasshopper model, using new insights gained from the evaluation of the first Grasshopper model.

2.2.1 Why Grasshopper?

Grasshopper is a program that visualizes code language to a line & box model, making its possibilities more graspable and understandable for its users (McNeel, 2014). Also, because Grasshopper is a plugin for Rhinoceros 5, Grasshopper can be used in the 3d virtual environment of

Rhino, allowing possibilities for parametric architectural design and visualization. With the help of this more visualized code language, Grasshopper can create parametric algorithms influencing models in a 3d environment, meaning that if the parameters of the algorithm change, it will also change the 3d model which is constraint by these parameters. This makes Grasshopper in combination with Rhinoceros an attractive design parametric design tool for geometry oriented designers, like architects.

Besides this, a lot of plugins also already exist for Grasshopper itself. These plugins allow the designer to test with various analysis based experimentations (for example sunlight analysis within a 3d environment using Ladybug (Roudsari, 2016)), as well as exploring the possibilities of generative design (allowing the computer to find the most ideal iteration on its own, based on certain rules and constraints, for example using Galapagos (Rutten, 2017)).

This and other features would make Grasshopper and its possibilities a suitable parametric design tool, that allows the designer to explore essential geometrical design variations in a flexible way within the early stages of a design process, which is subject to this research paper.

Other geometry based parametric design tools also were considered as design methods within this paper that allow this freedom in form finding exploration during the early stages of a design process, namely *Autodesk Dynamo* (a parametric design tool developed by Autodesk and similar to Grasshopper in visual lay-out as well as its parametric possibilities in correlation with geometry (Autodesk, 2017)) and *GenerativeComponents* (also a parametric design tool which is supported by various CAD software (also Rhinoceros), which allows for dynamic modeling of geometry as well as rule based modeling through expressed algorithms (Bentley, 2017)).

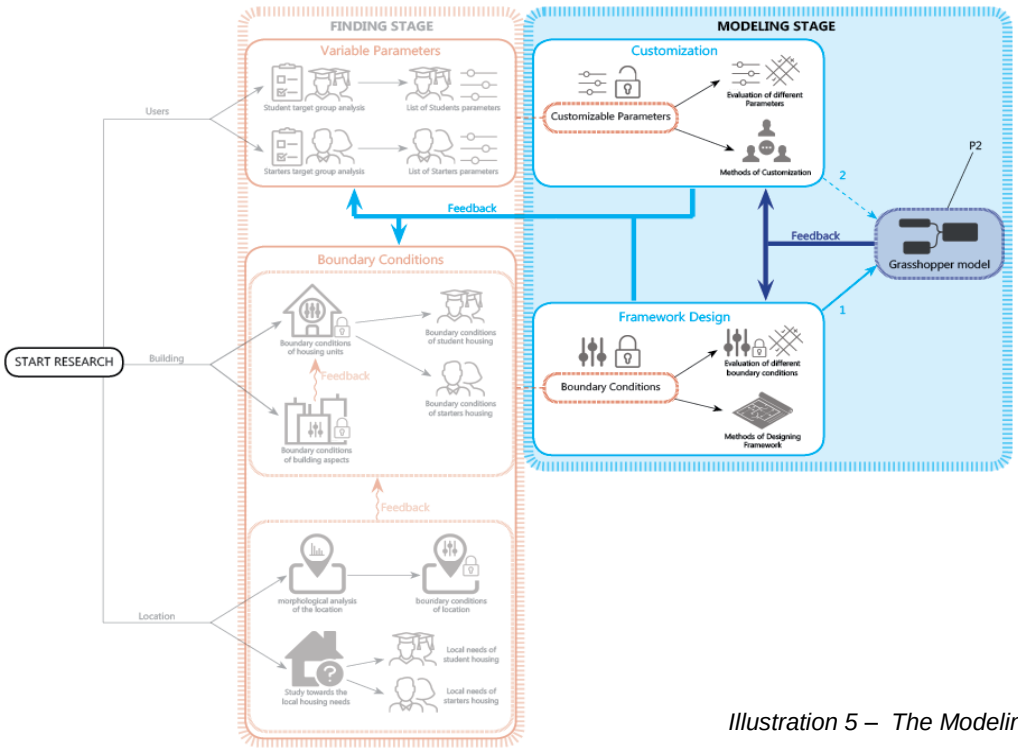


Illustration 5 – The Modeling Stage

In the end Grasshopper was chosen to be used as parametric design tool within this research project, because of its graspable and user friendly visualized code language, its possibilities with manipulating and creating geometry using rule based algorithms, its large active and helping community (McNeel, 2014), and its availability of supporting / useful plugins which allow various analytical and generative parametric possibilities within Grasshopper and Rhinoceros.

2.3 The Evaluation Stage

The EVALUATION STAGE can be considered to be the most important stage for this research paper. This stage will look more into the process of designing the different Grasshopper models in order to learn more about the possibilities of a parametric approach for this design project.

Therefore, this stage will also be looking more into answering the main research question of this paper. By doing *literature studies* considering the design process of creating architecture (for example the work of Elise van Dooren (2013) or the work of Herbert Simon (1973)), more insight can be given into the theory behind the design process itself. The focus of this stage is mostly on answering the last two sub-questions of this paper, considering the role of the parametric approach and its Grasshopper models in relation to the design process of designing affordable and high density housing.

Also, certain different approaches of software development will be discussed in this stage. These different approaches could allow for more insight to be gained about the further development of the parametric models, for the continuation of designing affordable an high density housing.

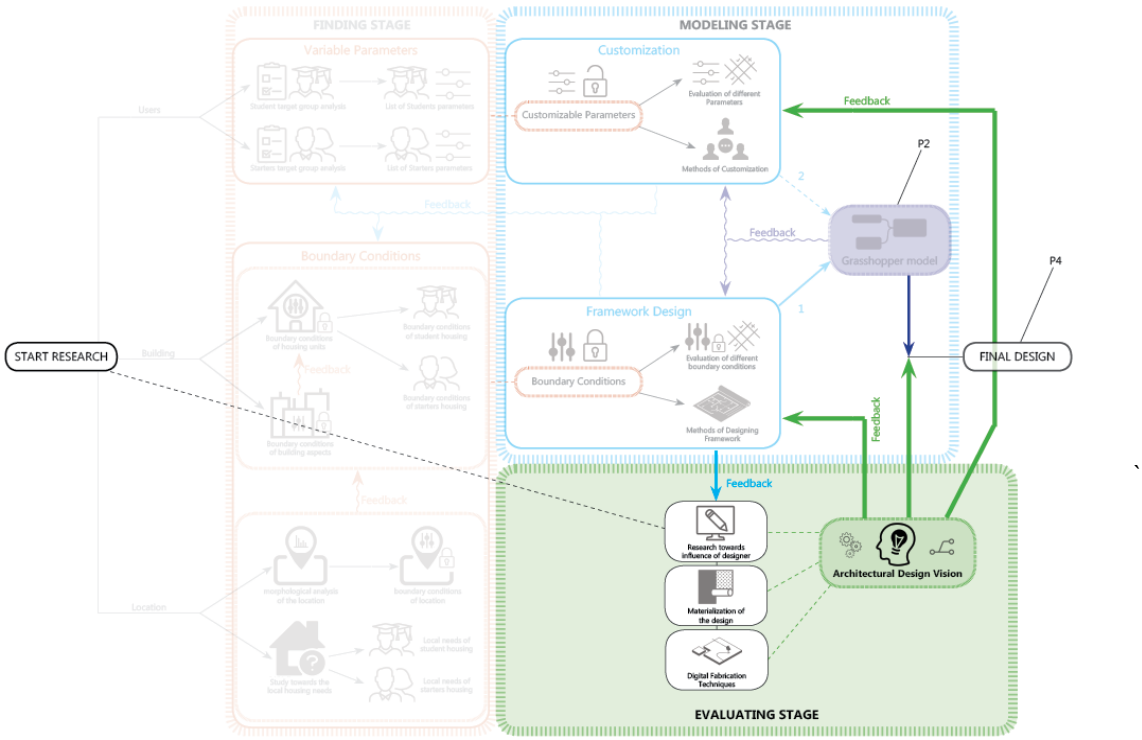


Illustration 6 – The Evaluating Stage

3. Results

In order to start the designing process with the help of Grasshopper, already some conditions and criteria need to be formed, in order to allow the necessary qualities (spatial optimization, spatial perception, low contextual volume impact and user quality in flexibility and customization) in high density housing. These conditions are important for this research, since they function as guiding principles for the Grasshopper models, in order to allow the designer to explore essential geometrical design variations in a flexible way within the early stages of the design process

The first paragraph of this paper looks into these conditions and criteria, that form the guiding principles for the to be created parametric models. After this, the structure of these parametric models will be explained and evaluated.

3.1 Conditions and Criteria

The first and, for the problem statement of this paper, most important condition, can be considered the amount of housing possible within the project site, since the subject of this paper is for a part about the design of high density housing. Therefore, it is important to allow the maximum amount of housing possible within the limited design space available, without losing the necessary qualities that are needed in order to create a successful high density housing project. This *quantity* versus *quality* balance is a key feature in the spatial optimization aspect of this project. Rudy Uytenhaak (2008) describes this as *Floor Space Index* (FSI) versus *Ground Space Index* (GSI). He explains that the FSI is a value that determines the spatial efficiency of a design by giving a value to the ratio of floor area versus ground area (or bebouwings- percentage * stapelingsfactor). Whilst the GSI determines the ratio between the footprint of a building versus the available terrain within the design location. By finding a balance between these two indexes (high FSI versus low GSI), the density aspect of the design could in a way be optimized focusing on a low contextual volume impact.

Besides this, also the amount of daylight is a condition / parameter that needs to be considered with creating high density housing. Daylight has a strong connection to spatial perception. The amount of light present in a room or space, determines in a way how we perceive this room or space. Besides this, daylight is also directly linked to human behavior. Daylight and darkness trigger the release of hormones in a person's brain. Sunlight itself increases the release of a hormone called serotonin. This hormone is associated with boosting a person's mood and helping him/her to feel calm and focused. A shortage of this hormone can lead to forms of depression (Nall, 2015). Therefore, daylight and sunlight can be directly associated with both spatial perception and user quality and satisfaction.

In his work *Steden vol Ruimte. Kwaliteiten van Dichtheid,* Rudy Uytenhaak also describes daylight as an important design aspect. Besides this, he also states that there are more criteria needed in order to create a qualitative high density design:

"(…) De dichtheid dwingt de architect tegelijkertijd om beter en dieper na te denken over uitzicht, daglicht, oriëntatie en privacy.

Ontwerpen wordt zo weer studeren en puzzelen en het zoeken naar de juiste balansen. De kwaliteit van dichtheid oftewel de dichtheid aan kwaliteit."

So, in order to create qualitative high density design; view, daylight, orientation and privacy all need to be considered (Uytenhaak, 2008).

View and daylight both contribute to the perception / experience of a space, but also to the user satisfaction of this space. Whilst orientation would also give structure to a design and its environment, according to the Adolf Loos principle (Uytenhaak, 2008), which states that the decoration and of space gives orientation and therefore also structure and readability to this space. Also privacy determines the quality in liveability of a high density design. Function / program and use of a design determine the relation between public and private domains. Mixed use would improve this relation and therefore also the liveability of the design (Kanteldenker, 2015).

This means that orientation and the relation between public and private domains both contribute to the spatial perception / experience of a design, but also the spatial optimization of this design. Although spatial perception / experience itself is hard to parametrize, since it is partially based on subjective judgement, some element that have effect on this spatial perception can be parametrized. For example, the layout, lighting, materialization and measurements of spaces within the building design, are all elements that affect the experience of these spaces. A program like Grasshopper can parametrize these more rationalized design elements, in order to allow testing with spatial perception and spatial optimization.

The various design aspects and conditions described within this paragraph will be suitable for forming clear guiding principles and testing criteria for the to be created parametric models, without adding reflexivity and unnecessary complexity. In the next paragraph the structure of the first Grasshopper model will be explained and evaluated, focusing on the second sub-question of this research paper.

3.2 First Grasshopper Model

By looking at the context and the location analysis (see appendix), already a global idea was formed of where the building would be created. In the first experiment with the Grasshopper model, this location analysis was directly translated into curves. One curve determining the boundary and shape of the building, the other curves determining where openings in the building should be (see illustration 7).

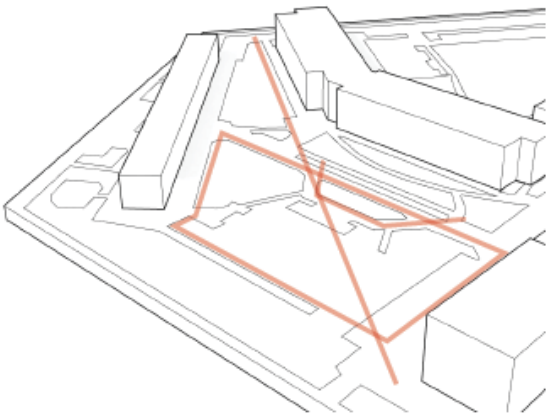


Illustration 7 – Starting curves of the first Grasshopper model

These curves are adjustable, meaning that the shape of the building would change according to these curves. From these curves floors and other building aspects could be created using sliders in Grasshopper. The process of the first Grasshopper model is shown in the

appendix of this paper. The model starts with the input parameter curves. In the second step the different floors are created from these curves. Also an open public space (patio) is created by offsetting the boundary curve. Next, a slope was created for the building, allowing more sunlight and more collective space for the housing units. However, the slope in this model is not based on any analysis. It is now based on a four points surface, which are pre-determined by the designers input (see illustration 8).

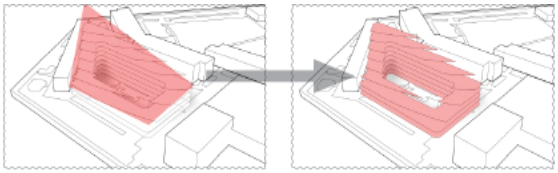


Illustration 8 – Creating the Slope in the first Grasshopper model

Based on the pre-determined curves, the different openings in the building where created in step four. In the last steps the skin / facade of the building was added, also a basic idea on internal routing through the building, created in curves.

The first Grasshopper model was a good start towards a parametric approach for high density housing. The curves and the sliders already allowed for a certain form freedom within the starting parameters. However, whilst the model already allows a certain kind of form freedom, it also is limited by the curves themselves. They (although adjustable) already determine the shape of the building from the beginning. This first Grasshopper model is already too much influenced by the prematurely defined choices of the designer and therefore it does not allow certain freedom in form exploration based on objective analysis.

Besides this, the model also does not give a good enough idea on the amount of housing possible within the building design. Since the research focusses on high density housing, having a global idea of the amount of housing possible would be ideal.

3.2.1 Evaluating the First Model

The term *parametric* originally comes from mathematics, but during the last few years there is a debate about its definition, since designers initially began using the word for parametric design. Parametric design is now defined as a process based on an algorithmic thinking process (or algorithmic logic). This process enables the expression of boundaries and rules (parameters) that encode and define the relationship between design input and its response (Jabi, 2013). This means that the designer could implement these boundaries and rules and based on these boundaries and rules the algorithmic process will create an output. This output can be just code language, but it can also be related to geometry.

This geometry could be changed and generated using input constraints and parameters, allowing the designer to create multiple design output possibilities within these constraints. This means that the designer is allowed to create multiple complex geometries and these geometries could then be researched within the design process to find possible problems. The solutions for these problems can be found by adjusting the input parameters, so that an optimal geometry could then be created. With the help of this parametric *form-finding*, the architect can create a certain kind of differentiation of elements within his

design, giving him more freedom in the composition of the design within the design process (Schumacher, 2011). Patrick Schumacher even sees the parametric design process (or *parametricism*) as a new but dominant style in the future of architectural design. He sees it as a style with many new possibilities and a lot of unexplored territory (Schumacher, 2008).

Tools like Grasshopper allow techniques like form-finding to be implemented in the design process. This means that with the help of computerized design, the architect can create differentiation in his design. This differentiation can be used for problem solving within the design process. Kas Oosterhuis (2014) refers to this as mass customization:

"The means of designing a parametric detail and producing it with unique parameters. Parameters supplied by the architect. A parametric detail is capable of incorporating different natures; one could think of it like a floor detail that is a façade detail with different parameters. In a sense parametric detailing is purely an architectural endeavour, trying to converge as many problems into one unifying solution."

If we relate this vision to the design process of the created Grasshopper model, we understand that the purpose of the different steps within this model, is to already tackle many problems that come with high density housing (such as its footprint, daylight, orientation & privacy) within one Grasshopper model. By adjusting the input parameters, a solution can then possibly be found, that accommodates all these problems. Form-finding within the design process would allow this possibility. This parametric form-finding has a lot of possibilities to create complex design, but it also has its dangers.

As discussed before, parametric codes are created with a certain algorithmic logic, a set of rules which will define the outcome of a design and this design will always be created within this rule based system. If the designer is not fully in control over the parametric design code he uses for his form-finding experiments, this code can also be limiting the creative freedom of the designer. A design created with this parametric code could then possibly lose its *human-touch* and the output of the design could create some sort of unwanted uniformity.

If we relate this to the created Grasshopper code, this code, for example, is limited to always creating an open patio and a sloping form within the created geometry, limiting the form-finding possibilities of the grasshopper design. In order to avoid this, the designer should always be in control over the algorithmic process itself.

This can be better explained by using the work of Elise van Dooren. In *'Making explicit in design education: generic elements in the design process.'* Elise van Dooren (2013) researches the architectural design process. She describes that the design process consists of a *Frame of Reference*. This Frame of Reference consists of certain domains, which, when narrowed down will eventually form the design. By switching between these different domains, and combining their results, the design can eventually be created. This is, however, a process which is not completely linear. It is a more iterative / trial-and-error process, which eventually narrows down the choices made within these domains (see illustration 9).

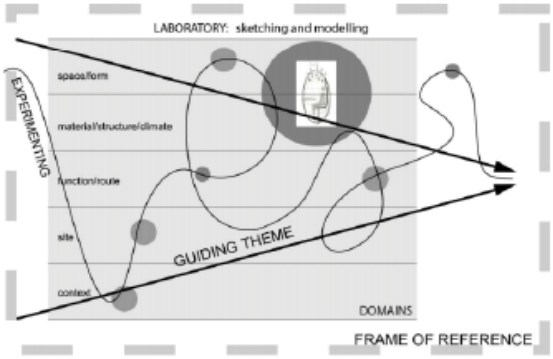


Illustration 9 – Frame of Reference, Elise van Dooren (2013)

The created Grasshopper model however forms a more linear process, in which every step within the created model is placed within one continuous code (see appendix). By adjusting the input parameters of the Grasshopper model, certain variation in form-finding is still possible within the created design. However, because of the continuous code, deviation between the different steps (domains within the grasshopper code) is no longer possible, without having to change large parts of the created Grasshopper code, which can be a complex and time consuming process. This means that the code itself is limiting the freedom in form-finding of the designer.

The relation between Grasshopper and the designer needs to be focused on this form-finding aspect. Therefore changing the role of the designer from a physical designer to a more virtual designer, a programmer, who is in control of all the aspects of parametric design (input, algorithmic process and output). Being in control over the different steps within the algorithmic process of the Grasshopper code is an important starting point for improving the first Grasshopper model.

3.3 Second GH Model

After learning more about the process of the first Grasshopper model, a second Grasshopper model was created (an attempt on improving the first model).

Instead of pre-determining the shape of the building through curves, instead its boundaries are more loosely constraint by a larger area within the design location. Instead of determining the shape of the building with curves, the building will now be shaped within these larger boundaries, using different techniques within Grasshopper. Some elements of the design are still determined with variable sliders (for example, the amount of floors and building height), allowing the necessary flexibility within the starting parameters of the design. Also, the size of the housing units are now determined in the starting parameters. The new model will now be created out of these housing units. Therefore, the new Grasshopper model will give a better idea on the total amount of housing units possible within the building constraints.

3.3.1 The Voxel Strategy

The definition of a voxel is a representation of a value on a regular grid in a three-dimensional space (Foley et al., 1995) In simpler terms it means that a voxel basically is a 3D pixel. It has dimensions, not only in the x and y direction, but also in the z directions, making it a cube or cuboid. A monitor screen has a certain amount of pixels in the x and in the y direction, creating a two-dimensional grid. Voxels are also repeated in the z direction, creating a three-dimensional grid.

By using voxels in the new Grasshopper model, a three-dimensional grid can be created on the design location. By giving each voxel the dimensions of a singular housing unit, a better representation can be given of the maximum amount of housing units available in a limited space. The dimensions of these voxels are translated into variable parameters in this new Grasshopper model. This way, different design iterations can be created with different sized housing units. However, by changing these parameters, it changes the dimensions of all housing units, not individual units.

By removing the voxels that do not allow the necessary quality for the design itself, a certain representation can be created of the qualitative high density architecture that is needed for this design project. Only the housing units remain that allow this needed quality. This way, the balance between the *quantity* of high density housing versus the *quality* needed for the architectural design itself can be better researched.

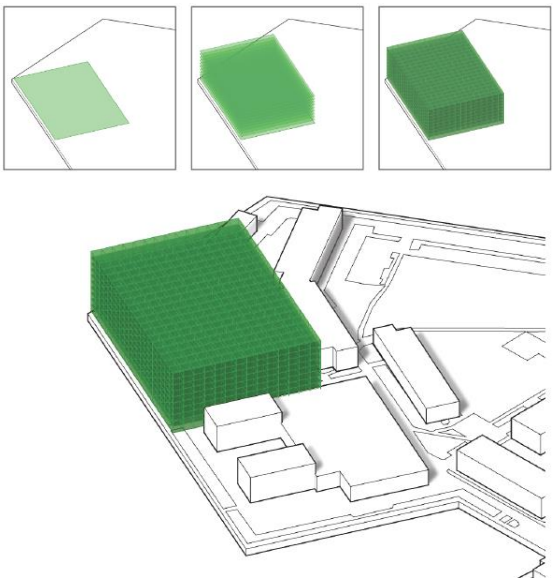


Illustration 10 – Voxel strategy on the design location.

3.3.2 Gradient Mapping Strategy

After the voxel strategy was implemented into the second Grasshopper model, a strategy was created for removing certain voxels (or housing units), in order to allow the necessary qualities for high density housing, mentioned in the research question. In order to do this, a gradient mapping strategy was used. This strategy means that a two-dimensional, black and white gradient image is used in order to determine which voxels need to be removed within the design location. This gradient image could possibly be more directly based on the different location analysis already done by the designer.

The black and white gradient illustration is placed over the ground level of the design location. The gradient from black to white within this illustration determines the height of a point based cutting surface created by this image.

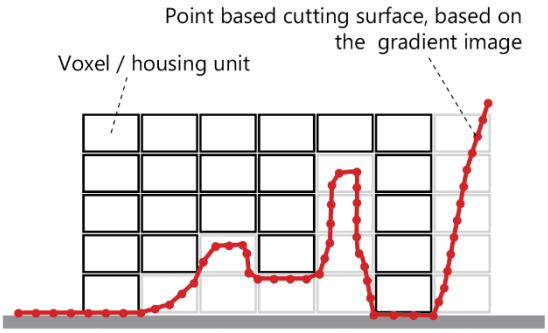


Illustration 11 – Point based cutting surface

If the gradient is completely white, all housing units within this white area are removed, and if the gradient is completely black, all housing units within this area are allowed. This way openings can be created in the building, where for example, the existing morphology of the area is more important for the quality of the location, and housing units can exist where this existing

morphology is less important. If we compare this gradient mapping strategy to the curve strategy of the previous model, we can see that the goal of each strategy is still the same in a way. Both strategies determine rules and constraints on where the design can exist on the project site and where is cannot exist.

The gradient mapping strategy is an attempt on making the input parameters closer related to the location analysis already done by the designer. The gradient image used as input for this stage within the Grasshopper model, could in the future be a conclusive location analysis image. This way, it would skip the step of translating the results of the location analysis into curves within Rhinoceros, bringing the input parameters closer to the ideas and vision of the designer.

However, In the end this gradient mapping strategy worked counter-productive and against the form-finding possibilities of a parametric approach. This problem will be further elaborated in paragraph 3.3.5, when this Grasshopper model will be evaluated.

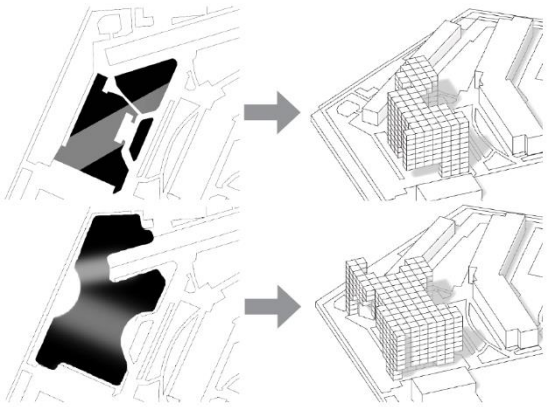


Illustration 12 – Examples of gradient mapping to three-dimensional design

3.3.3 Using Solar Analysis

In the beginning of this chapter, the importance of daylight and sunlight as design conditions for designing high density housing, was already mentioned. Daylight and sunlight not only contribute to spatial perception, they also contribute to user quality and user satisfaction (see paragraph 3.1). Therefore, it is necessary that a form of solar / daylight analysis is implemented into the Grasshopper model. By doing this, the Grasshopper model would be more informative and useful for the design process of designing high density housing.

In the first Grasshopper model a slope was created in the building design to allow sunlight in most housing units. This slope was based on a simple cutting surface, which was loosely based on the direction of the sun. However, in this first model the surface was not yet influenced by the sun itself. Instead it was created by several points, which locations were all determined by the designer himself. This meant that the first Grasshopper model was a rather uninformative model. The slope of the design was already too much based on the vision of the designer, making an iterative research process with solar analysis impossible within the design process.

In the second Grasshopper model, a form of solar analysis was added to the Grasshopper code by using ladybug (Roudsari, 2016), a plugin for Grasshopper. This plugin can be used to create a virtual sun path (see illustration 13) within rhino based on meteorological data of different locations in the world, that can be found on the internet (EnergyPlus, 2016).

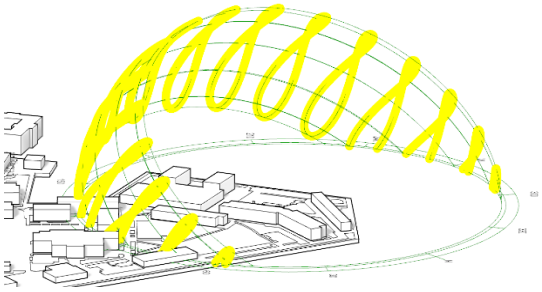


Illustration 13 – Creating a sun path in Rhino 5 using Ladybug

This virtual sun path was used to analyze the amount of hours, each housing unit / voxel is getting direct daylight during a year. By removing the housing units that do not get enough daylight during a year, only the housing units remain which have enough quality, based on both the gradient mapping and the sunlight analysis. The minimum amount of sunlight during a year (which determined the unwanted the housing units) is variable in the current Grasshopper model. This way, different researches can be done, using the solar analysis, allowing multiple design variants to be created within the design process.

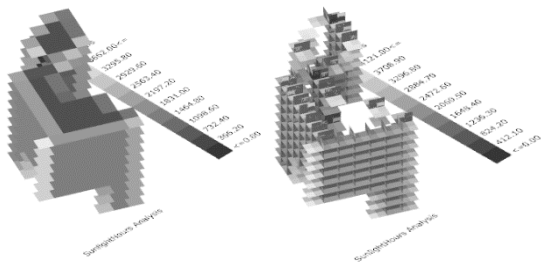


Illustration 14 – Solar analysis of the second Grasshopper design

In order to separate the daylight analysis between each housing unit, separating walls between the housing units are created. However, the walls separating the housing units from the outside, the façades, are removed within this Grasshopper model. This way, the outer façade does not block the daylight for the individual units, but the separating walls do. A more accurate daylight analysis can now be derived from this Grasshopper model.

3.3.4 Vision on Internal Space

After using both gradient mapping and solar analysis, a certain amount of possible housing units remain within the Grasshopper model. Because of the gradient mapping and the solar analysis a lot of open space is created in between the different housing units / voxels. This open space that is now created can possibly be used for internal public space and routing within the building design, connecting the different housing units and allowing spatial quality in spatial distribution, spatial optimization and spatial perception.

In the first Grasshopper model, already a basic idea is created for the internal routing of the design, using curves. However, since there is no exact location, scale and distribution formed for the individual housing units in this Grasshopper model, these curves are uninformative.

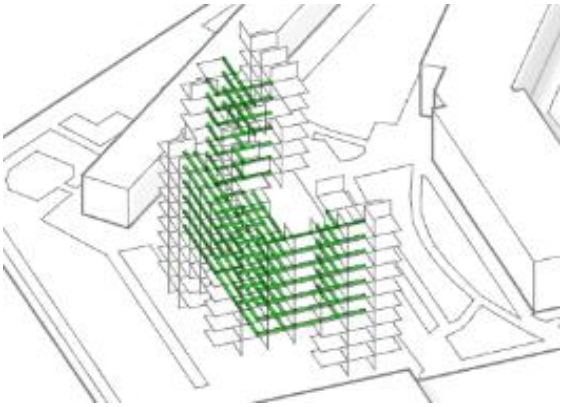


Illustration 15 – Ideas for routing in the second Grasshopper design.

In the second Grasshopper model, there already is a better vision created for the individual housing units. Because of this, different connections between the housing units can be created and translated into a basic vision for routing and internal public space within the building

design (see illustration 15). Horizontal routing and vertical routing are visualized using the Grasshopper model. However, they are still visualized as curves in this model. By combining this routing with a more conceptual idea for public and recreational spaces within the building, an architectural vision for the internal space of the building can possibly be created in the future of this design process, using the output of the Grasshopper model as helping guidelines.

3.3.5 Evaluating the Process

If we again look at the work of Elise van Dooren (2013), *'Making explicit in design education: generic elements in the design process.',* the design process is described as mostly implicit. In this process the designer has to become aware of certain aspects of his design in order to make them explicit. This process of making the implicit explicit can be considered to be an iterative process, in which questioning the design choices and reflecting on these choices are key aspects in order to allow this iterative process and therefore to be able to further develop the design (see illustration 16).

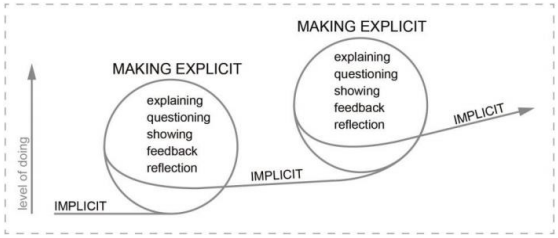


Illustration 16 – Making the implicit explicit, Elise van Dooren (2013)

To be able to further develop the design of high density housing, this iterative process also needs to be part of the design process.

By reflecting on the different steps within the first Grasshopper model, still problematic aspect of this model became explicit. By learning from these aspects, a second and more improved Grasshopper model could be made. This new model revised certain aspects of the first model, but new possible problematic aspect of the Grasshopper model also come to light if we evaluate the second Grasshopper model.

Whilst the goal of the gradient mapping strategy, used in the second Grasshopper model, is to more directly relate the input parameters of the model to the vision of the designer on the design location, in the end this instead worked counter-productive. One of the strengths of a parametric approach is parametric form-finding. With the help of form-finding, a designer can create a certain kind of differentiation within his design process, giving him more freedom in the composition of the design within the design process (Schumacher, 2011).

By basing the input parameters of the second Grasshopper model on a created gradient image, the input parameters become in a way fixed within this model. Only by changing the gradient image (which can only be done outside of Grasshopper), the design output of Grasshopper will change, excluding the malleability of the existing geometry within the Grasshopper model; and therefore excluding the possibilities of form-finding within this Grasshopper model.

The Grasshopper model itself now becomes in a way uninformative, leaving the form-finding aspect of the design process to the creativity of the designer. This can be compared to form-finding using sketching or making a physical model.

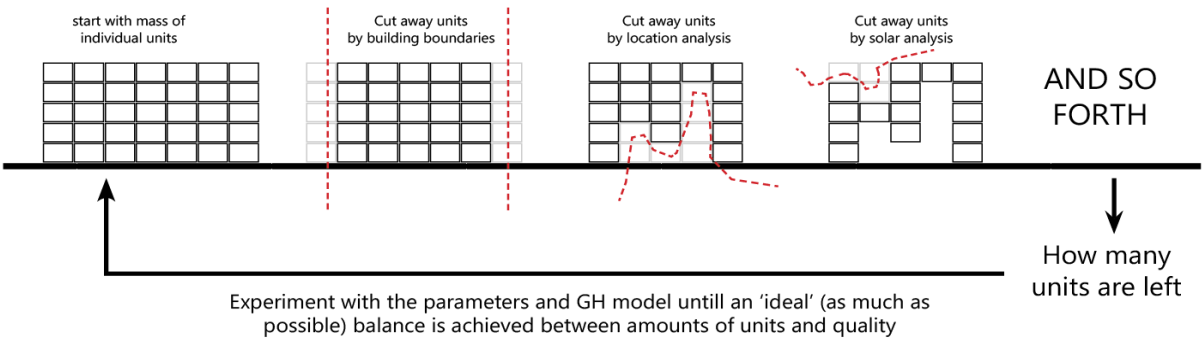


Illustration 17 – The Process of the second Grasshopper model

This means that the possibilities of the computer within the design process are not exploited within the new Grasshopper model. In order to fully exploit the possibilities of a parametric approach, this needs to be improved for the future of this parametric design project.

By reflecting on the second Grasshopper model, the iteration of making the implicit explicit can again be repeated in order to improve the design. The iterative process, as described earlier, in this parametric design approach, can be considered to be very similar to a non-parametric design approach. Reflecting on and improving certain problematic aspects of a design can be a very tedious and a relatively long process in both a parametric and a non-parametric design approach. However, because of the flexibility and variability within a parametric model, optimizing the design itself could be a less complicated process compared to a non-parametric design approach.

A non-parametric design approach usually uses more intuitive and more approachable design methods. This means that, only by knowing the possibilities and understanding the *language* of the parametric design tools themselves, the iterative design process could be considered to be viable in a parametric design approach.

Using a parametric design approach and, therefore, combining the mathematical strength of the computer with the intuition and creativity of the architect, could result in a more efficient design process. This cooperation could even be used to further rationalize the design process itself, in order to make this process more understandable.

3.4 Rationalization of the Design Process

During the 1960's and 1970's several studies attempt to contemplate the design process as a fully rationalized process by excluding all forms of intuition and personal creativity. The design methods, in these cases, exist of a more precise description of the design problem, allowing an optimal solution to be directly derived from the analysis of the problem (Cross, 2007).

However, if we look deeper into the design process, the existence of one optimal solution does not seem to be a realistic scenario. A design process is very complex and usually exists of multiple problems, with multiple solutions. Some solutions can only be made at the expense of other solutions for other problems. In order to create a good design, there needs

to be a good balance found between these different design solutions. The iterative trial-and-error process of finding these design solutions and finding the balance between these design solutions could be considered as the definition of a design process.

By rationalizing the different design problems, the process of finding the different design solutions, and the process of finding the balance between these solutions; the complex thinking process of designing and its nature would most likely become a more understandable process. This would mean that the process of making the implicit explicit would become more approachable and understandable for designers and architects. However, multiple studies describe that certain design aspects and certain design problems within a design process cannot be fully rationalized.

3.4.1 Ill-Structured Problems

In *'The Structure of Ill-Structured Problems'* Herbert Simon (1973) distinguishes two kinds of problems in a design process: *well-structured p*roblems and *ill-structured p*roblems. Well-structured problems are problems like described in the previous paragraph. These problems have a more precise description and the solutions can usually be derived from analysis. For example, we can look at the second Grasshopper model, where analysis can directly be translated into geometry using gradient mapping. For these kind of problems, there are in most cases methods available to compare the different solutions to each other, in order to find the best solution.

Ill-structured problems are problems that do not fit the criteria of a well-structured problem, because their criteria is almost impossible to write down. Spatial perception, for example, can be considered to be ill-structured, since its criteria is partially based on intuition and subjective judgement by the designer and therefore impossible to write down.

Ill-structured problems exist, when the designer needs to be creative in order to solve these problems. It is not possible to solve these problems with standardized solutions (Simon, 1973). Also, there is no testing criteria on which different design solutions can be tested and compared. Besides this, Herbert Simon also explains that it is almost impossible to narrow down the direction in which to find the solution, since all the aspects of the design (space, material, structure etc.) need to be considered in order to find the solution itself.

In '*How designers think – The design process demystified'* Bryan Lawson (1980) distinguishes three different, but important aspects in the thinking process of an architect or designer: *design problems*, *design solutions* and the *design process*. In his work Lawson states that design problems cannot be comprehensively stated, because certain aspect of these design problems can only be described after the design itself is done and other aspects of these problems cannot even be described at all. He states that design process itself is dynamic, meaning that the process is always interpreted subjectively by the architect.

Because of this, there are an inexhaustible number of different design solutions. This would make it impossible to find a clear direction in which to find the

different design solutions. This is directly in contrast to mathematical problems, which usually have a certain goal / solution for the researcher to work towards. Both Herbert Simon (1973), as well as Bryan Lawson (1980) describe the design process in such a way that it is impossible to rationalize this process. The design process relies too much on intuition and subjective judgement as guiding principles, making it impossible to write down clear design problems and, therefore, also making it impossible to directly derive its solutions.

If this is related to the parametric approach of this paper and the associated created Grasshopper models we learn that the disadvantages of this approach can be found within these ill-structured problems. The dangers of a parametric approach can be found in the logical need for rationalization in order to use the strengths of a parametric approach (which are the many possibilities in form-finding and optimization). However this need for rationalization can also lead to reflexive design criteria and unnecessary complexity. The architectural design could then be reduced to a meaningless cast without any social relevance. This is often referred to as *computational decoration* (Shea et al., 2005). In order to avoid this, it is necessary to understand the need for ill-structured problems within the design process. Therefore it is necessary to understand the importance of social and contextual relevance as concept or guiding theme within a parametric design process.

According to Janssen (2006), the traditional architectural design process consists of four stages: analysis, synthesis, evaluation and implementation. Each stage would then give information for the next stage, much like the different steps within the created Grasshopper models. However, as discussed in this paragraph it is impossible to write down all aspects of certain design problems and therefore it is impossible to derive their solutions. The traditional design process as described by Janssen (2006) now appears to be invalid, since the analysis stage would not give enough information to continue to the synthesis stage; there is a *missing link* needed between these two stages. Bryan Lawson (1980) describes this missing link as:

"As an activity, architectural design relies heavily on intuition, preconceptions, heuristics and guiding principles."

He states that in order to continue to the next stage certain preconceptions or heuristics are needed. If we relate this to the created Grasshopper models, an example of such a preconception is the choice to use a voxel strategy in order to get an idea on the amount of housing possible. There are a lot of other possible methods that could also give an idea about the amount of housing possible. However, the voxel method was chosen in this case. In order to answer the various well-structured and ill-structured problems, that come with designing high density housing, a preconception or heuristic choice was made in choosing the voxel strategy, in order for the project to be able to continue.

3.4.2 Heuristic Design Process

The previous paragraph shows us that the design process can be described as an heuristic process, a process of trial-and-error in order to eventually answer the large variety of design problems. Rowe (1982) states that the design process is inherent heuristic. The architect namely bases his intuition and design choices on

prior knowledge. This prior knowledge could for example exist of various rules of thumb, all influencing the final outcome of the design. Because of this, it is possible for the designer or architect to determine several conditions for his design. The design itself is finished when it meets these conditions. For example, in the case of this project, these conditions are written down in paragraph 3.1.

The design process, in this case, would be a process where prior knowledge would be evaluated, but it would also be combined with new knowledge. If this is the case, it means that the end criteria for a project would be clear enough to write down, even if the design problems themselves are not. The designer now has a certain goal to work towards in his design process. If these end goals were to be strictly described, it would mean that the design process indeed could be rationalized in a way.

However, This rationalization would be a more dynamic rationalization, since new end goals and new design problems could appear during the design process itself. These would then have to be evaluated and combined with the already existing criteria. This means that the further a design process advances, more conditions become clear and therefore, the better the design process could be rationalized by the designer or architect. However, this can only be done by always reflecting back on the different design stages, much like the arrows show us in the strategy scheme of the methods chapter of this paper (chapter 2). It would mean that after reflecting, certain conditions would need to change. The parametric model would need to be flexible enough in order to accommodate these changes.

3.5 Computerization in the Design Process

The rise of computer technology started a new interest in researching and understanding the design process. Since the introduction of computer technology, the main interest in the design process is now shifting towards the computer as a helping and supporting, or even stimulating tool in the design process (Pauwels et al., 2011).

In this process the computer and the architect work together in a more cooperative context. The creative and intuitive role of the architect will in this process be combined with the fast and powerful analytical role of the computer. This mathematical strength of the computer could possibly result in a more efficient and more rational architectural designing process (Coons, 1963).

The analytical power of the computer enables the designer to quickly create a large variety of different design solutions for different design problems. The designer is then able to compare these different design solutions on different aspects in order to optimize his final architectural design. This form of *Computer-Aided Design* (CAD), allows the architect to quickly answer the well-structured problems, of which the problems are clear enough that different design solutions can be found and compared to each other. However, it could not directly give the designer an answer for ill-structured problems.

3.5.1 Heuristic Software

In 1976 Newell and Simon introduce the '*Heuristic Search Hypothesis´*, which is an algorithm designed to find solutions for more complex problems in an iterative manner. The algorithm consists of process of steps, where each step is analyzed on how much positive or negative influence it would have on the final outcome of the design. This information is then passed forwards to the next step (Newell & Simon, 1976).

This means that the algorithm is *learning* from its prior steps. Therefore, prior knowledge could be evaluated and also combined with new knowledge, much like the heuristic design process already described in the previous part by Rowe (see paragraph 3.4). This computer-aided design process is a selective process, where, with the use of smart selection, design solutions can be found in a relative short period of time.

Perhaps an heuristic computer program could be designed in such a way, that it can rewrite itself according to what it has learned. In this case the design solutions are not a clear goal to work towards. Instead, they reveal themselves during the design process, as a result of a selective sequence of steps (Littlefield, 2008). In this case the design solutions would evolve as new design problems and new design solutions would appear during the course of a design process. Now, the algorithmic process of the computer could be considered to be a more dynamic process of rationalization. The further the design process advances, more conditions become clear and therefore, the better the design process could be rationalized by the computer.

The heuristic design process of the architect could, in this case, be combined with the heuristic algorithm of the computer, resulting in a more cooperative design process. The computer would be able to support the iterative trial-and-error process that is needed in order to make the implicit explicit within the design process. However, it could not fully do this iterative process by itself. Since, for certain design problems, there are an inexhaustible number of different design solutions, like Lawson (1980) already stated in the previous part (see paragraph 3.4).

There are no optimal solutions for design problems, making the algorithmic process of the computer to be an endless process, in which the subjective judgement of the architect himself becomes a very important factor for evaluating different design solutions. In order to create a good design, there needs to be a good balance found between the different design solutions, during the design process. Because of this, the computer would not be able to make certain choices in the design process by itself, since these choices can only be made by the intuition and subjective judgement of the architect or designer.

Understanding the need for this cooperation between the computer and the designer within the design process, the advantages of a parametric design approach also become clear. By combining the effort of both the computer and the designer; Also, understanding the limits of the computer in solving the ill-structured problems; We now know, where in the design process the strength can be found in the parametric program, or whether this strength can be found in the intuition and subjective judgement of the designer.

3.5.2 Programming Languages

To create computerized and parametric design, there are multiple tools available for the computer. These tools each have their own advantages to help the designer achieve his design goals and help the designer or architect help them achieve these goals during the design process itself. Besides the advantages of these tools and programs, there are also disadvantages. These disadvantages mostly lie in the boundaries and limitations of each tool and program, but also in the way people use these tools for their designing purposes. In order to research these design tools and programs, we also need to look at the basic logic behind these tools. Programming languages are the usually the foundation of most designing programs. Therefore this paragraph will focus on the logic behind those programming languages.

Programming languages are constructed languages which are designed to communicate instructions to a computer or machine. These constructed languages can be used to create different tools and programs which control the behavior of a computer or machine to express algorithms (Aaby, 2004). There are many programming languages available. Most of them are designed from scratch, partially or completely altered, or even combined with other languages to meet the desires and needs of their users. A lot of programming languages are not being used anymore, because they were not able to fit those desires and needs. Still, there are now a lot of programming languages available to help their users with creating their desired tool, program or even design itself. Each programming language is designed to fit certain needs on certain fronts, but also inevitably has its disadvantages at other fronts. There have been attempts to design one universal programming language that fits all needs and desires of its users. However all of these attempts have failed to fit all these needs and to be generally accepted as the universal programming language (Lévénez, 2011).

However, this does not mean that there are no good programming languages out there. A lot of programming languages do fit most of the needs and desires of their users and therefore are extensively used. A good example of such a programming language is Python. This programming language is a good example of a programming language that is suitable for the foundation of a parametric design tool. Python is a general-purpose programming language. A general-purpose programming language means that it is a language capable of creating and writing a wide variety of programs and applications (Foundation PS, 2001). Python supports multiple programming paradigms, such as imperative and functional programming, but also *Object Oriented Programming* (OOP). ‘

How other programming languages are organized around actions and the logic behind those actions, an OOP is organized around objects and their data. Usually a program could been viewed as a logical procedure that takes input data in one action, processes that data in a second action, and in a last action it produces output data. In languages that support OOP an *object* can be seen as a structure that contains both data and procedures (Pierce, 2002). By changing the data of an object, procedures will automatically follow, changing the outcome data of the object.

An OOP like Python therefore is a suitable program to create and generate parametric geometry. By changing the input parameters of the geometry, the geometry will automatically process those parameters and change the form and shape of the geometry according to those given parameters. Python scripting is therefore widely used as a plug-in or feature in 3D-modeling programs, like Maya, 3ds-Max or Rhinoceros and Grasshopper (Foundation PS, 2001).

Still, Object Oriented Programming also has its disadvantages. As a designer who uses OOP, it is easy to let the fast calculating program be in control over the often complex procedures that eventually create parametric geometry. This makes it easier and faster for the designer to create parametric design. However, these tools also have their limitations and shortcomings. These limitations and shortcomings are often overlooked by the designer and could influence the output of the designing process towards a less favorable design. This less desirable design could have lost its *human touch* and possibly have the same unwanted uniformity like discussed before.

3.5.3 Software Development

A software development methodology can be described as the partitioning of software development into distinct stages, each containing certain necessary activities for the development of the to be created software or tool. The intent for this partitioning is to better manage the design or development process of this tool (CMS, 2008). Common software development methodologies are Waterfall Development, Rapid Application Development, Prototyping or Agile Software Development. Waterfall Development consists of a more sequential process of different phases within the design process. There is some feedback between different phases possible, however, this method is mostly linear (CMS, 2008).

Rapid Application Development (or RAD) is a methodology which focuses on the iterative rapid construction of software testing iterations, instead of large amounts of up-front planning. The lack of extensive pre-planning generally allows the software to be written much faster. However, during the design process it could lead to extensive revising of new and unforeseen problems within the iterations (Geoffrey, 2004).

Prototyping, focuses on creating software prototypes (incomplete parts of the software program being developed). Some of the basic principles of prototyping are: Attempts to reduce inherent project risk by breaking a project into smaller segments and therefore providing more ease-of-change during the design process; and, the client is involved throughout the development process, which increases the likelihood of client acceptance of the final implementation (Geoffrey, 2004). Prototyping is therefore closely related to Agile Software Development

Agile software development can be described as a methodology for the creative process that already anticipates the need for flexibility and adaptability within the delivery of the finished product (Agile Alliance, 2013). Agile software therefore focuses on keeping its code simple, in order to allow for iterative testing during the software development process. This way small functional bits of the

application can be reflected on as soon as they are ready. The goal of agile software development is to create small (client-approved) parts, which can be built upon as the design process progresses. This, in contrast to creating and delivering one large program at the end of the design process. Which, after reflecting on it, could still have a lot of unideal aspects. Improving these aspects would then be almost impossible, since the code itself would already be too complicated. Smaller parts of code are easier to alter and improve. Also, they are more graspable and understandable.

Agile software is not a new phenomenon in software development, but it is mentioned in this paper, because of its advantages in a computerized design process. This means that the development of a parametric Grasshopper model could easily be compared to agile software development. By understanding the methodology of using agile software development, it can also be applied to the development of a Grasshopper model, in order to structure it and improve its process.

By creating smaller parts of Grasshopper code and reflecting on them / building upon them, the final Grasshopper code can be better structured and choices within the design process will become more focused, more logical and well-founded. Also, this allows the parametric design process to shift between different design domains and make alterations within these domains, without having to change large parts of the Grasshopper code. This way the design process could be fully exploited and the designer can always be in control over the parametric code he created, limiting the disadvantages of a parametric approach.

4. Conclusion and Discussion

The architectural design process is a complicated process, consisting of a complex thinking process, that of which its nature and exact operation is not clear. This paper looked into this design process in the context of improving the design of high density housing, using a parametric approach.

4.1 Conclusion

The subject of this paper looks into designing high density residential architecture that enables certain qualities, in order to improve the livability and perception of the design. a parametric design approach was chosen for this project in order to enable these qualities. In this paper, not only the resulting parametric Grasshopper models were researched, but also the design process which eventually formed these models.

First, several important conditions and criteria for this project were formed. These form guiding principles for the parametric models. This is important, since parametric modeling software can often lead to reflexive design criteria and unnecessary complexity. In order to avoid this, relation to context and social relevance are becoming testing criteria for the parametric models, using these guiding principles. *Quantity* versus *quality* balance is a key feature in the spatial optimization and density aspect of this project By finding a balance between two indexes, *Floor Space Index* (FSI) versus *Ground Space Index* (GSI) (high FSI versus low GSI), the

density aspect of the design could in a way be optimized focusing on a low contextual volume impact. view, daylight, orientation and privacy all contribute to the spatial perception / experience and user satisfaction of a design, but also the spatial optimization of this design.

Although spatial perception / experience itself is hard to parametrize, since it is partially based on subjective judgement, some element that have effect on this spatial perception can be parametrized. For example, the layout, lighting, materialization and measurements of spaces within the building design, are all elements that affect the experience of these spaces. A program like Grasshopper can parametrize these more rationalized design elements, in order to allow testing with spatial perception and spatial optimization.

The first Grasshopper model was then created. By switching between different domains (or design aspects) within a design process, and combining their results, a qualitative design can eventually be created. This is a process which is not completely linear. It is a more iterative / trial-and-error process, which eventually narrows down the choices made within these domains. The first Grasshopper model, however, forms a more linear process, in which every step within the created model is placed within one continuous code. By adjusting the input parameters of the Grasshopper model, certain variation is still possible within the created design. However, because of the continuous code, deviation between the different steps (domains within the grasshopper code) is no longer possible, without having to change large parts of the created Grasshopper code.

A second Grasshopper model was then created (an attempt on improving the first Grasshopper model). This model showed promise by allowing better differentiation between de different design domains. However, by basing the input parameters of the second Grasshopper model on a gradient image, the input parameters became in a way fixed within this model. Only by changing the gradient image (which can only be done outside of Grasshopper), the design output of Grasshopper can change, excluding the malleability of the existing geometry within the Grasshopper model; and therefore excluding the possibilities of form-finding within this Grasshopper model.

With the help of parametric *form-finding*, the architect can create a certain kind of differentiation of elements within his design, giving him more freedom in the composition of the design within the design process. This parametric form-finding has a lot of possibilities to create complex design, but it also has its dangers. Parametric codes are created with a certain algorithmic logic, a set of rules which will define the outcome of a design. This, however, can be limiting the creative freedom of the designer and a design could then possibly lose its *human-touch.* Therefore, the designer should always be in control over the parametric code he created, changing the role of the designer from a physical designer to a more virtual designer, a programmer, who is in control of all the aspects of parametric design (input, algorithmic process and output).

By reflecting on the different Grasshopper models, the process of making the implicit explicit can be repeated within each the reflection of each model. By reflecting on each model, in order to understand the strengths and weaknesses,

the complexity of the design process of designing high density housing itself could be made more understandable and rationalized.

However, according to certain Literature (Simon 1973; Lawson 1980) the design process can never be fully rationalized. This is because the design process can also be described as ill-structured. Certain design problems cannot be described and therefore design solutions cannot be derived from these problems in an analytical and rational process. This means, there are an inexhaustible number of different design solutions, making it impossible to find a clear direction in which to find the design solutions. Because of this, the computer by itself would not be able to fully rationalize the architectural design process.

Understanding the advantages of the computer in well-structured problems and its disadvantages in ill-structured problems, is also important for understanding the need for heuristics within the parametric design approach. Rowe (1982) states that the architectural design process is inherent heuristic. The architect namely bases his intuition and design choices on prior knowledge. Lawson (1980) describes this as the need for preconceptions within the design process, in order to make certain distinct choices, of which there are a lot of possible options available. These choices have to be made in order for the design process to continue and in order to later be able to reflect on these choices. This means that the further a design process advances, more conditions become clear and therefore, the better the design process could be rationalized by the designer or architect. This more dynamic rationalization can only happen if we reflect on the heuristic choices we make, after we made them, in order to learn from them; and to possibly derive new design problems and solutions from them.

Within the design and development of the parametric tool itself, this would mean that perhaps a parametric code / program could be designed in such a way, that it can rewrite itself according to what it has learned. In this case the design solutions would evolve as new design problems and new design solutions would appear during the course of a design process. The heuristic design process of the architect could, in this case, be combined with the heuristic algorithm of the computer, resulting in a more cooperative design process. The computer would be able to support the iterative trial-and-error process of the architect in order to optimize certain design aspects.

In languages that support *Object Oriented Programming* (OOP) an *object* can be seen as a structure that contains both data and procedures. By changing the data of an object, procedures will automatically follow, changing the outcome data of the object. Understanding these procedures will enable the designer to reflect on the input and output data, making a cooperative and iterative trial-and-error process possible between the computer and the designer. By using a design methodology like Agile Development, it would allow the parametric design process to shift between different design domains and make alterations within these domains, without having to change large parts of the Grasshopper code. This way the design process could be fully exploited and the designer can always be in control over the parametric code he created, limiting the disadvantages of a parametric approach.

4.2 Discussion

By reflecting on the results of this research paper, an interpretation of the validity of these results can be formed. Instead of purely focusing on the structure and working of each Grasshopper model, the strength of this paper can be found in focusing on the design process of creating these models, and the critical reflection on this design process.

However, for this paper this means that there is no research done into the exact data and information given by the Grasshopper models. For the future of this project, this information, for example, could be used for making important choices with integrating the necessary qualities in the design of affordable and high density housing. This is not the case within this research paper.

Instead, this paper focusses on the design process of a parametric approach itself. It is much more important for the continuation of this project, to understand the structure of this design process, and the possibilities within this design process. By understanding this process, more insight is given, not in which choices to make, but in *how* to make these choices during the progression of this project. By understanding the balance between Grasshopper and the need for intuition and subjective judgement, the question of *which* choices to make will follow automatically.

Therefore, the limitations of this research paper can be found in *which* choices to make during the design process of this project, but the strengths of this research paper can be found in *how* to make these choices.

4.2.1 The Continuation of this Project

The results of this research paper allow us to better understand on how to continue this graduation project. By further narrowing down the different conditions and criteria needed for this project, and sub-dividing them between *well-structured* and *ill-structured*, we can get a better grip on which conditions can be integrated into the design, using Grasshopper. Also, which conditions need a form of intuition and subjective judgement by the designer in order to do this.

Some criteria would generally be considered to be ill-structured, but certain elements of these criteria could still be rationalized. This way Grasshopper could still be used in a way to process this criteria. For example, spatial perception is very subjective and could therefore be considered to be ill-structured. But, elements that influence spatial perception, like the dimensions or the lighting within a space. These are all elements that can be rationalized and used in a Grasshopper model.

Also, Shifting the focus of the to be created Grasshopper models, from larger models (trying to solve multiple problems in one model) to smaller models (in which the focus in more on solving one condition or design element), the different design problems that need to be faced during this project will be more graspable. Using this methodology of creating smaller and more focused parts of Grasshopper code, and reflecting on them / building upon them, choices within this design process will be more logical and well-founded.

5. Literature

Aaby, A. (2004). Introduction to Programming Languages. Retrieved from: http://www.emu.edu.tr/aelci/Courses/D-31 8/ D-318-Files/plbook/intro.htm.

Agile Alliance. (2013, June). What is Agile Software Development. Retrieved from https://www.agilealliance.org/agile101/

Autodesk. (2015). Company Autodesk. Retrieved from: http://usa.autodesk.com/ company/

Bentley. (2017). Generative Components. Design and explore the unimaginable. Retrieved from https://www.bentley.com/ en/products/product-line/modeling-and-vis ualization-software/generativecomponents

Caldas, L. (2007). An evolution-based generative design system. Advanced Engineering. Informatics 22, pg. 59-70.

CMS. (2008). Selecting a Development Approach. Centers for Medicare & Medicaid Services (CMS) Office of Information Service. Retrieved from https://www.cms.gov/ Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/ SelectingDevelopmentApproach.pdf

Coons, S. (1963). An outline of the requirements for a computer-aided design system. In AFIPS Spring Joint Computer Conference, pg. 300-310.

Cross, N. (2007). Forty years of design research. Design Studies 28(1), pg. 2-5.

Dooren, E. van, et al. (2013). Making explicit in design education: generic elements in the design process. International Journal of Technology and Design Education.

EnergyPlus. (2016). Weather Data: Amsterdam 062400 (IWEC). Retrieved from https://www.energyplus.net/weather-download/europe_wmo_region_6/NLD//NLD_ Amsterdam.062400_IWEC/

Foley, J.D., van Dam, A., Feiner, S.K. & Hughes J. (1995). Computer Graphics: Principles and Practice. The Systems Programming Series (2nd ed.). Addison-Wesley. ISBN 0-201-12110-7.

Foundation PS. (2001). About Python. Retrieved from: https://www.python.org/ about/

Gemeente Amsterdam. (2013). Strategienota: Het Marineterrein. Rijksvastgoed- en Ontwikkelingsbedrijf. Ministerie van Binnenlandse Zaken en Koninkrijkrelaties, Amsterdam.

Geoffrey, E. (2004). Global Business Information Technology: an integrated systems approach. Pearson Education.

Hulsman, B. (2009). Die Domme Steden. NRC. Retrieved from https://www.nrc.nl/nieuws /2009/02/28/die-domme-steden-11690654-a96 1610

Jabi, W. (2013) Parametric Design for Architecture. London: Laurence King.

Janssen, P. (2006). The role of preconceptions in design. Design Computing and Cognition, pg. 365-383.

Kanteldenker. (2015). Van monogame stad naar polyamoreuze stad. Retrieved from https://kanteldenker.wordpress.com/2015/11/16/van-monogame-stad-naar-polyamor e uze - stad/

Kilian A. (2006). Design exploration through bidirectional modeling of constraints. Department of Architecture, MIT, Massachusetts.

Krish S. (2010). A practical generative design method. Computer-Aided Design, Volume 43, Issue 1, January 2011, pg. 88-100.

Lawson, B.(1980). How Designers Think - The Design Process Demystified. Problems and Solutions 7, pg. 82-93.

Lawson, B. (2010). The Social and Psychological Issues of High Density. In: Ng E, editor. Designing High-Density Cities for Social and Environmental Sustainability. London: Earthscan.

Lee G., Sacks R., Eastman C. (2006). Specifying parametric building object behaviour (BOB) for a building information modeling system. Automation in Construction 15(6), pg. 758-776.

Lévénez, E. (2011). Computer Languages History. Retrieved from: http://www. levenez .com/ lang/

Littlefield, D. (2008). Space craft: Developments in architectural computing. RIBA Publishing, London.

Nall, R. (2015, November). What are the benefits of sunlight: Sunlight and Serotonin. Healthline. Retrieved From http://www. healthline.com/health/depression/benefits-sunlight/

McNeel. (2014). Why Grasshopper?. Mcneel. Retrieved from: http://wiki.mcneel .com/labs/explicithistory/home.

Newell, A., Simon, H. (1976). Computer Science as Empirical Inquiry: Symbols and Search. Comm. of the ACM, 19, pg. 100-128.

Oosterhuis, K. (2014). Next Generation Building. Delft: TU Delft Faculty of Architecture.

Pauwels P., Jonckheere T., De Meyer R., Van Campenhout J. (2011). Increasing information feed in the process of structural steel design. Sustainable Construction and Design 2011, pg. 180-189.

Pierce, B. (2002). What is Object-Oriented Programming? Types and Programming Languages. Cambridge, Massachusetts: MIT Press.

Rijksoverheid. (2015). Bevolkingsgroei, 2010-2015. Retrieved from http://www.clo .nl/indicatoren/nl210205-bevolkingsgroei-nederland

Roudsari, M. (2016). Ladybug Analysis Tools. Retrieved from http://www. grasshopper3d.com/group/ladybug

Rowe P. (1982). A Priori Knowledge and Heuristic Reasoning in Architectural Design. Journal of Architectural Education 36(1), pg. 15-25.

Rutten, D. (2017). Galapagos. Retrieved from ttp://www.grasshopper3d .com/group/ galapagos

Schumacher, P. (2008) Parametricism - A New Global Style for Architecture and Urban Design. AD Architectural Design - Digital Cities.

Schumacher, P. (2011) Parametricism And the Autopoiesis Of Architecture. New York: Anyone Corporation.

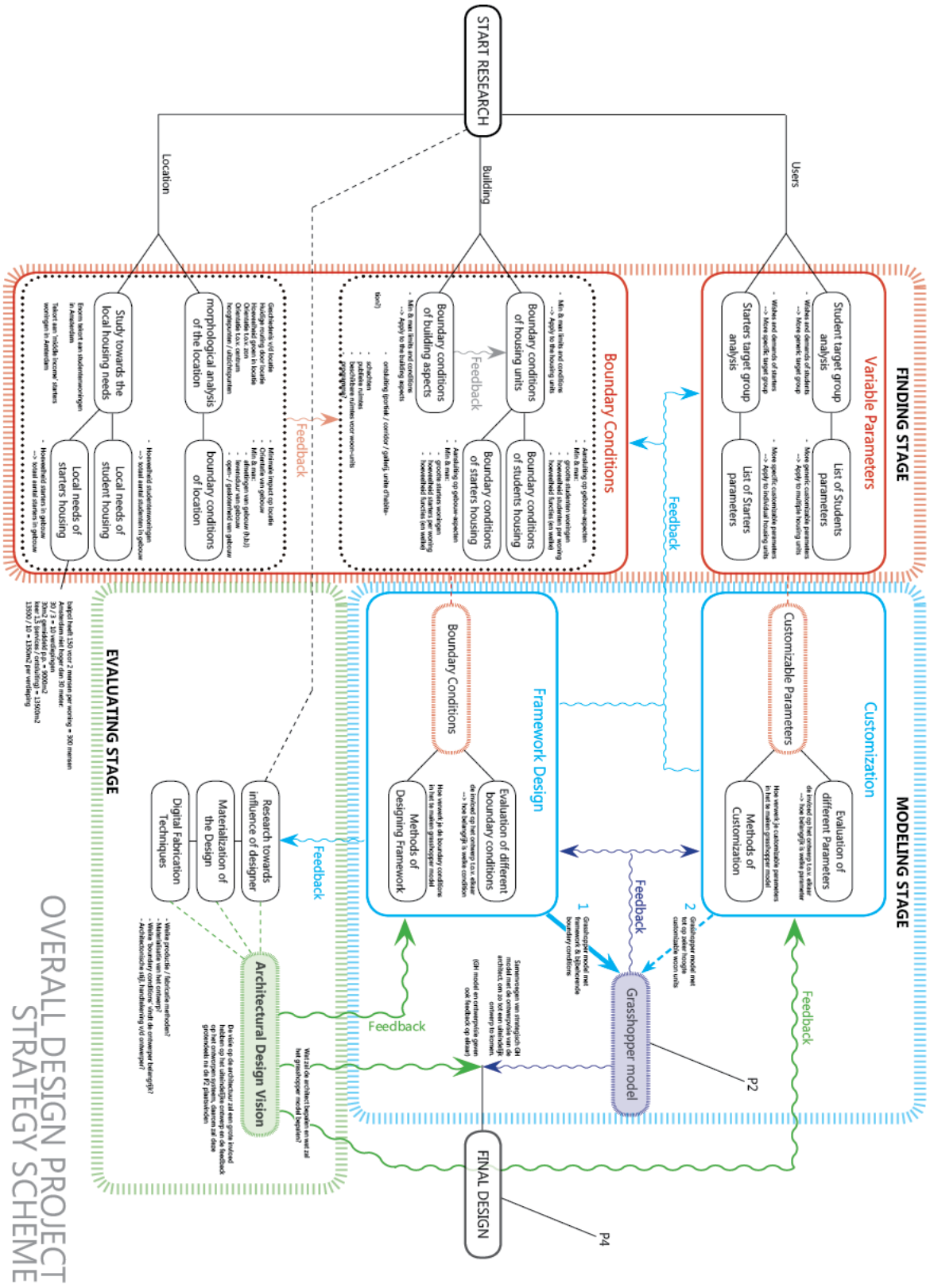
Shea K., Aish R., Gourtovaia M. (2005). Towards integrated performance-driven generative design tools. Automation in Construction, Volume 14 (2) Digital Design, pg. 253-264.

Simon H. (1973). The Structure of Ill-Structured Problems. Artificial Intelligence 4, pg. 170-200.

Uytenhaak, R. (2008). Steden vol ruimte. Kwaliteiten van dichtheid. Rotterdam: Uitgeverij 010. ISBN 978-90-6450-669-7.

6. Appendix

6.1 Strategy Scheme



6.2 Context

In Amsterdam there is an existing problem dealing with the shortage of affordable housing in the centrum of Amsterdam. There is not only a shortage in student housing, but also other low to middle income people have a hard time finding housing in the centrum of Amsterdam. Also, like in most cities, there is a shortage of space to expand in the inner city, therefore no solution can be found for this shortage of affordable housing.

Illustration 18 – Location of Marine Terrain

Recently, an area in Amsterdam, previously occupied by the Dutch Royal Marines, called Marine Terrain, became vacant. The marines are leaving this area, leaving a lot of unoccupied space within the inner city of Amsterdam. The municipality of Amsterdam already started to create a vision on how to spruce up the area in order to make it more attractive for the city. This vision is written down in the *strategienota ~ Marine Terrein (*Gemeente Amsterdam, 2013*).* This area could also be used to solve the housing shortage, however, without negatively effecting the vision and current structure of the area.

Attraction & Connection

In the current vision for the Marine Terrain (strategienota – Marine Terrein), we can see that there are multiple strategies placed in clusters within Marine Terrain. The chosen design location is within the horecapark cluster. The function of this cluster could be helping with creating a certain attraction for the design location.

Besides the new bridge, north of Marine Terrain, there is also another bridge option within the current strategy for Marine Terrain, which will connect Marine Terrain to Nemo, and therefore this will create a direct route from the Central Station of Amsterdam to Marine Terrain.

.

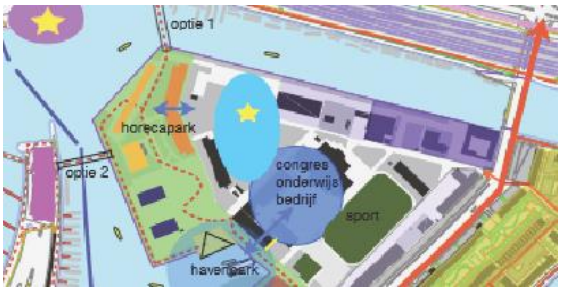


Illustration 19 – Clusters in Marine Terrain, Strategie nota, gemeente Amsterdam (2013)

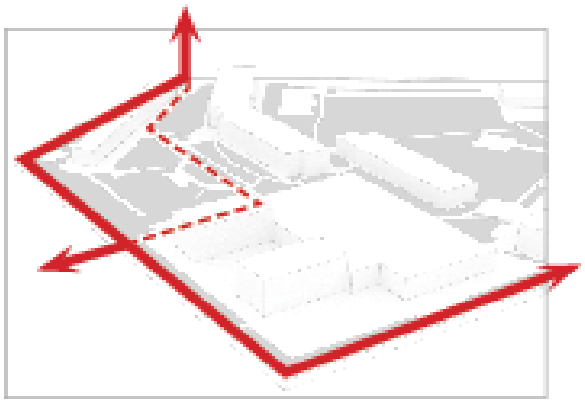


Illustration 20 – Primary routing through Marine Terrain

6.2.1 The Existing Stock

For the project location, I have chosen an area within the horecacluster and along the main routing through the area. This way the design will be better connected to centrum of Amsterdam and the central station. Below the already existing buildings within the horecacluster will be discussed.

Monumental building

The current strategy for this building by the 'Strategienota' of Amsterdam, states that this monumental building will be used for small innovative horeca businesses. The strategy for this building could help to create attraction towards marine terrain and the design location. Therefore, this building will be important for the location integration and interior / exterior spatial planning of the new design.

Student housing / Hostel

In the current strategy for Marine Terrain, this building will be transformed. Therefore, in the future this building will accommodate a combination of student housing and a hostel for tourists.

Makerversity Amsterdam

Currently already active on the location is the Makerversity of Amsterdam. A school / small business creator that focusses on making things using new techniques, like laser cutting, milling and 3d printing. Since this project is about creating affordable architecture with these new techniques, the makerversity can offer great value to the location.

Because of the diversity of these buildings, I have chosen to create this project in between these buildings

1. Monumental Building

2. Student Housing / Hostel

3. Makerversity Amsterdam

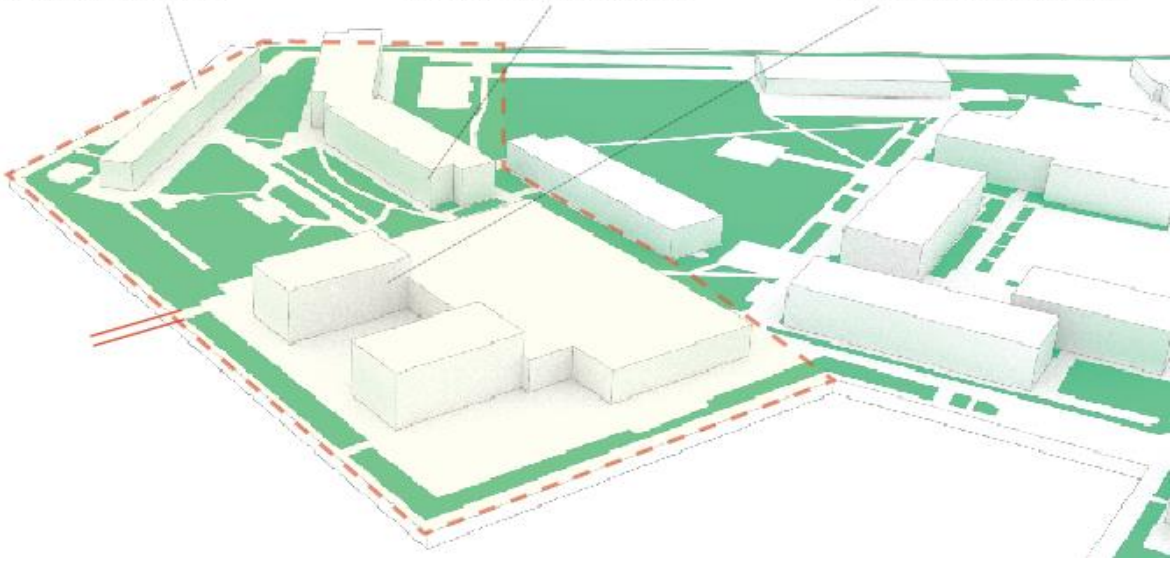


Illustration 21 – The existing stock on the chosen design location

6.2.2 Four Different Areas

Waterfront Area

This area has an aesthetic view over the waterfront and towards Nemo. Because of this aesthetic view, the area would be for starters housing. Also, this area will be connected to the central station of Amsterdam and its inner city. Besides, this the area will also function as an important connection route between the Piet Heinkade and the Prins Hendrikkade. Most people traveling through the Marine Terrain will follow this Waterfront route.

Student Area

Because of the transformation to student housing within the opposite building. This area within the project location would be suitable for student housing.

Business area

The Business Area is located in-between the Makerversity building and the project site. Therefore this area has a good connection to the Makerversity and its workplaces, meaning that a lot of start-up businesses (possibly funded by the Makerversity) could be located within this Area.

Horeca Area

This Area is next to the monumental building will, according to the current vision of Amsterdam, be transformed to accommodate small horeca businesses. This area therefore could create a lot of attraction for this area. The new building should take this into account in order to stimulate the area.

This diversity in functions would allow for a divers and interesting high density design.

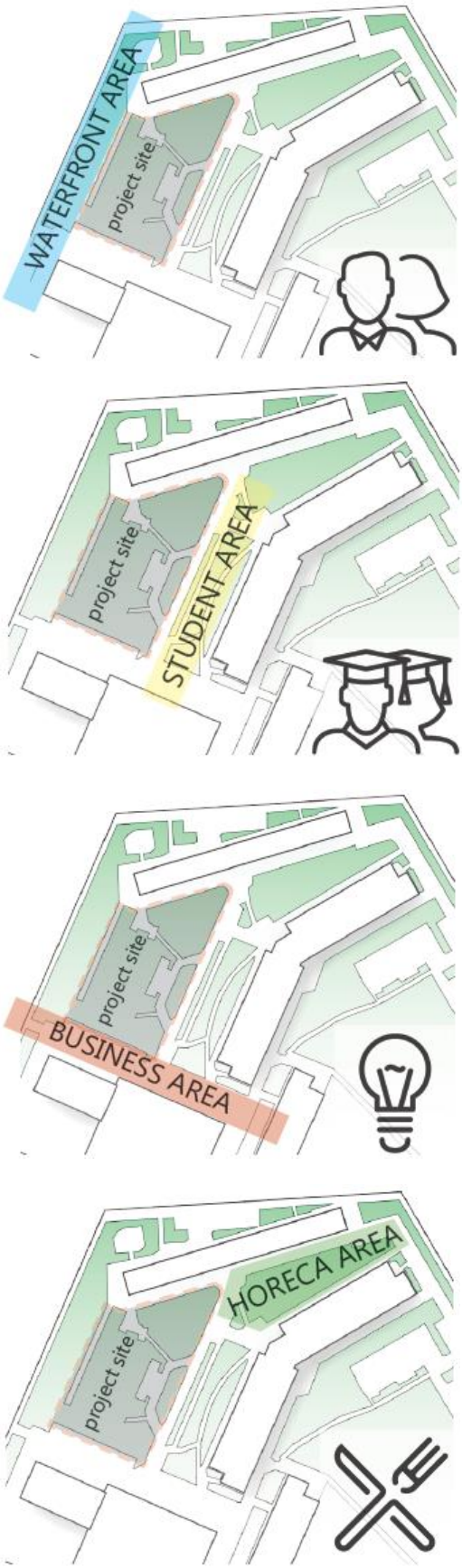
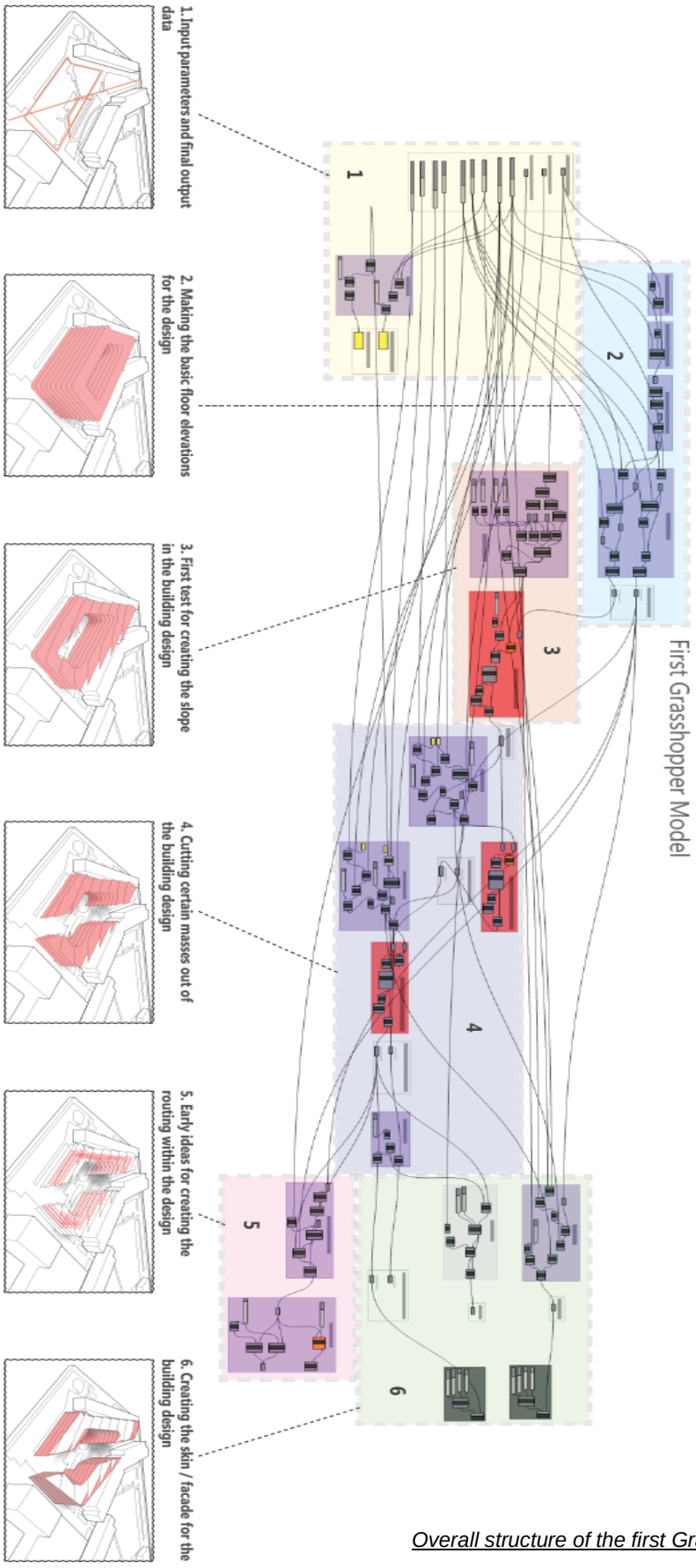


Illustration 22 – Different areas on the chosen design location

6.3 Grasshopper Models



Overall structure of the first Grasshopper model

Overall structure of the second 2nd model