Parametric Design – a Paradigm Shift?

by

Victor Gane

Bachelor of Arts, UCF, 1997
Diploma in Architecture, TUM, 1995

Submitted to the Department of Architecture
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE IN ARCHITECTURE STUDIES
AT THE MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2004

© 2004 Victor Gane. All rights reserved.

The author hereby grants to MIT permission to reproduce
and to distribute publicly paper and electronic copies
of the thesis document in whole or in part.

Signature of Author: __________________________________________

Department of Architecture
May 19, 2004

Certified by: __________________________________________

William Lyman Porter
Norman B. and Muriel Leventhal
Professor of Architecture and Planning
Thesis Advisor

Accepted by: __________________________________________

Julian Beinart
Professor of Architecture
Chair, Committee on Graduate Students
DISCLAIMER NOTICE

The accompanying media item for this thesis is available in the MIT Libraries or Institute Archives.

Thank you.
Readers

William J. Mitchell
Professor of Architecture and Media Arts and Sciences, MIT
Head, Media Arts and Sciences

Axel Kilian
PhD Candidate, Design and Computation
Department of Architecture, MIT
Parametric Design – a Paradigm Shift?

by

Victor Gane

Submitted to the Department of Architecture
On May 19, 2004 in partial fulfillment
of the requirements for the degree of
Master of Science in Architecture Studies

Abstract

The variety reflected in constant change became an imperative in the development of the modern world. The society is more insistent in seeking the implementation of quality and customization in most of humans’ activities. Such notions as satisfaction or contentment are achieved through having choices at one’s disposal.

While other creative disciplines are rapidly adapting to depict this evolving reality, architects continued to rely on traditional design methods, which in most cases is synonymous with a process resulting in few slowly developed choices. More recently, the adoption of computational aids did not have a significant impact as the latter are primarily used as tools to facilitate representations of designs. To keep up with the society’s dynamism the architects must adopt a new approach to design, one which will facilitate the exploration of rational variety, allow them to programmatically search the solution space and develop systems or tools used in conceiving multiple designs. This thesis investigates parametric design as a possible remedy.

Thesis Supervisor: William Lyman Porter
Title: Norman B. and Muriel Leventhal Professor of Architecture and Planning
Acknowledgements

My appreciation for making the experience at MIT significant is directed but not limited to the following people:

Bill Porter – above all for his universal wisdom and guidance. The Design Inquiry sessions will be long remembered for helping me think about and understand from a different perspective the design process. I was truly lucky to catch Bill before his retirement and to have had the honor of being advised by him.

Axel Kilian – for his inexhaustible energy and urge to reason things, which were an important source of inspiration. I thank him for introducing me to the very topic of this thesis and being my reader, as well as for his constant availability to discuss and question things.

Dennis Shelden – for finding time in his busy schedule to talk about the meaning of parametric design.

Takehiko Nagakura – for the fruitful collaboration on several design competitions and for my participation in his Unbuilt Projects series (Le Corbusier’s Governor’s Palace in Chandigarh, India). I appreciated Takehiko’s drive to perfection in visualization issues.

and my colleagues from whom I have learned a lot.
TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Worlds and objects .................................................................................................................. 9
1.2 Complexity in design – concept of rules, variations and ambiguity ..................................... 13
1.3 Meaning of parameters in design .......................................................................................... 18
  1.3.1 Abstract / implicit parameters .................................................................................... 18
  1.3.2 Explicit parameters .......................................................................................................... 23
  1.3.3 Variations and constraints ........................................................................................... 28
1.4 Connecting to architectural design realm ................................................................. 34

CHAPTER 2: PARAMETRIC THINKING

2.1 Nature of tools ......................................................................................................................... 37
  2.1.1 Physical environment .................................................................................................... 37
  2.1.2 Digital environment .................................................................................................... 41
2.2 Building a parametric system ................................................................................................. 46
  2.2.1 Getting started ........................................................................................................... 46
  2.2.2 Expectations ............................................................................................................... 52

CHAPTER 3: EXPLORATION

3.1 Early explorations .................................................................................................................. 54
  3.1.1 Examples ..................................................................................................................... 54
  3.1.2 Conclusions and issues ............................................................................................... 60
3.2 Experiments ......................................................................................................................... 62
  3.2.1 “Reasoned” ambiguity .............................................................................................. 63
  3.2.2 Parametric “freedom” ............................................................................................... 69
  3.2.3 Rules and emergence ................................................................................................. 73
3.3 Alternative ways ................................................................................................................... 78
CHAPTER 4: REFLECTIONS AND CONCLUSION

4.1 Reflections on the exploration results ................................................................. 81
4.2 Conclusion and recommendations ................................................................. 85

APPENDICES:

Appendix A: Rules and emergence: VB script .............................................. 88
Appendix B: Constraint types ................................................................. 94

BIBLIOGRAPHY
CHAPTER 1: INTRODUCTION

1.1 Worlds and objects

This work is an attempt to address the greater issues of complexity and variations in design. The contemporary society evolved to a point when it demands mass-customization, quality and efficiency from most of the products it consumes. People are the end users of designs we create, whether it is a work of art such as sculpture, painting or music, a utilitarian item like a cell phone or a meaningful space to inhabit like architecture. All the same, designs can be seen in things that humans did not originate. Obviously, nature is the ultimate designer that we are striving to understand through rationalization and, why not, even biomimicking. And what is not understandable, we still tend to imbed meaning into it. In the complexity of nature, despite the randomness and chaos that may at first characterize it, we see patterns and rules that govern this “system”. This is why it is imperative to first understand the notion of the world(s) in which all these creations or designs are born or already exist. What does comprise a world? Do we live in one or in countless worlds? Do we create new worlds or interpret existing ones?

Nelson Goodman approached this issue from an “irrealist” position, or in other words neither a realist nor an anti-realist. “If there is but one world, it embraces a multiplicity of contrasting aspects; if there are many worlds, the collection of them is all one. The one world may be taken as many, or the many worlds taken as one; whether one or many depends on the way of taking.”

one single world is absolute. But this position is not one of pure indeterminacy or subjectivism. Within Goodman’s framework it is possible to examine a painting, drawing or photograph and develop a precise description of what distinguishes these forms. Distinctions are crucial for any world to be valid because they are made only within a frame of reference. Thus, worlds act as references for the objects and ideas they encompass. In the case of a building such as Ando’s Church on the Water we experience colliding worlds as the one formed by the building is unconceivable without the surrounding landscape. One world acts as a reference for the other. “Worldmaking as we know it always starts from worlds already on hand; the making is a remaking”2 Goodman continues. Shall this building be placed in the midst of a dense city neighborhood, the entire frame of reference changes along with the perception of the inner world formed by the church’s interior. Or is the world of a different scale conceived by Le Corbusier in Chandigarh, India complete without the unbuilt Governor’s Palace? Further distinctions can be made that would point to the tangible or intangible, physical or nonrepresentational character of the worlds that we accept or create. For instance, the only way to comprehend Plato’s Theory of Ideas or more particularly his Idea of Good is to reference the latter to the deeds that connect it to the tangible world. But there is no guarantee for this connection to occur as it is one’s choice to accept or ignore that idea or altogether its implementation.

Despite the wealth of nonrepresentational worlds most of us prefer a tangible frame of reference when creating or experiencing the existing. Even abstract minds like Kandinsky characterizing art as an “adaptation of form to its inner meaning” was employing in his compositional work primary

---

forms representational of the surrounding physical world. Generally, ideas and designs are epitomized through objects which inhabit the world(s). “Our perceptual world is a world of objects, whole figures, and relations, not a simple integration of dark and light patches on the retina, or of sensations.” So far I have determined that the world acts as a reference and objects use these to establish their relation in the world(s). But let’s zoom in and inquire on how do the objects connect to the tangible world and interact among themselves. By interaction I refer to Newton’s Third Law – when A pushes or pulls B, then B pushes or pulls A. Or in other words, the action applied to an object by another results in a reaction force that affects both objects. The resulting change in the state of the objects depends on the types of objects involved and the way these are constrained among themselves or to the world they inhabit. The Oxford English Dictionary defines a **constraint** as the *exercise of force to determine or confine action*. The physical world that we know around us is made out of objects that in most cases use some sort of constraints. When considering any object, the way these are constrained will define their condition in reference to the world. These can be summarized into three main categories: fully constrained, partially constrained or non-constrained excluding gravitational force. A fully constrained or fixed object will have the specified linear and angular velocity obstructed (not considering the case of the force being greater than the constraint can resist before it fails). Hence, the displacement of a fully constrained artifact such as a high-rise will be none when the wind force acts upon it. I will leave out the deflection in this case for the clarity of the example. The structural pillars along with the foundation are confining the movement or rotation of the building in any direction.

---

When considering partially constrained objects the rotational or translational motion is possible only to a degree. The Newton’s Cradle experiment demonstrates the principle that action and reaction are equal and opposite. If one of the steel balls is lifted and allowed to fall back at one end, then one will swing out the same distance at the other end. This is possible only because all the wires holding the balls use constraints that allow axial rotation. Metaphorical similarities can be found in nature as well. A flower that opens and closes its petals at the dawn or dusk has these constrained to the receptacle, which in its turn is fully constrained to the peduncle.

The non-constrained objects experience “full freedom” when acted upon by a force. A ball will freely roll in the direction the force was applied until it stops because of the friction. Although, its path will also be determined by the landscape. Hence, a non-constrained object is the most “vulnerable” in terms of its constancy in the world(s). Nature again serves well in pointing out such examples. The snowflakes fall freely while being affected by the wind. Or, the raindrops after falling continue their journey depending on the configuration of the terrain they encounter.

In summary this chapter pointed to two major notions – there is a close interrelationship between objects and worlds because both act as references for each other. And, objects cannot exist outside of the idea of constraints, which define their position and interaction with the world(s).
1.2 Complexity in design – concept of rules, variations and ambiguity

"The movement from a view of life as essentially simple and orderly to a view of life as complex and ironic is what every individual passes through in becoming mature." ⁴

From this August Heckscher passage, one can point to the metaphor that from an initial perception of the world as a complex, chaotic system of interwoven designs we gradually start imbedding meaning into what at first seems to be incomprehensible complexity. Such complexity can be characterized as an assemblage of human-generated designs, integrated into the natural environment that serves as one and probably the most tangible of the worlds discussed in the previous chapter. The meaning grows out of an enduring rationalization of the world that starts with a mere simplification of its components. Such a process helps shape the facts used in understanding the world. A simple psychology test proves the inability of most humans to cope with remembering several tasks presented linearly. Without an initial simplification into facts the human mind would simply be overwhelmed when attempting to reason the meaning behind this complexity. So, what generates the complexity is the connection of our designs to the world.

Rules are probably the best illustration of this connection. "There is no work of art without a system", Le Corbusier once said. Any design process relies on multiple sets of rules but generally no single set is in a governing position until the designer decides which one is dominant. Hence, before proceeding to designing a building, an architect is confronted with a program from which he develops sets of explicit rules.

These can either operate independently or semi-independently, and take form of certain spatial configurations or in other words be driven by the aesthetic urge, follow some performance criteria, or in case of a poor rule-maker ignore all together constraints that would refine the design results. The implementation of the rules will have a direct impact on the immediate world around the developed artifact – the site and surrounding environment, or at a greater scale when considering a high-rise, the city as a whole will be affected. Consequently, good rule-making also means the ability to reveal and incorporate satisfactory conditions into the sets of rules that are to govern the design and the world they affect. But what serves as the selection criteria when choosing the rules is a little fuzzier. In the case of an architectural design program the probability for two designers to detect and implement identical sets of rules is virtually nil. This clearly points to the fact that a designer's selection of rules is quite arbitrary.

The use of rules and the way these are established along with the implementation of the chosen constraints defines the range of variations in any design. Variations play an important role in imbedding complexity into a design practice. “Facts are small theories, and true theories are big facts. This does not mean that right versions can be arrived at casually, or that worlds are built from scratch. We start, on any occasion, with some old version or world that we have on hand and that we are stuck with until we have the determination and skill to remake it into a new one. Worldmaking begins with one version and ends with another.”5 Expanding on Goodman’s idea I can imply that without the ability to introduce variations while creating, a designer may quickly run into a deadlock. The creative

process normally undergoes numerous revisions while gradually reasoning the world intended to be inhabited by the artifact.

It is important now to differentiate between the types of variations, which impact the end results and timing needed to explore the range of solution content. Charles Rusch identifies four strategies when addressing the concept of variations. He begins with the case of Blind Variations in which these are made independent of one another following tests (see below the context) that may or may not contain errors from a designer’s perspective. The incompatibility of this strategy is apparent in an architectural design realm simply because it lacks coherence of reason. The resulting designs vary but do not respond to the need of correcting errors. Thus, there is no reduction in the solution space after tests with errors are encountered because there is no change in the selection criteria. In the Trial and Error strategy the variations are responsive to the errors in previous iterations but these are not necessarily consciously made. This would yield adjustable results in a non-systematic fashion or in other words there will be a narrowing in the solution space following the change in the criteria. The Insight strategy is close to trial and error but in this case the designer is more conscious of the pursued goal and seeks to understand the depth of the problem. By exploring variations more systematically he is expecting a revelation to occur in terms of understanding. The last and most efficient strategy is the Gradual Analysis. Now the designer from the very beginning has a clear understanding of how to adjust the variations to achieve a goal and be consistent in his exploration.

---

4 Rusch, Charles W. Graduate Student, UC Berkley, graduated in 1966. His thesis used in current analysis of types of variations.
Rusch’s analysis was done in context of a series of eleven lithographic bulls by Pablo Picasso from his 1945-46 period. The lithographs went through a sequence of transformations from realistic representations of a bull at the beginning to very abstract renditions in the end. The author draws a parallel between adjustments to the lithographic stone (used in developing the entire series) and the application of tracing paper by architects when these are revising early ideas. He juxtaposes conceptual architectural design as “a rather lengthy series of overlays built before a satisfactory form is reached” with Picasso’s lithographs, “much more controlled than the average architectural series.” Here one can argue that variations in early stages of an architectural design can be attributed to the insight strategy simply because architects normally do not know exactly what the end result will be, although the search is done systematically by addressing simultaneously different subproblems that give direction to the overall solution. Similarly, Picasso’s method fluctuates somewhere in between insight and gradual analysis, although closer to the latter. From the beginning his scope was clearer than an architect’s, since he intended to simplify the bull into an abstract representation. Furthermore, his process indicates an apparent understanding and consistency of the needed modifications to achieve his goal.

So, variations are important in achieving complexity. But the idea of design is larger than merely the use of rules and the consequent variations that emerge out of these. Besides, variations alone are not enough to support complexity. The use of rules generally also eliminates ambiguity from the design process unless these are nondeterministic. Without an element of surprise, discovery or in other words controlled ambiguity the designer will be in a position to predict the outcome – something that defies one of the essential notions of design, namely to invent. By controlled ambiguity I
suggest the ability to work towards a goal through means that are not entirely in the designer’s control – a process that can possibly generate irregularity. These means can be in form of tools that one uses to complement the creative process. In our classic understanding of the use of tools a designer would have full control over these. But more recent developments point towards a new philosophy, one in which the designer develops his own tools, normally digital, that act as co-designers, assisting in creating the content of the solution space. In such “collaboration” the human designer would simply synthesize the outcomes generated by the digital system with his own thought process that gave the initial direction to this system’s mechanism of generating design variations. Obviously, an important question must be answered in this context, namely what are the dimensions of employing ambiguity before the resulting designs obstruct a valid solution range? This is an issue that will be amply addressed in Chapter 3.2.1 of this thesis.

Obviously the meaning of the term ambiguity is itself quite ambiguous. Among its definitions given by Oxford English Dictionary one reads *uncleanness by virtue of having more than one meaning*. Jastrow’s classic example of the Duck-Rabbit7 duality of perception proves that the human mind has a tendency to pursue new understandings after becoming familiar with an initial one. In the same way, the ambiguity in design supports complexity through allowing the reading of multiple meanings. Robert Venturi identified numerous such examples in his *Complexity and Contradiction in Architecture* book. From Luigi Moretti’s Casa il Girasole in Rome, in which he questions whether this is one building split in two or two joined together to Le Corbusier’s Villa Savoye, where the simplicity of the exterior juxtaposes the

---

7 Duck-Rabbit image by Jastrow, published in Fact and Fable in Psychology, 1900.
complexity of the interior, all point to “richness of meaning over clarity of meaning”\(^8\).

In summary, it has been established that complexity in design is generated by employing different rule sets that produce variations. The selection of rules by the designer is arbitrary. And the concept of complexity gains from the introduction of “reasoned” ambiguity both in the meaning of the resulting artifacts and the method used in developing the latter.

1.3 Meaning of parameters in design

Now that a framework for understanding designs within a world has been developed, I am ready to move on to inquire into the meaning of parameters. Parameters are critical for the rules to operate and accordingly for the variations to be possible. They are the main building blocks of any design, be it physical or virtual. So, a parameter can be formulated as any factor that defines a system and determines or limits its performance. These can vary from a set of measurable factors, such as temperature, pressure, distance, etc. to a set of nonfigurative measures like an individual’s state of emotion (i.e. happiness and sadness) or the aesthetics of an artifact. But my investigation of parameters will be limited to the context of design because of the vastness of the topic. Next, I will introduce a high-level delineation between the types of parameters.

1.3.1 Abstract / implicit parameters

An interpretation of abstract or implicit parameters can be in their role of defining rules which result in ambiguity of meaning in terms of being opened to interpretations. It is

harder to identify and establish such parameters when keeping in mind their meaningful use but they are easier to implement. When designing with these parameters one can obtain most freedom simply because the design process is less constrained when using the same parameters to obtain various emerging results. "Mies, for instance, makes wonderful buildings only because he ignores many aspects of a building. If he solved more problems, his buildings would be far less potent." 9 Or, continuing building from the idea in this Paul Rudolph excerpt, if using parameters with explicit meaning (see following chapter) the resulting designs will be of a totally different nature because of the way parameters affect the implementation of rules.

The work of abstract artists is probably most representative in illustrating the concept of implicit parameters. Piet Mondrian’s De Stijl movement proposed a parameterization of the world through an abstract visual language, characterized as a "grammar of shape and colour". Mondrian’s paintings, driven by a set of rules, embody a cluster of meanings. This concept was applied to various forms of art like painting, sculpture, architecture, furniture and interiors. The designer was the one to set the rules and regulations for parameters employed to portray an inner working of color, form and meaning.

Both Paul Klee and Wassily Kandinsky formulated laws of art as simple rules. Such notions as natural and artificial measurements for instance, emerge in Klee’s writings to define design parameters. These are his rationalizations of movement and countermovement or rise and fall – ideas that imply an extensive range of design connotations. In Klee’s works employing such parameters can either signify

a "crescendo and diminuendo between the poles of white and black"\textsuperscript{10} or in other words point to how manipulation of colors can convey movement, or suggest a depiction of literal movement in a composition through the means of graphic representation like points, lines or shapes emerging out of these. Hence, in figures 7–8 natural movement is represented in a dual fashion – literally by suggesting a direction and in terms of continuity of the shape representing this direction. On the other hand, figures 9–11 express the artificial movement, again suggesting a directional application of this abstract parameter but this time as an orderly, stepped, human-reasoned continuity.

So, the abstract parameters that act as constraints in this example are also rules defining the direction and type of movement. Now let’s look at Klee’s application of these. The pen-and-ink drawings in figures 12–14 are examples of the artificial measurement depicting increase or decrease (Fig. 9) and simultaneous increase or decrease (Fig. 10). In figure 12 the painter is representing the theme of rain through highly stylized illustration of raindrops. The base on top suggests a plane of reference that can be viewed as the sky from where the rain originates or possibly a rooftop from which the drops undergo a new round of formation to then continue the fall. The multiple meanings of both the scale and shape of the drops are due to the abstract nature of the motion parameter, which leaves room for speculative perception. This is somewhat similar to Jastrow’s Duck-Rabbit image in the sense that the raindrops can be seen in each individual large element of the composition and simultaneously in the multitude of lines forming the larger entities.

The second drawing (Fig. 13) is also an example of using the increase or decrease parameter but with the base or plane of reference in the middle. The reason behind the latter is the painter’s intent to illustrate a reflection scene, in which the application of decrease parameter was employed to clearly represent the pagodas’ general characteristics – a gradual, stepped reduction in each floor’s dimensions. The increase occurs to the opposite of the water line – in the reflections, the true representation of which has been abstracted through horizontal displacement. Multiple meanings can again be spotted. For instance, at the detail level the variation in the line types delineate the individual levels in a pagoda. Similarly, in the drawing shown in Fig. 8, Klee used only the decrease parameter to portray a city skyline. Note that the base this time is at the bottom to form a reference of a ground plane. Types of lines again act as boundaries between buildings, their levels and depth of space. The element on far right, for instance, is partially obstructed by its bigger neighbor. As a result one could perceive it as being farther.

Let’s examine another set of Klee’s abstract parameters, this time addressing “dynamic density”. By dynamic density the painter implies “discharge of tension from within” – a concept illustrated in Fig. 15, where tension takes form of a dimensioned motion, a progressive decline in density. The rules in Fig. 16 establish the directions of tension distribution and in Fig. 17 – a differentiation between regular and irregular motion. The result of combining parameters is the diagram shown in Fig. 18 – dynamic density in two directions. It is apparent that all these rules just set the stage for a general theme such as dynamic density in this case, leaving the solution space undefined in terms of concrete implementation of rules, which are ambiguous. This narrow set of parameters just points to a possible direction, leaving
at the artist's discretion the very representation of the design. The dynamic density idea of Fig. 15 can either be a literal execution of the rule as a set of lines (i.e. Klee's *Variations*, Fig. 5) or rather serve as a conceptual guide, open to multiple interpretations and representation techniques like Andy Warhol's *Atomic Bomb* (Fig. 20). In the latter, the dynamic density is conveyed through a series of cells gradually decreasing in size but increasing in color intensity, thus supporting one of Klee's parameters defining tension distribution.

In all these examples the application of parameters serves to define a "big" idea at a macro or conceptual scale. The implementation in terms of details has been left to the designer's discretion, making the interpretation of the abstract parameters ambiguous. Just consider the increase / decrease rule that implies a dual meaning. In the *Pagodas by the water* example some may see the decrease occurring to the reflections and not in the pagodas above the water line.

In this chapter I have established the notion of abstract parameters as key elements of rules resulting in ambiguous implementation procedures. Because of their implicit meaning these parameters are mainly preferred by artists (abstract painters, poets, etc.). They can achieve a simultaneous satisfaction of the urge for freedom in pursuing designs with complex connotations combined with a rationalization of the process of designing in terms of rules. In the end, the artist creates a rule set that acts as a design guide.
1.3.2 Explicit parameters

Explicit parameters are the ones that we normally use when designing. These are preferred because of their clarity and the predictability of the resulting variations in the developed artifact. “The artificial order is impoverished but clearer, more comprehensible.” Unless dynamical relationships are established among them, when employing explicit parameters the designer has full control over the evolution of a design conceived parametrically. With the introduction of dynamical relationships an additional layer of complexity is introduced and expressed in the typically inefficient human ability to perceive simultaneous responses in parameter changes.

The concept of parameters in design is quite old. In antiquity the idea of proportions and beauty was long challenging the human mind. Early on attempts to reason these were done in Egypt, where squared grids were introduced to assist the artisans in obtaining desired proportions of human figures and also to lay out the composition as a whole. The system made possible for multiple people to work simultaneously on individual sections of a fresco or relief. The grids were first drawn on the surface before the scene was sketched and the content of each section of the grid was transferred at the appropriate scale. This standardized grid made the human proportions remain virtually unchanged for several millennia. Hence, the parameterization of design in this case occurred through the use of the grid, which established proportional conventions and allowed dimensional scaling.

The Classical period knew a similar evolution. It introduced a rigorous use of orders and rules of scale and proportions,

---

which became the foundation of the Western architecture up to modern times. The well-known Doric, Ionic, and Corinthian orders emerged and each had a distinctive character defined by different proportions and decorative conventions. The rules were later modified by Romans who created the Tuscan and the Composite orders.

Artists in the Renaissance times were employing rules of proportions to define their compositions. In Leonardo’s view, a pyramid was an ideal form in the two dimensions of art and often the figures in his paintings were grouped in a pyramid composition. In studying the human form, artists of the 16th and 17th centuries were applying at an elementary level the principle of coordinates to the study of proportion. They were using methods that were classical in origin and amply described by Albert Durer in his “Treatise on Proportion”. Similar to Egyptian frescos, through a parametric grid Durer reasoned the human figure, facial expressions and features, which were transformed by slight variations in the relative magnitude of the parts.

All of the examples above point to dimensional-based parameters such as length, width, height, radius, etc. These are used in both analyzing and referencing to the world existing objects or designing most of the human-made world – be it an aircraft with its more than one million individually crafted parts or such a simple assembly as a pen. But no matter how simple a design is, it is successfully completed through a harmonization of parameters defining its parts. In case of a car collision we are able to replace the damaged part because of the parametric “identity” of the replacement, which will make it fit within the entire assembly. Hence, the manipulation of parameters whether for analysis or design must suggest a meaningful process of inquiry into the solution space. For Durer’s studies, for instance, one
can easily identify a range within which the variations would make sense. Therefore, if examining the 3rd illustration (Fig. 23) in which the height parameter of the second upper row has been altered to represent a tall forehead, should this be twice the value, then the iteration would lose its significance as it would be outside of the meaningful range. It is also important to identify this as an independent parameter because its manipulation does not affect the value of other parameters in the design.

Whether complex or simple, human-generated physical artifacts can be rationalized into a set of basic 3D entities that in their end can be represented by two-dimensional ones. A cube can be viewed as being made of six surfaces, each of which is defined by four lines. A line can mean either a trace left by a moving point or a length without thickness that connects two points in space. Therefore, in this context a parameter can be defined as an independent variable in terms of which each co-ordinate of a point is expressed, independently of the other co-ordinates.

But the concept of explicit parameters is present in cases outside of merely dimensional values. Music overall evolved into a highly parameterized form of art based on a series of rules that quite often define the outcomes. Starting with the Benedictine monk Guido d’Arezzo this field underwent historical changes in terms of systematization. With his invention dated around 1025 the composers were able to begin recording their work in format of a manuscript. D’Arezzo created a system of musical notation using a 4-line staff which has evolved into the contemporary standard notation system that uses a 5-line staff. Before the invention of musical notation, every singer had to memorize the entire repertoire and then teach it to the next generation. Consequently, over time errors attributed to memory or
differences of taste caused the music to change. The notation that d’Arezzo developed became the foundation for reasoning the structure of music by systematizing its parameters.

We know that the world is full of ambient sounds, the majority of which can be regarded as noise. The parameters in music reason this noise and through compound relationships contribute to conceiving works of high complexity. The first such meaningful parameter is the tone, which is distinguished from noise by a definite pitch. Tones are inconceivable without the notions of pitch, intensity, and quality. The pitch represents the frequency of vibration of the tone’s source. Intensity of the pitch is established by the amplitude and quality is determined by the overtones, the distinctive timbre of any instrument. Parameters are the components of the rules used in conceiving music, which makes most of the latter identifiable and measurable. These rules can be regarded as a complex design mechanism, which contains multiple levels of parametric interrelationships to shape the final assembly in form of a melody. One can instantaneously detect dependencies in the case of the pitch that cannot exist outside of such parameters as intensity and quality. Furthermore, another layer of pitch’s dependency can be attributed to the system of notation, which in its turn is tightly integrated with the five-line staff. This example establishes the meaning of a dependent parameter as one which cannot exist outside of another parameter.

An additional example of such interdependency is the higher level musical parameter denoted by the tonal orders. Differentiated into pentatonic, diatonic and chromatic, the orders operate as systems allowing certain variations of parametric schemes to evolve. This is due to the range of tones contained in each of these orders. With the most
number of tones – twelve, the chromatic order juxtaposes its opposite, the pentatonic with only five. We mainly tend to rely on the diatonic order as the reasonably complicated one. In a sense, the tonal orders act as a parametric skeleton that defines the way tones combine to form intervals.

An interval is yet another basic parameter in music. Being merely a two-note combination or the difference in pitch between two tones, intervals serve as the foundation for all tonal music. Multiple such combinations vary between half-step and whole-step intervals. The abundance of these is determined by the chosen tonal order. Obviously that in the chromatic order we’ll encounter most half-steps as opposed to pentatonic, which has none. Consequently, the music played with intervals from a pentatonic order will sound much simpler than the one in chromatic. In other words, the initially chosen skeleton will dictate the outcome because of the range of parameters that this skeleton can afford.

A more complex example of a parametric skeleton is embodied by scale patterns. These act as series of tones arranged in a step-by-step rising or falling order of pitch of which tonal music is built. There is a strong parametric relationship between the pitch and the scale, as the employed scale dictates the involved pitches. Each scale has a distinct character and the interval patterns define the melody and the harmony of music written in that scale.

Obviously that the wealth of music as a subject is enhanced through its multiple levels of parameters, culminating with more intricate ones like harmony, which can afford be both "cloudy" or clear and still be meaningful. The variation of interpretations in music can be attributed to the complexity and relationship of parameters used in conceiving music.
In summary, this chapter discussed the notion of explicit parameters and what differentiates them. Independent parameters, which are not affected by and do not affect other parameters in a design have been compared to their opposite – the dependent ones. The choice of parameters has to do with the identity of the design. I have determined that such complex design as music is conceived of explicit parameters that mostly have non-linear relationships. If a melody is regarded as an assembly of parameters, the manipulation of one will implicitly affect the others. An architectural design process is very much non-unidirectional and can be compared metaphorically to the process of composing music. The difference is in the less rigorous reasoning approach of parameters used in conceiving architecture. The ability to bring reasoned rigor in an architectural design process through the means of processes or methods of manipulating parameters harmoniously will significantly impact the contemporary architectural practice.

1.3.3 Variations and constraints

*What we find, or succeed in making, is heavily dependent on how and what we seek.*

Variations are at the very essence of this axiom – they are the driving mechanism of a search through the solution space. Designer establishes the skeleton or the rules that make a range of variations possible. “*All styles, the traditional ones as well, are based on repetition. What is style other than the self-control of the person who limits himself to carry out some choices within the range of his own taste?*”

---

original intent. In Chapter 1.2 I have established the notion and types of variations along with the importance that these play in making complexity. In this chapter I will look at several examples to illustrate how variations affect the design and what determines their range.

"Variation is that kind of repetition which changes some of the features of a unit, motif, phrase, segment, section, or a larger part, but preserves others. To change everything would prevent there being any repetition at all, and thus might cause incoherence."\(^{14}\) Variations emerge as a result of either direct manipulation of a parameter(s) or from implementation of rules that affect a design through its parameters. In both cases a designer will be faced with the necessity of carefully reasoning the changes he makes in order for these to have a meaningful impact on the overall design and thus avoid "incoherence". But how can one define meaningful variations in this context? It is apparent that in case of manipulating the values of an independent parameter\(^{15}\) the resulting variations are generally easily foreseeable. This suggests that the meaning in terms of the end result of a variation is entirely in the designer's grasp. In a Mondrian painting for example, if the artist were to modify the initially employed parametric rule, which established through vertical and horizontal lines the proportional convention of the color blocks, the result would be instantaneously tangible. The independent parameters in this case are the vertical and horizontal lines and their positioning. Should the position of the vertical line be modified in the example from Fig. 1, the horizontal lines would not be affected. Similarly, adjusting the height of the parameter defining the forehead in the earlier examined


\(^{15}\) See Chapter 1.3.2
Durer’s studies will not affect the immediate parameters describing the nose and tip of the head. The meaning is visually simple in these illustrations because of the linear character of the impact the parameter has on the design.

Dependent parameters on the other hand reveal a more complex structure of meaning. Since these exist and operate in the context of other parameters, the variations resulting from manipulating a dependent parameter will require more thoroughness of thought or in other words a more profound pre-rationalization of the impact these will have on the design. In this case the outcome is still tangible but not immediately obvious because of established dependencies between parameters and the non-linear character of their action. Consider for example a watch mechanism with its multiple ratchet wheels (Fig. 25). Each of these wheels is controlled by a counterpart, which in its turn drives another counterpart wheel to ultimately achieve a highly reasoned, meaningful process. The meaning in this assembly is possible because of the harmonious operation of its numerous dependent parameters. The range of variations is different for the three output parameters that indicate the hour, minutes and seconds.

Now let’s look back at some earlier examples of parameters in music and inquire on how these determine the variations. Patterns of whole and half step produce either major or minor scales. “Some scales use one or more intervals larger than the whole step. This variation in interval size gives each scale, and the resultant music in that scale, a particular color, quality, or ambiance. The unique interval patterns of a scale are transferred to the melody and the harmony of music written in that scale.” Both major and minor scales

16 See Chapter 1.3.2
consist of the same number of notes (7) with identical number of whole and half steps (5 and 2). Nevertheless, they produce an absolutely different series of variations because of the difference in the patterns of whole and half step intervals.

Continuing with the idea of intervals, rules are used in establishing their position on the staff. The interval recognition is done through incremental combinations of notes of a second, third, fourth, etc. In the case of notes of a second the distance between these is just one half of a step, in the notes of a third – a whole step and so on. Consider a simple example of variations in an interval of a third (Fig. 26). Played on a piano the la (F) and do (A) will sound entirely different from a la and do sharp or la flat and do sharp or la sharp and do despite all the intervals being thirds.

Thus, by manipulating a dependent parameter (a note in this case) the resulting designs will have distinctly different qualities. Furthermore, "the consonant-dissonant properties of intervals may be used to support or oppose, for various expressive purposes, other forces such as instrumental timbre, dynamics, and tempo. However, the same intervals assigned the timbre of muted strings create an entirely different effect" affirms Persichetti. This once again supports the potential and the multidirectional character of dependent parameters used in producing complex designs.

So, it is evident that in a design process meaning is quintessential for variations to be valuable. Optimization became a method for some disciplines to bring more

---

meaning to the idea of variations. Engineering is among the most active fields to implement this method. Today, for instance, the range of variations in structural design is often established through optimizing for minimum weight of the structure. Topology optimization is also notable. It helps search for a best arrangement of a minimum volume of structural material within an environment in order to achieve an optimal mechanical performance of a design. But optimization is not only a matter of material weight or structural topology. An architect for instance, can employ optimization to achieve an objective function describing data such as the total volume, area, the cost of a design, etc. Therefore, optimization can also be described in terms of constraints.

The meaning of variations is inconceivable outside of the idea of constraints. "Design is a process of expressing and exploring constraints and trying to achieve objectives." Constraints, which are none other than parameters, must be introduced to ground the numerous variables that a design process deals with. Without constraints a designer would be overwhelmed when considering the possible factors that may impact a design. In an architectural environment, by introducing constraints a designer gradually brings clarity and coherence into the design process. "Constraints provide a knowledge representation scheme that supports reasoning about designs and designing." Hence, an architect starts conceiving by first analyzing the program and the site, which represent the initial constraints. These are employed in developing a subset of additional constraints that ultimately refine the solution.

---

Constraints act as tools to systematize a design and are the raison d'être behind its development. The chosen constraints delineate the range of possible variations. But this range will be obstructed in the case of over-constraining a design. This will lead to the inability of performing any variations because of conflicts between parameters defining the design. Consider the simple example of a polygon shown in figures 27 & 28. In the first illustration the polygon is fully constrained, a state which allows adjustment of parameters and makes variations possible. Each of the sides except the diagonal has a dimensional constraint defining its size as well as geometric constraints that always keep the elements vertical and horizontal. There are also two more constraints establishing the position of the artifact in reference to the origin (the world). As soon as a dimensional constraint is introduced for the diagonal member, the polygon becomes over-constrained because of the conflict that the latter inflicts on the other parameters. Hence, when attempting to adjust the value of the diagonal, the connected vertical and horizontal members would need to respond by either increasing or decreasing their length value. Obviously, that this is not possible, since both members already have assigned dimensional parameters.

In this section I have shown that variations are central to the process of designing. For the variations to be meaningful there is a need to pre-rationalize the bearing they have on the design. Variations resulting from manipulation of independent parameters are easier to foresee because of the generally linear impact these have on the resulting artifacts. The more complex designs are created through the means of multiple dependent parameters with a non-linear process of generating variations. Finally, variations are impossible outside of the concept of constraints, which are the principal factor in establishing a range of possible variations.
1.4 Connecting to architectural design realm

So far I’ve determined the parametric character of virtually any design we encounter, be it abstract or physical. I looked at the components of a parametrically-driven process. Based on all the established notions, how can one define parametric design? It is a system that affords “inputs and outputs and that generates design spaces and mechanisms to arrive at a solution.” 21 Obviously that the inputs in this case are the parameters and the outputs are the variations resulting from their manipulation.

The product of an architect is very much a consequence of employing a parametric process of designing. Just consider his original tool – the drawing board in which two-dimensional manipulation of parameters is achieved through the versatile arm that allows angular and dimensional adjustments. But how effective and efficient is an architect in his pursuit of meaning in design? Meaning cannot be achieved through directly developing a single final solution, especially when considering all the variables that a designer has to address. This is why architectural design can be regarded as an iterative process of generating meaningful variations resulted from implementation of designer-chosen constraints. But the search through the range of design possibilities is currently regarded as a laborious, inefficient and expensive process that consumes a good deal of an architect’s resources. “Architecture still takes years – many years – to design and build.” 22 John Frazer denoted in his Spring 2003 lecture at MIT’s Department of Architecture the fact that most of their effort architects direct towards

producing illustrations of their ideas and only less than 20% of the time is dedicated to actual design. This developed into a tangible problem for me while being a designer in several architectural practices, when it became apparent how little creative work is involved beyond the initial conception stage of any project, when all the effort is directed towards representing a single solution. Also, a recent experience illustrated the impact that changing regulations can have on the design process. Following the tragic events of September 11, 2001 new stringent rules dealing with safety were established for airport designs. These were introduced in the midst of developing the extension to the Orlando International Airport, which resulted in a major overhaul of the project. With the now-conventional CAD tools implemented, the office still needed to undergo a lengthy and costly process of redesigning the scheme and its means of representation. This time the search through the solution space was even further limited. Combined with the time constraints, the additional design parameters that were introduced collapsed most of the logical hierarchy of functional and aesthetic considerations developed more or less in a manual process. It also became obvious that in their traditional sense the architectural design methods are unable to face the growing demand for efficient reaction to changing conventions. Furthermore, the currently employed computational tools have a narrow contribution to the realm of design. “No architect today can honestly say that he can achieve a totally satisfactory solution to a complex design problem. He simply does not have the tools to solve problems of the complexity we face today.” 23 These tools primarily support the post-design process and are merely used in design representation.

Parametric design has undergone an interesting evolution in the recent years. With the development of high performance computing platforms in the early 1990’s, this methodology started to be visualized in a different connotation than its traditional understanding, in which parameters are manually manipulated in a linear fashion. These new tools are opening possibilities for innovation in architectural design both in terms of productivity growth and increased complexity of the overall design process. In the following part of the thesis I am proposing to investigate parametric methodology as a partial remedy to some of the limitations that architectural design currently faces. Architectural solutions are normally achieved through solving problems incrementally. This is a slow, manually performed procedure that results in few designs. Furthermore, architectural design is normally greatly under-constrained, especially in the incipient stage. I will attempt to address this issue in the context of employing parametrically-enabled computational tools to allow for better understanding of the solution space. In other words, will a designer be able to explore more systematically a wider range of solutions for a given problem by constraining the prospective design through computational parametric frameworks that would act as mere skeletons? Will these frameworks allow the designer to respond to changes in his thought process at any stage of design development? Therefore, I am interested in establishing the potential of parametric methodology to act as a co-designer in terms of offering the architect choices of which he may not be aware of. Obviously that this exploration will lead to establishing the shortcomings and strengths of computationally-aided parametric design and suggest new avenues of inquiry.
CHAPTER 2: PARAMETRIC THINKING

2.1 Nature of tools

"A parametric representation of a design is one where selected values within the design model are variable, usually in terms of a dimensional variation. But any other attribute like color, scale, orientation could be varied parametrically, through a parameter. To design parametrically means to design a parametric system that sets up a design space which can be explored through the variations of the parameters." "

In other words, parametric design is a process of choosing appropriate parameters for a design problem and setting up the model definition that then can be used to explore the solution space. This model definition is constructed through employing tools that help designing in either physical or digital environments. Before proceeding to experimentally addressing the issues raised in the previous chapter I will first look into the nature of tools that make an architectural parametric design process viable.

2.1.1 Physical environment

In their conventional understanding the architectural solutions are created two-dimensionally on a drawing board or computer screen, which makes potential problems hard to visualize. Physical models still act as main complementary tools when designing in a physical environment. "Making by hand was the only way we had of fabricating artifacts for most of our history. It required a great expenditure of human energy." Indeed, for numerous architectural practices the iterative character of design was and continues

being done through the means of physical modeling. Normally this is a slow but sure instrument in the designer's hand, as it represents a scaled simulation of the reality. It helps visualize the problem in its entirety and hence test various parameters that make a solution successful through constantly refining its meaning. By learning of problematic parameters through constructing physical scaled representations of designs, an architect proceeds to elaborating the solution through multiple consequential variations. The parameters vary in their form and affect both positive elements and negative spaces within a design. Consider for example a wind tunnel test performed on a solution model of a building. The obtained parameters will indicate the architect how well the design performs in terms of suction forces and pressure and play a significant role in shaping the exterior envelope as in the case of Foster's Swiss Re tower in London.

Furthermore, larger scale models can serve to test details for structural performance, which will implicitly affect the interior spaces, thus putting at the designer disposal an unconventional set of parameters. In the case of Antonio Gaudi, who developed a process of designing with physical models, these new parameters were the weights in the hanging models. "Hanging models enable one to determine the optimal form of structures carrying loads purely in compression, particularly those consisting mainly of vaults." This was an innovative way of solving structural problems as the use of weights made possible to easily change the test conditions. In these models the suspension points were associated with the bases of the columns. The designs for pillars and vaults reflected the pattern of stresses, which were simulated by suspending weights on wires fixed to the

---


---

Figure 30: replica of Gaudi's model, exhibited in Casa Mila, Barcelona, Spain.
ceiling. The weights were equivalent to the estimated loads, making this a meaningful testing technique continued by disciples like Frei Otto.

In contemporary terms Frank Gehry’s practice is another good example of designing parametrically by testing solutions exclusively with physical models. "Physical models are the primary elements of the process where the project design is developed. These physical objects define and embody the formal design intent as it is developed over the course of the project." In the case of Disney Concert Hall, early on in late 80’s before the adoption of computational tools, multiple iterations of physical models were used in testing different parameters. Starting with acoustical performance of the hall, the physical models served in tests to determine for instance the optimal position of reflective acoustical panels. "The idea was to develop an ideal acoustical shape as the form generator for the building." This was the case of the exterior envelope studies that reflected the acoustical test results in the interior. The latter tests also helped in defining the seating scheme. So, in Gehry’s example the design process starts with sketch models that are quickly generated out of easily obtainable materials like paper or plastic cups. This proves to be a powerful method of developing schematic solutions in a physical environment. The efficiency of physical modeling in achieving relative freedom of parameter manipulation makes the paradigm valuable. Gehry’s modeling technique can also be compared to sculpturing, in which the artist adjusts the clay to imbed meaning into form.

---

All of the above examples addressed the use of physical modeling in terms of developing static solutions. A more intricate application of the latter is for testing the operability of kinetic structures or in other words the accurate performance of parameters allowing kinetic transformations. Santiago Calatrava is the architect that employs this method in many of his projects (Fig. 31). Prototype scaled models, which were first generated as mathematical systems, are used in verifying the entire deployment procedure of the structure and account for possible tolerance problems, as shown in Fig. 32. Furthermore, Chuck Hoberman tests of collapsible geodesic domes ultimately led to construction of full scale kinetic designs like the *Hoberman Arch* in Salt Lake City.

In summary, I believe that apart from more utilitarian benefits, the use of physical models will have a modest contribution to the contemporary discourse of parametric design. Today we refer to parametrics as the ability to make associations between elements that have their sizes and positioning defined by measurements. The very fact that in order to construct a physical model intended to depict this understanding one must build from scratch multiple iterations or foresee entirely the model’s behavior in case of kinetic capabilities already indicates to significant limitations. Furthermore, considering that some architecture today is done outside of the comfort zone of Cartesian geometry, using physical models in exploring the solution space would be less feasible from both a cost and time standpoints excluding the unique case of Gehry’s practice. This is why digital tools became of paramount importance when designing parametrically.
2.1.2 Digital environment

"Handcraft was once the tool of commodity, but today it is the machine." 29 "This technology provides a way for me to get closer to the craft. In the past, there were many layers between my rough sketch and the final building, and the feeling of the design could get lost before it reached the craftsman. It feels like I've been speaking a foreign language, and now, all of a sudden, the craftsman understands me. In this case, the computer is not dehumanizing; it's an interpreter." 30

Programming became a method for some architects to develop their own tools, in which the role of the architect shifts from a mere creator of single designs to a designer of tools that allow the development of multiple design solutions. This is a method regarded as generative and related to parametric design. Granted that it may be a powerful new way of approaching the problem of searching through the solutions space, the question is whether designers should be passively or actively engaged in programming. Those who use existing software are accepting the limitations embedded within the code and the graphical user interface. Still, few architects are interested in becoming skilled programmers. For instance, my earlier inquiries into the nature and constructability of compound, double-curved frame-based structures culminated in the role of a passive programmer when developing TekCAD (a java-based, mathematically enabled system that primarily uses generative principles) with a software engineer. This emerged as a measure to account for frustrations encountered when previously using conventional tools in similar studies.

30 Gehry, Frank. Quote from CenitDesktop website.
It was a collaboration in which my contribution as a designer was to provide a set of functionality with a simple and consistent interface that would appeal to the realm of architectural design. On the other hand, practices like Foster’s and Gehry’s have computer programmers not only as permanent staff but as essential members of their design teams. Nevertheless, I believe neither model to be accessible to the majority of architects considering their traditional lack of resources required for such undertakings to occur. Nor the majority has the background that would allow them to engage into programming.

This is why I consider parametric design to be a good alternative in terms of making use of existing computational frameworks to allow for advanced rule-driven design systems be developed. The architect now becomes a “half-programmer” as he elaborates computational models operated by various types of interdependent parameters and constraints often controlled through formulaic expressions.

Parametric CAD tools are relatively new to the architectural community and are based on the concept of constraints, features and associations between parameters or objects. These can also be called “feature-based” or “associative geometry” computer-aided design systems. Features are elements of an object such as chamfer, hole, pocket, etc. that a designer can use during product definition and adjust parametrically or eliminate at any time afterwards. The distinctive characteristic of parametric systems is in their ability to store in a sequential order all the operations defining the product model. These are saved in a database that is a tree structure (Fig. 33), which enables anyone at any point to join the design team and follow the logical steps of model creation. Furthermore, depending on the users’
techniques normally model’s components in the tree can have their parameters adjusted.

Although originally developed for the engineering needs, parametric software started slowly penetrating the architecture market in the early 1990’s. This was the result of a trend in terms of architects relying on technology transfer in their search of innovative approaches to design or in response to the inability of addressing specialized problems with conventional tools. Successes in aerospace or automotive industries are inspiring architects to ponder about efficiency, quality, and mass-customization — requirements emerged in the current hectic age. The new acquisitions for architects range from software satisfying visualization needs such as Maya (Alias) to the more recent parametric packages like CATIA, Solid Works (Dassault Systemes) or Inventor (Autodesk). Catia is best known among architects because of its implementation by Frank Gehry’s practice. This is probably the most versatile CAD tool currently available commercially and it offers an impressive array of functionality complemented by parametric capabilities. According to its creator Dassault “it allows manufacturers to simulate all the industrial design processes, from the pre-project phase, through detailed design, analysis, simulation, assembly and maintenance.” Its primary applications are in the aerospace and automotive industries. Despite extensive potential, CATIA’s main application at Gehry’s is as a post-design processing and cost-optimization tool. Its implementation came with the Disney Concert Hall project in Los Angeles when building the limestone cladding mock-up for the Venice Biennale. The design started by building a paper model illustrating the curvilinear character of the surfaces. The model was then digitized and the resulting coordinates were used in CATIA to rationalize the surfaces in order to achieve
similarity without sacrificing form. This was the beginning of a process characterized by a dialog between physical and digital models, where the digitized physical designs were refined in CATIA to then be used in manufacturing. Optimization was done through the means of parametric rules by operating an optimization program inside CATIA. This was used in automatically creating patterning layouts for the exterior envelope.

Currently we are witnessing an emerging trend, in which software developers are closely collaborating with architects on creating parametric tools specifically targeting the architectural community. As mentioned earlier, the implementation of parametric tools has been sluggish. First, the cost is prohibitively expensive for most architects, especially when considering the dominant position of small and often financially inapt practices that cannot afford time and effort on researching and adapting new technologies. Second, powerful computing platforms are normally required to run most of parametric software. Third, designing with parametric software implies a significantly different mode of thinking than the current paradigm, in which elements needing adjustment are recreated from scratch. Thus at Gehry’s practice a new unit has been founded called Gehry Technologies, which is collaborating with Dassault Systemes in creating an architectural version of CATIA. Furthermore, a similar collaboration is ongoing between Foster’s Specialist Modeling Group and the software manufacturer of Microstation – Bentley Systems. Robert Aish from Bentley is working on developing Generative Components – “a model-oriented end-user programming environment which combines direct interactive manipulation design methods based on feature modeling and constraints, with visual and traditional programming techniques.”

Parametric techniques were

---

recently used in Nicholas Grimshaw’s Waterloo International Terminal in London. The complexity of the site and such programmatic constraints as the overall length of a train determined the variation of spacing between trains throughout the terminal. By employing parametric software the designers were able to create conceptually similar “double banana trusses” and generate multiple real-time variations that were responding to the chosen constraints.

More recently Morphosis joined Bentley’s early development program of Generative Components, yet another attempt to involve architects in a role of passive programmers when developing parametric tools. Again, the architect’s main appeal in utilizing the software was reflected in the ability to generate design variations based on a series of rules that act as constraints.

In summary, when looking back at designing parametrically both in physical and digital environments it is important to distinguish commonalities and differences. These will assist in clarifying the value that each method brings to the discourse. Shared characteristics are the easiest to detect. Hence, the identification and manipulation of parameters or implementation of constraints make a parametric model viable in both cases. Furthermore, the thinking mode when designing can be applied to either method. Computational thinking, for example, is not solely associated with the digital environment but may well complement the physical realm as in the case of models generated through shape grammars (Fig. 35) or a geodesic geometry build out of plastic struts (Fig. 36). Here, variations are attained through the means of spatial relations and computational rules (i.e. platonic solid type or frequency in defining a geodesic), and physical models are used in representing these. At the same time differences are also notable. Most significantly, in a physical setting it is very hard to achieve the level of complexity of
parameter interdependencies possible in the software environment. These are often established through parametric equations and offer more coherent and intelligent implementation of computational tools in design. Hence, the added value of a parametric digital environment compared to the physical modeling environment or the conventional CAD counterparts is in the versatility of ways to create and manipulate geometric relationships complemented by the rigor of constraints.

2.2 Building a parametric system

2.2.1 Getting started

In this chapter I will talk about the procedure of building a parametric system from scratch. Parametric methodology in terms of using computational tools can be employed for two distinctly different purposes (see diagram in Fig. 37). First is the paradigm of practices like Gehry’s, excluding the case of engineering applications. As established earlier they employ CATIA not as much for its parametric capabilities but rather as a medium to refine and optimize designs. In this case parametric tools act as more sophisticated design representation methods and assist in such post-design activities as bidding or communicating the construct to the contractors (manufacturers and construction companies).

The second application is the domain of inquiry of this thesis, namely to utilize parametric tools in conceiving new designs by imbedding more intelligence in the design process. The presumed advantage is in the resulting meaningful variations that may not have been pictured by the architect, thus expanding the range and quality of his efforts. So, in this case a designer must first think of a starting point. What
does he need to consider before developing a parametric system? For designs to be developed there must first be a harmonization between disparate specifications, constraints, and design data. Since the latter are unique for every design task, it is virtually impossible to think about re-using a

**Strategy for building a parametric system**

![Diagram of Strategy for building a parametric system]

- **Problem definition**
- **Clarification of goals**
- **Design rationalization**
- **Design development**
  - Exploration of multiple designs;
  - Gradual reduction of the solution space to a single design iteration for further development / manipulation
- **Development of design strategy**
  - Top-down geometry control
  - Bottom-up geometry control
  - Bottom-up / top-down control
- **Identification and selection of initial constraints**
  - Consider types of variations sought
- **Rules, Laws, Formulas**
  - Expand the constraint mechanism
  - Creation of parametric skeleton(s) (2D, 3D or 2D+3D space)
  - Variations from parameter manipulation
  - Creation of geometry on parametric skeleton
  - Rationalization of chosen iteration
- **Site**
  - Geometric constraints
  - Performance goals
- **Choose implementation environment**
  - Build elements defining the design
  - Refine elements defining the final design

**Figure 37**: Building a parametric system diagram
previously built parametric model, which relied on a different set of data when being built.

The solution space can be limited or vast, depending on how the designer opts to implement the chosen constraints and rules. This can become a dilemma when intending to account for a multitude of possible solutions in a single parametric model. Designers often revise their thought process not only with every new project but also within the ongoing one. This is why a clear strategy for conceiving the parametric model must be developed in order to be at least partially in a position to change the course of the model development at any point. “When you define the problem in a certain way, you define the solutions.”

This strategy can be reduced to three main methods – through implementation of a top-down, bottom-up geometry control or a combination of both. Each of these will have a significant impact on the way the model works overall and what kind of adjustments will it afford. The top-down control method has a highly structured character as it requires a rigid organizational hierarchy of all components. These are normally built with direct dependence on the other elements and should one be erased or modified, the entire parametric design will break or update, depending on established relationships. The bottom-up method uses a less rigorous approach when it comes to hierarchical organization of the model’s components. These are separately created as independent entities and brought together to form an assembly. Unless specific relationships (i.e. constraints, dimensional dependence) do not define the artifacts of an assembly, the deletion or addition of elements will impact the rest of the model. Furthermore, the bottom-up control

---

method allows to individually adjusting components outside of the assembly, which in turn will update both the modified element and the entire assembly.

Once the designer is clear of the implications that the method he chooses will have on the design, then the next step would be identification and implementation of initial constraints. These can be a set of dimensional parameters that will define the future artifact, or boundaries, which the design cannot exceed. The boundaries can either be geometric and dictated by the need to respect, for example, code provisions (i.e. distance to the sidewalk), visually connect to other elements on the site by means of axes, etc. or in terms of sought numerical values such as the building’s total volume, area, etc. These can be summarized into constraints related to the design process and called conceptual constraints. Among them are:

- the site - circulation paths, the landscape, placement of neighboring buildings, their heights and the way these affect the space in the vicinity (i.e. cast shadow, etc.)
- the geometric constraints - length, width, height, etc. of the edifice and its components or parameters that define the overall desired geometric configuration
- materials used
- the design’s performance - lighting, material usage, amount of energy consumed, etc.

But one will also encounter implementation or technical constraints that are attributed to the computational tools used in setting up parametric systems. And this may result in a biased approach to a design problem because the software environment with all its strengths and shortcomings will dictate the direction of a designer’s exploration. “Designers often establish design worlds implicitly, through
their choices of design media and instruments." Unless he does not accept the limitations of the interface and functionality and employs scripting or programming to complement the exiting tool, granted that the latter allows it, the designer is bound to that environment. This implies that a conventional CAD user will initially have difficulties adjusting, for instance, to the CATIA environment simply because of the unique perception of possibilities that such CAD tools establish in a designer’s mind. Their semantic level is low because they merely act as digital drafting aids with very little of the intelligence of parametric systems. Hence, the formal or design-development language will differ significantly due to the different sets of options at the designer’s hand.

Based on the established constraints the designer can start creating the parametric skeleton, which serves as the foundation for the future design iterations. This can take multiple forms depending on design goals and performance criteria and be conceived in either 2D or 3D environment or a combination of both. In a parametric setting the skeleton can be made out of points, lines, curves, surfaces or Boolean solids. Each implementation environment has advantages and limitations and no method is better simply because every design task has unique variables to address. A car wheel, for example, would be easier to build in the 2D environment by first generating its representation as a closed profile which is then extruded in the 3D environment to form the base wheel (Fig. 38-41). In this case the skeleton is generated entirely in 2D because of the constraint mechanism being more versatile for the task. The CATIA 2D sketcher, for instance, provides a cyclical constraint mode, with which the designer is informed whether the artifact is under, over

---


**Figure 38-41:** Car wheel built from multiple 2D skeletons
or properly constrained\textsuperscript{34} – important features not obviously present in the 3D setting.

All these point to the rigor and discipline that can be brought to the architectural design process by properly constraining the model from its inception. The extent of constraints is not limited to the spatial relationships between elements and dimensional values that define them. Parameters can also be constrained by formulas, rules and laws (CATIA terminology) – features present in most of parametrically-enabled applications. Formulas are normally mathematical expressions denoting parameter dependencies (i.e. $A=B+C$ or $A=(B+C)/2$, etc.) Rules are more sophisticated mechanisms, as they introduce generative principles through basic scripting (Visual Basic in CATIA) and normally affect the elements in the model on a conditional basis (i.e. if $x$, then do $y$). Laws are used to specify a relationship in which a parameter is defined with respect to another. These are also based on mathematical expressions and are specifically used in the creation of designs with parallel curves (Fig. 42).

The actual design geometry is built on top of the parametric skeleton and generally is made either of surfaces or solids (see wheel example above). This establishes a dependence of the latter elements to the skeleton, which means that with parameter manipulation not only the skeleton will respond but also the geometry defining the design. But one must be very conscious of the way parametric variations will affect the geometry. In the case of complex, double-curved entities it is easy to run into a cusp, which will lead to the model breakage. This is why it is critical to build the skeleton by pre-rationalizing its implications on the geometry (i.e. instead of 3-point or more spline defining a profile, build it

\textsuperscript{34} see Fig. 27-28, page 33
out of several 2-point splines using tangency constraints at the ends).

Finally, once the designer agrees on a specific iteration of a design, only then it is feasible to rationalize it in terms of constructability. First, the limited processing power of existing hardware will greatly slow down the design process when running variations of an elaborate (detailed) model. Furthermore, there is a greater chance of having the model break because of hardware/software limitation or an unforeseen conflict between elements or constraints of multiple skeletons that may form the design. It is also critical to note that building a parametric model will be impossible without initially establishing an objective in terms of a design language. This can be supported by a set of sketch studies or mental annotations used to clarify a starting point.

### 2.2.2 Expectations

So, supposing that I am new to parametric methodology and have at my disposal all the design criteria (the program and the site) needed to start developing solutions, what is it exactly that I expect from a parametric model in this context? Also, to avoid the danger of a biased thought process, at first I will speculate with principles free of implementation constraints. These expectations will be measured later against the experiments’ results.

By creating a parametric system I anticipate to be able to investigate programmatically the solution space. This means that a tandem between the architectural program and a set of initial constraints should help me set up a basic skeleton of the design that would serve in defining a broad range of meaningful functional and geometrical configurations. I expect to be able to explore in a reasonable amount of time
as many potential and contrasting solutions as possible. This computational model should allow radical changes in the design in terms of overall configuration (structure, exterior and interior), the spatial location of the program’s components (i.e. spaces representing certain functions within a building) and an appropriate response of the overall design to adjustments or introduction of new constraints. The effort invested in developing the parametric system should be reasonable in the generally accepted meaning of this term. Furthermore, the model would allow at any time additional layers of information to be added without breaking the intelligence of the entire system. This would come as a measure to account for the natural process of design development in which ideas evolve in time and require modifications to the model in progress. By gradually introducing additional constraints I expect to identify a more meaningful segment of the solution space for further investigation. In a parametric system the design process is guided in part by a network of parametrically controlled interdependent components, which makes the optimization procedures play an important role in achieving the proposed design requirements.

In the end, this model will ideally become a “living” organism of components that work harmoniously to reflect any adjustments in the design data and serve as an important assistant in the design process all the way to design implementation stage. Again, in the experiments to follow I will test these suppositions.
CHAPTER 3: EXPLORATION

3.1 Early explorations

My initial studies directed towards familiarization with parametric method of designing were done in the context of several workshops at the Massachusetts Institute of Technology. These addressed different topics but the underlying goal was similar – to develop a cyclical process of generating and evaluating meaningful variations of performance driven designs. All experiments were completed in CATIA V5.

3.1.1 Examples

The first two examples resulted from my research position at the Media Lab in preparation for the design workshop that followed. The latter had the aim to create a highly personalized car that would take full advantage of the 21st century resources in terms of networking, new engine technologies (in response to environmental concerns), customizable exterior and interior interface, etc. This became a collaboration between MIT, Frank Gehry’s office and General Motors. My role was to investigate ways of building a parametric skin of a car that would be assembled later with the parallel study of a generic parametric interior (see ergonomic model in chapter 3.3). The objective was to attain the ability to harmonize these two separate efforts by means of parametric manipulation.

The exercise commenced by following an approximate visual goal as a starting point and at a moment when I had just an introductory perspective on how parametric design works. Having had extensive modeling experience with conventional CAD tools, my thought process was biased

Figure 43-45: parametric skeleton
from the very beginning. I started by sketching a lateral profile intended to serve as a guide when constructing the parametric skeleton—something one would do when making a conventional 3D model (Fig. 43). Also, the assumption was that the car would be a roadster, which pointed to approximate proportional conventions already established by existing vehicles of the same class. The skeleton that emerged was very simple—a literal box. It consisted of five main pairs of control elements that were used in accommodating a secondary set supporting the initial guiding profiles. The controls were the wheelbase length (red lines), wheelbase width (yellow), front height (green), back height (cyan) and front length (orange) in Fig. 44. Then, because of the symmetry for half of the design I’ve developed the secondary skeleton made of multiple individually-manipulated parameters (red lines) defining the control points of the surface guides (yellow) in Fig. 45. All these elements were constructed with dependence on the main skeleton, which would make them responsive to changes in parameter values of this system. Based on the surface guides both an approximate frame and the detail-free car envelope were constructed (Fig. 46). Some of the parameter manipulations are shown in Fig. 47-49, which were exaggerated for the example’s clarity.

The next attempt resulted in a more sophisticated approach as it employed multiple inter-dependent skeletons developed two-dimensionally in different planes. It was based on an integrated system of constrained elements that responded to the direct manipulation of several main parameters. This design exercise started without any particular visual goals in mind but rather with the intent of developing an array of solutions that would emphasize the compactness of the car’s footprint as well as investigate alternative seating configurations. Thus, the form evolved out of the process
of building the parametric system driven by performance goals.

In this example, the chassis, which were the initial constraints, served in establishing the direction of the design, controlled by two main dimensional parameters—wheelbase length (red line) and width (yellow) in Fig. 50. Note that the symmetry of the car pointed again to developing only half of the chassis skeleton, which allowed later on to mirror its solid representation built from this skeleton. The geometrical elements defining the wheelbase length and width were constrained to the origin of the working environment. This means that with any change in the numerical value of these parameters, the elements will respond in relation to the origin. In other words, the reduction or increase of the element’s length will occur symmetrically in both directions from the origin because the lines representing the wheelbase length and width had their staring points at the same level as the origin (Fig. 50). The distance to both end points of the wheelbase length, for instance, was defined by means of a formulaic expression, which is obviously the numerical parameter for wheelbase length/2.

A series of smaller skeletons established the other elements of the design. Similarly, the profiles representing the seats (Fig. 51-52) or engine blocks were using the same constraint (origin) and a formulaic expression defining the dimensional reference of the profile to the origin. The formula consisted of the wheelbase length parameter plus/minus a numerical value defining the logical positioning in relation to the body—something decided by the designer and based, in this case, on functional needs. Furthermore, each of these separate skeletons is fully constrained in terms of having all its components numerically controlled. Thus, elements that may be perceived as independently manipulated are actually
directly dependent on the value of a main parameter (wheelbase length in this case). Note that these skeletons are created in a different plane than the chassis. Furthermore, the control points employed in building the reference system for the envelope used a parametric ratio distribution on the chassis’ skeleton. Thus, the auxiliary system composed of primary (green and cyan) and secondary (blue) surface guides ultimately defined the vehicle’s envelope (Fig. 53). Consequently, when main parameter values are modified, the entire design works in unison to reflect the change (Fig. 54-57).

The third example was a result of the workshop organized with Foster and Partners and called “Generative and Parametric Tools for Design and Fabrication.” The goal was to investigate multiple temporary art pavilion designs by employing the generative or parametric computational methods along with rapid prototyping techniques.

This exercise was more intricate as it used multiple parametric controls driving both the design’s performance and its configuration. Furthermore, new conceptual parameters were introduced in addition to the geometric ones. These were in form of constraints of an existing site (Boston Museum of Fine Arts’ interior court) and its performance in terms of maximum exposure to daylight. A digital study of the daily and annual sun path performed in specialized software called Ecotect, determined the location and preliminary characteristics of the structure, intended as an outdoors exhibition pavilion. The performance of the structure was measured in its ability to manipulate through a set of mechanically controlled louvers the indirect lighting in the interior of the space.
Initially, a primary reference system consisting of three arcs (each subdivided into four equal segments) was constructed with the intent to define the pavilion’s plan definition and height (Fig. 58). The configuration was driven by two unique sets of parameters attributed to each arc – the length of the segments and the angle between them (Fig. 59). This allowed for an extensive series of variations for the footprint alone to be investigated as the configuration ranged form a straight spine to a regular or irregular wave form. Furthermore, the number of variations was extended by manipulating the overall length or that of individual sections (arcs) of the pavilion by means of the three length parameters.

The height was controlled by a secondary independent parametric mechanism - a dummy system that had the mere role of controlling the pavilion’s height. This consisted of a number of segments equal to those defining the pavilion’s height in the main skeleton (Fig. 58). The length of each segment of the main system was constrained to the one defining the same position in the secondary system. This time instead of numerical values I used parametric equations to regulate the height. Thus, I realized that the pavilion’s height configuration obtained from a sin or cos curve would serve well in visually defining the location of the hypothetical artwork to be exhibited and become an indication of where the louvers should be placed for indirect illumination of the works. Through one parameter only I was now able to fine tune the height in several ways by changing the values of the amplitude in terms of maximum displacement of the wave and its frequency.

A separately developed pavilion profile (Fig. 60) was inserted at end point of each of the spine’s 12 arcs to form a skeleton for generating the enclosure (Fig. 61). Some of these profiles ended up taller than others, depending on the
auxiliary element defining the height at the profiles insertion point. Ultimately, these profiles acted as surface guides. A series of parameter manipulations are illustrated in Fig. 62-65.

I knew that establishing the skeleton by means of the above-described set of parameters will give me the flexibility to adjust the pavilion’s configuration quite dramatically. This was possible at any point in the design process, granted that I was able to clearly reason the implications of any such variations, especially when observing the relative complexity of the emerged skin. But at the same time I wasn’t limited to merely the shape I had, since at any time the system I created allowed me to replace the profile with a new one and thus generate a substantially different design language. The skeleton was directly responding to the site in terms pavilion’s potential performance by capturing most of the day light. Despite countless ways of setting it up, in this case having had a configuration other than linear (elongated) would have obstructed the best performance scenario because of the building’s shadow.

Finally, the third parameter controlling the structure’s performance was attributed to the sun, the movement of which defined the positioning of the louver system constrained to it. This constraint meant that louvers always faced the sun by following its daily movement and thus maximizing the diffusion of light in the pavilion’s interior. I constructed the louvers manually on the structure’s envelope after a careful selection of potential art pieces that indicated the required proportional conventions for the louver’s opening and the blade size to avoid interference (Fig. 66).

Obviously, the surface model in this state did not bear too much meaning because of the uncleness of how to build
it. This is why rationalizing a construct is critical for a parametric system to have a greater meaningful impact on the design process than merely generating variations, even if these are important from a functional and aesthetic standpoint. As mentioned earlier, in most cases it is recommended to reason a single chosen design variation (Fig. 67). I selected a section (Fig. 66) and analyzed it in terms of constructability. First, rapid prototyping techniques such as 3D-printing and laser-cutting served to visualize physically a range of solutions for the same segment of the design at a small scale (Fig. 68) to then finalize the study with a 1:10 cardboard mockup representing one of the possible meaningful ways to address construction (Fig. 69) built from a rationalized digital model (Fig. 70).

3.1.2 Conclusions and issues

Now, looking back at the first two examples and measuring the outcomes against my earlier expectations, I concluded that both can be regarded as failures. In the first model, despite the apparent regularity of the parametric skeleton the presence of fifty individual parameters defining the surface guides’ skeleton made the investigation of variations cumbersome. Even if all these were using relational dependence to the main skeleton, it was very difficult and time consuming to trace in the tree’s long list the right parameter for manipulation.

Furthermore, the first model was driven exclusively by dimensional parameters in tandem with a single main skeleton, and its control mechanism built entirely in 3D environment. The implications are obvious, as constraining in this mode generally leads to less powerful outcomes. The resulting variations were either easily foreseeable, as in the

---

35 see Fig. 33, page 42
case of varying the main skeleton's wheelbase length or width, leaving the surface guide parameters intact (Fig. 47-48). This means that the results are not topologically different from the starting design iteration, making such variations part of a limited solution space. Or, it was pointing to a time consuming process of adjusting multiple surface guide parameters to achieve a meaningful and substantially different solution, thus undermining the significance of using a parametric computational platform because of the extensive "manual labor" involved in the latter approach.

To design parametrically in a computational sense means the ability to rely on designer-conceived systems that semi-automate the design process through constraining variables. In other words, taking full advantage of parametric methodology suggests basing the design on a skeleton that can afford "to define, determine and reconfigure geometrical relationships." The second model, even if it used a more intricate parametric structure of multiple skeletons and a combination of 2D and 3D environments to build these, the outcomes were very much similar to the first case. The established geometric relationships did not lead to creating new solutions but rather interpret the already developed one. It became obvious the importance of choosing the appropriate parameters and defining their relationship to determine the range and meaning of a solution space.

Through its outcomes, the third example was closer to supporting the understanding of a more efficient use of parametric methodology. The system became further meaningful with the introduction of performance constraints in addition to the geometric ones. A combination of independently controlled but interrelated parameter skeletons allowed for a relatively wide range of solutions to

be rapidly investigated in a semi-automated manner. Now, the manipulation of a single parameter (i.e. arc angle) had a more significant and reasoned effect on the overall design. The results were more encouraging this time because of a greater flexibility of the parametric skeleton, which made possible the consequential addition of elements without hampering the well-functioning of the model when adjusting parameters. Furthermore, numerical values were not the only attributes of parameters. The introduction of parametric equations brought an additional layer of reasoned complexity to the design mechanism. Despite the pavilion being made of compound surfaces, overall the example still had a predictable character, which pointed me to exploring the idea of “reasoned ambiguity” in parametric design. I realized the need to balance the excessive skeleton regularity of any of the examined explorations with an element of controlled ambiguity to attempt to support an initial claim for parametric methodology to act as a complementary tool in the design process. Does “reasoned ambiguity” mean randomness or complexity? What are the dimensions of employing ambiguity without loosing meaning of the emerging designs? The next set of experiments is meant to address these issues.

3.2 Experiments

The initial studies identified a substantial drawback of parametric design – the rigidity of the environment and the need to pre-rationalize the design, which “appears to be the enemy of intuition.” Also, some new avenues for inquiry were suggested. The following three experiments will attempt to address these issues in a purely architectural

---

37 see Chapter 1.2
design setting by means of different parametric implementation methods. The major question pursued is to what level can parametric design be employed without restricting the designer? In what form should the skeleton be built so this restriction does not occur?

3.2.1 “Reasoned” ambiguity

“Topological variations are very hard to implement in dimension-driven parametric systems like CATIA.” A starting point for this experiment was to challenge this limitation and attempt to extract substantially different solutions from the same skeleton. This meant defying the conventional ways of conceiving a parametric skeleton, the manipulation of which normally leads to predictable outcomes. The idea behind “reasoned ambiguity” was for the designer to conceive a parametric mechanism that would generate a meaningful range of possible variations but still retain an element of “surprise” for the results to come. This way the designer designs the tool to generate ideas and employs his intuition in choosing the design iteration that he finds fit for the implemented constraints.

The first step was to choose a site for the previously developed hypothetical architectural program. The site was on Vassar Street next to Simmons Hall, the MIT campus (Fig. 71). The program was developed for a multifunctional facility and intended to house the following:

1. game room – 150m”
2. bar – 100m”
3. lounge / café
   - kitchen – 50m”
   - dining area – 150m”

4. mini market
   - goods area - 500m"
   - storage area - 150m"
5. circulation - 300m"

**total area:** 1450m"

A 5% deviation from the total suggested area was allowed. The height limitation - 10m

I started by first treating the future design as an assembly. This meant that all major entities (i.e. site, building, etc.) would be conceived as separate but interrelated groups of components for purely organizational reasons. Constructing the site may seem a straight forward procedure as it was a mere representation of the existing condition and CATIA being a parametric tool greatly facilitated its implementation because of the versatile constraining mechanism in its 2D environment (Fig. 72).

All the same, it is critical to be very clear of the constraining procedure in order to account for the sought results. When constructing the site, at first I thought this was not as important because of the absence of variables. In other words, I did not expect any variations to be needed. Therefore, I have constrained all the elements in reference to the origin (Fig. 73). Then I realized I needed to be able to move the entire site in reference to the future building. This meant the constraining process I used was invalid because when I modify the value of x, for instance, only the vertical line constrained through this parameter will be moved unless there is a coincidence constraint at the junction with the horizontal line (y), which will make it also be displaced horizontally. I had to re-constrain the model following the principle shown in Fig. 74. This time all the elements were

---

40 See Appendix B for types of constraints.
constrained to a single object (orange building, Fig. 72) that in its turn was constrained to the origin. This allowed horizontal moving of all inter-constrained components through a single parameter (Fig. 72 – yellow highlight).

Moving on to the design part of the experiment, the challenge became to construct the skeleton and avoid too much predictability of the outcomes of the parametric manipulations. In order to achieve a meaningful element of surprise, the parametric system needed to contain either a high level of interdependency among its parameters, making it hard to see the implications of changing their values or be described by some random functions, which would operate within a previously reasoned range.

The creation of the skeleton was based on the initial constraints derived from the site – the strong axis of Vasaar Street supported by Simmons Hall, as well as the secondary axes – the pedestrian path linking the site with the nearby park and the small street leading into the residential area (Fig. 71). Note that the site contains additional elements that can serve as potential constraints (i.e. neighboring residential buildings not parallel to other axes, etc.) and the skeleton must have the ability to respond accordingly. The latter consisted of a series of ten tangentially inter-constrained circles, which had their radiuses driven by four different parameters and intended to support the elements defining the outline of the future building. Each parameter had a value range of 1m to 5m with an incremental step of 0.25m (Fig. 75), which I derived from a quick inquiry into how their extreme values would affect the building’s outline. The nature of the site and the area of the program influenced the overall positioning of the circles in reference to each other. Furthermore, additional parameters were introduced to allow the entire design to either move vertically,
horizontally or be rotated in reference to the site (Fig. 76). The latter parameters allowed at any time to satisfy the above-mentioned additional constraints (i.e. buildings), as well as compensate for extreme variations that may need positional adjustment within the site.

The next step was to address the sought reasoned ambiguity issue. For that I created seven types of circles (Fig. 76), which had their radiuses constrained through simple formulaic expressions. These denoted combinations between the four radius types (i.e. \(\text{Radius}_3 + \text{Radius}_1\), \((\text{Radius}_2 + \text{Radius}_4)/1.5\), etc.) (Fig. 77). Note that two interior circles were not dimensionally constrained in order to avoid over-constraining the system. Being tangentially constrained to the neighboring elements, their radiuses constantly respond to the changes in any of the four main radius parameters. Because this is an integral system, the value change of one parameter will propagate through the entire design. With 17 increments in each radius parameter I was conscious of the large extent of emerging variations that were not fully predictable but were part of the previously rationalized meaningful range. So, what I created was a system that started responding to my initial goal.

The skeleton served in defining the building’s footprint. For this I created a series of extremum points on each circle in reference to either the x or y plane and used them as control points for the outline. This meant that each time variations occurred, the extremum point assumed a new position on the circle that was closest to the plane that defined it (Fig. 78). This obviously changed the configuration of the building’s footprint – yet another controlled but simultaneously ambiguous process due to the difficulty of seeing the emerging shape (Fig. 77). Note that the footprint
can take any form despite using the same skeleton and control points. Again, the designer chooses the design language by employing elements that give a distinctly different character to the solution (i.e. lines vs. splines).

Similar to the previous example, this parametric system was based on multiple skeletons created in different planes but related through common parameters. Hence, the roof profile made of a parametrically controlled spline allowed various configurations and heights to be efficiently explored (Fig. 79, 84-87). The extrusion of the roof profile and a scaled intersection of the footprint defined the roof and the parapet, which allowed generating a preliminary building envelope (Fig. 80). Further skeletons were developed to integrate performance constraints into the design. Solar panels seemed most appropriate for the emerged roof and the first step was to develop a skeleton that would define the location of the panels. The latter consisted of a series of parallel lines projected onto the roof (Fig. 81). The spacing of these elements was controlled by a single parameter. The number and the length of the lines were designed to account for the building’s extremes in terms of length or width. In other words, when all the radius parameters are set at the maximum value of 5m from the rationalized range (Fig. 75), then the footprint will not exceed the boundary created by the lines defining the positioning of the solar panels. This was a simple and specific way to address the issue of automated emergence or deletion of elements in a parametric system.

The additional skeletons defining the solar panel and the track on which it would move to follow the sun for maximum performance were again created with dependencies on other skeletons. Hence, the track in Fig. 82 was constrained to the intersection of the panel guide and the roof profile, which will make it follow the latter when its parameters are
changed. Similarly, the skeleton of the solar panel was constrained to the track and a system simulating the daily sun path, defining its position in reference to the sun (Fig. 83). Note that at any point the designer can go back and modify any of the profiles without breaking the established hierarchical inter-dependence, thus gaining the ability to make changes to the design language by using the already defined skeletons.

![Figure 84-87: variations derived from manipulating roof profile parameters](image1)

![Figure 88-89: solar panels follow the sun; Figure 90-91: automatic emergence and deletion of solar panels](image2)

In the end, having the comfort of manipulating just a few parameters I was able to obtain a substantially wide range of solutions. The four radius parameters alone gave more than 83,500 variations, most of which were quite meaningful. With this parametric mechanism I was able to search for solutions either randomly or in a systematic manner, as shown in the studies above. Furthermore, based on the architectural program and the emerging building footprint which had the area calculation attached, I was able to quickly assess the potential spatial solutions in terms of number of floors and distribution of functions within the building.
3.2.2 Parametric “freedom”

The goal of this experiment was to attain a more flexible parametric mechanism for multiple useful definitions of the architectural program and its consequent manipulation. Since the previous study successfully addressed the issue of “reasoned ambiguity” in the emerged building envelope but put a lesser emphasis on the detailed implementation of the architectural program, this exercise was meant to test a parametric system driven primarily by the latter in an environment that permitted straight-forward manipulation of the program’s components (i.e. rooms) and avoided the usual rigidity of parametric systems.

The framework this time was much simpler than in the previous example. I started by creating a two-dimensional representation of the program’s components. These were individual dimensionally constrained parallelograms matching the required areas that could be freely moved around for investigating various functional solutions (Fig. 92). This also meant that the shape of each component could quickly be adjusted through manipulating its parameters but acknowledging the required areas. In the end this became a conglomerate of individually constrained entities that were built in the same plane but not inter-constrained.

Next, a dimensional parameter per each program component was created to define the height of the latter in reference to the ground level that matched the plane where these components were first generated (Fig. 93). This would allow individual program pieces to either be spatially located at the ground level or raised to the desired height. The heights were represented by lines built at the center of the parallelograms. Note that there are three types of heights

41 See page 63
shown in different colors for clarity. From the program I estimated that the mini market and its storage area will always be at the ground level for accessibility and servicing reasons. This is why these two elements alone had the height lines represent the actual heights of the future volumes and not their vertical position (Fig. 94 – black lines). The white elements, on the other hand, served as the vertical repositioning mechanism for the rest of the program’s components that were intended to allow such manipulation. The third type (shown in orange) was meant to control the height of the volumes that had their vertical position adjustable. In the end, there was a relatively small amount of control parameters to manipulate, which made the exploration of variations simple.

The upper end points of the height lines were used to define planes, which served for projecting the footprint of the rooms to the levels indicating the extremes of their future volumetric representations (Fig. 94). In other words, identical representations of the game room’s footprint, for instance, were generated by projecting the original profile onto the planes that defined the lower and upper levels of the volume. This established a dependence of the new elements to the driving height lines. Hence, whenever the height parameter defining the position of the volume was changed for any of the components, the dependent planes and the subsequent projected elements responded accordingly. Hence, this simple parametric skeleton of interdependent elements was built to function in a more rigorous manner only vertically. But looking back at the initial setup of the program’s components (Fig. 92), it is important to also remember about the direct dependence of the all the elements described above on the dimensionally constrained parallelograms. Whenever a parallelogram would have its configuration changed, then this would lead
to the modification of the dependent elements (i.e. projections).

Now that the variations mechanism was working, there was a need to reason the construction of the future building envelope. The dilemma was how to build an intelligent skin that would be responsive to the extreme movement of the program’s interior components (i.e. bar moved from front of the building to back or raised from ground level to a particular height, etc.) This meant that if the envelope were to be built out of a series of splines dependent on the emerged volumes, then the model would break because of the cusps that such an extreme move would lead to.

The solution was to build an “intelligent” reference element on each side of the building that would always remain on the same side despite extreme moves of the program’s components. These would serve in supporting the envelope’s guide splines. Such elements were generated from four “rubber bands”\textsuperscript{42} that connected with continuous polylines the same sides of all room profiles (Fig. 95-98). This allowed for whichever profile margin was the farthest out in the chosen direction to generate an extremum in form of an emergent element coinciding with the edge of the farthest room (Fig. 99). In other words, the extremum was not bound to a particular room profile but rather was freshly generated each time a new room had its margin take the outer most position. This meant the size of the extremums was variable considering that all parallelograms were proportionally different. Points built on these extremums also gained their properties by jumping to new locations when a new extremum was generated. Thus, the first four extremums served in supporting the bottom control points of the envelope’s guide splines.

\textsuperscript{42} concept by Axel Kilian
A similar procedure based on "rubber bands" was employed in determining the extremums at the two levels of earlier described projected profiles. Since I decided the envelope's guide splines to be built out of three control points for stability reasons, the second and third layer of extremums made the latter possible (see yellow extremums and blue and orange spline guides in Fig. 100). Finally, the emerged envelope used tangentially inter-constrained segmented surfaces for obtaining face continuity (Fig. 101-102).

Obviously that this example is only an approximation of a building but it clearly illustrates the capacity of designing by freely modeling the program components. It pointed to the ability of obtaining a set of sophisticated surface manipulations from a fairly simple and efficiently built parametric system. Note that the interior shapes are simple volumetric guides, as the actual interior solution can be further developed from a chosen design iteration.
3.2.3 Rules and emergence

This last example in the series of late explorations was primarily to address the "drawback" encountered in the "reasoned ambiguity" study (Chapter 3.2.1). The emergence or deletion of solar panels, as you recall, was solved but the issue was the manner in which this was done. Because I rationalized the extremes to which the potential designs extended, I created a skeleton that would simply cover all possible scenarios in terms of the number of needed solar panels. In the end the roof acted as a magnifying glass showing only the panels that fell into its footprint and through a Boolean mechanism was cutting or eliminating the unnecessary parts or whole elements. A different methodology of dealing with the issue of emergence or elimination of elements in terms of user interface is discussed in this chapter.

The example in this investigation is more hypothetical than the previous ones and does not follow the proposed architectural program but rather is an approximation of a structural solution complemented by an envelope that is proportionally appropriate for the earlier shown site (Fig. 113).

The idea behind the experiment was to have a parametric system that would either eliminate or add structural elements depending on a set of rules established by the designer. The starting point was to create a pavilion-like structure with a roof supported by equally spaced (6m) peripheral structural elements (i.e. mullions) and interior braced columns also equally spaced in the direction of the mullions but at twice the value (Fig. 113-114). Note that the roof was created at a constant decreasing slope, which meant the building was limited in its length. If compared with the previously
discussed examples, this time all the elements were based on a series of points with fixed coordinates that were not parametrically related to other components in the structure. The points allowed for mullions, columns and roof to be created. For instance, each of the mullions used a coincidence constraint to fix the profile to the point (Fig. 115). The fixed character of components was chosen because of the possibility of working with known structural systems, which employ well defined structural conventions (i.e. spans of elements) that would not be affected by changes in the design language.

Now, looking at the skeleton of the structure, the presence of elements not found in one of the many possible design iterations shown in Fig. 113 can be noticed and emphasized with orange color (Fig. 116). The version in Fig. 113 represents what I designed to be its outermost solution in terms of pavilion length (36m) and width (18m). This means with the chosen structural system (6m x 6m spacing) it consisted of seven unique pairs of peripheral mullions with four unique columns, the height of which was determined by the parametrically controlled slope of the roof. The configuration of the columns also depended on structural assumptions related to the pavilion’s width. Hence, a series of rules were established to determine the columns’ configuration depending on the span these had to support. For the largest extreme shown in Fig. 117 the established rule was to have two equally spaced columns 6m apart forming an integral structural system through bracing the roof and the side mullions. In other words, the rule stated that for every 6m of span a new column must be generated. The same rule applied to Fig. 118, although a slight modification occurred. Because the second column was now gone, the horizontal bracing between the two columns needed to be eliminated. In Fig. 119, since the overall span
is 6m, no column or bracings were necessary. Obviously, the rules needed to expand to cover the cases outside of a mere 6m stepping. Fig. 120 & 121 are just two out of many such possible cases. The rule established that if the span was >6m but <12m, then the outcome would be similar to the case of the 12m span (one column with two bracings) but one of the braces would have a variable length. Similar conventions were established for spans between 12m and 18m, and >18m.

Now, how could I implement these rules into a parametric model? The method I illustrated in the example addressing emergence of elements would not apply in this case for obvious reasons. Even if the structure may seem simpler, it requires a much more complicated transformation mechanism in terms of appropriate emergence or elimination of elements, especially when considering that the building is variable in both length and width. This means that another set of rules needs to act in parallel to the first set to determine the number of mullions and columns in the length direction of the building. So, with a conventional approach to parametric design solving this issue would be very hard if not impossible. This is why an automation procedure is best obtained through employing a generative method, or in other words either scripting or programming.

Hence, all the rules were explicitly established in form of a Visual Basic script running within CATIA as a rule (CATIA terminology). The simple VB code did nothing other than include or exclude elements (columns, bracings, mullions) that corresponded to conditions established in the rules. Note that Visual Basic script and the rules are illustrated in Appendix A of this thesis.
It is now important to mention the logic in writing the code. Please refer to Appendix A, which also includes the initial ("incorrect") version of the script. This code was unsuccessful because of the logic I initially used, in which two conditionals were setting two separate sets of things. In other words, the first rule was turning elements off, while the second was turning them on. The problem was that conditionals were used in series rather than in parallel and they were not mutually exclusive, so this yielded incorrect results.

Consider a truth table in which there are four states for AB: FF, FT, TF, TT. (T is true, F is false). If the actions based on these truth values are mutually exclusive, no conflict will arise. If they aren’t, a conflict could appear, as it was the case of my first code. So, what I needed was to implement a logic like this:

```
if (A & B)
    do X_1
else if (A & ~B)
    do X_2
else if (~A & B)
    do X_3
else if (~A & ~B)
    do X_4
```

This makes rule interaction complicated. Unfortunately, that’s why simple parametric techniques fail. There are two basic alternatives:

1. the designer can make the rules depend on the current values: start with all the activities set to true, and set them false when a condition applies. No condition can set a false object back to true.

2. "make the rule integration code smart: have meta-rules that change the rules based on the situation. This is the
general problem, but it is a "hard" problem to solve generally."\textsuperscript{43}

The final code implemented the first of the above alternatives. Of the many types of conflicts that can arise in a system, this is the simplest type. From the aspect of an element, rule one says do one thing, rule two says do another. The easiest way to solve this is to make a decision – only set the element to true if both conditions are true, otherwise set it to false (see Appendix A).

\textbf{Figure 122-133}: design variations

The series of variations in Fig.10-21 clearly illustrate that the rules will apply to elements that are parametrically manipulated and result in different configurations. By manipulating a small set of parameters (base length and width, roof slope, plane angle for mullion orientation) I was able to obtain a highly reasoned set of variations.

In the end, what does this exercise prove? It clearly shows that a procedural combination in terms of a parametric system that has its elements controlled through explicit rules but defined by simple programming techniques such as scripting may lead to powerful reasoned methodologies of approaching an architectural design process.

\textsuperscript{43}Anderson, David. CTO, TekStar (Software company). Quotation from a discussion held in March, 2004.
3.3 Alternative ways

At this point it is obvious that there is no limitation in the ways a parametric system can be built. Its structure directly depends on the rules the designer chooses to implement and the expected results. The earlier examples I have explored were meant to point to some generally accepted techniques rather than act as guidelines to how a parametric system should be built.

Before proceeding to discussing the results of the late experiments, in this chapter I will give a quick overview of two additional examples to provide a perspective of how others are employing parametric techniques for distinctly different purposes. The first case study comes from the earlier discussed Media Lab - Frank Gehry - GM workshop and represents the ergonomic test model for the interior of the car. This is the generic parametric interior intended to be assembled with the envelope study from Chapter 3.1.1.

The model was prepared by a mechanical engineer and based on a diagram developed by the automotive industry (Fig. 134). This diagram was a methodical consideration of ergonomic principles that denote comfort angles or ranges of vehicular floor planes for virtually all vehicle types – from race cars to agricultural equipment. So, by simply transferring the data into a digital, fully constrained parametric model of the interior, an interactive, highly efficient search process was made possible through automating the variations procedure. Now, the multiple variables that can also be calculated manually through a time consuming and meticulous process were integrated into a system that was functioning harmoniously due to proper constraining and associativity of elements. This example

---

44 Will Lark. Graduate student, MIT Media Lab.
shows the advantage of parametric tools in simulation-related tasks.

And finally, the last case study will address the importance of parametrics in successfully solving particular architectural problems. This quick overview is based on an example from Dennis Shelden’s PhD thesis. He discusses the use of parametric modeling in tackling the practical problem of variations applied to rationalized complex curvilinear surfaces. For this, the Museum of Tolerance roof system was chosen for analysis.

Since a curved surface can be easily constructed out of triangulated facets (i.e. British Museum courtyard by Fosters – Fig. 139), the author was interested in the alternative of rectangular glazed panels for fabrication and cost savings considerations – a study initiated and promoted by the office of Schlaich Bergermann and Partner and most notably Dr. Hans Schober. But using this structural system raises serious challenges in terms of design freedom as it “imposes a substantial constraint on forms that can be constructed.”

Employing a quadrilateral tessellation as structural solution requires building the curved surface out of planar flat sections. This means the geometry should conform to a specific set of curved shapes generated through extruding without rotating a directrix along a generatrix (Fig. 140). A directrix according to Websters dictionary is “a fixed curve traversed by a generatrix in generating a conic section or a cylinder.” So, the quadrilateral grid will work only on translation surfaces constructed through two splines as long as these generatrix splines are coplanar and identical or scaled, which extends the range of possible variations.

---

The Gehry office was already relying on geometric scripts to deal with similar cases but the downside of this approach was the need to "re-apply these on any modified input geometry." This is why parametric design was seen as a solution. By parametrically controlling the height of the directrix and generatrix curves, all subsequent variations resulted in rationalized solutions in terms of fabrication (Fig. 141-146). Shelden argues that this method could facilitate solving many of the firm's constructability issues.

Figure 141-146: parametrically manipulated structurally rationalized surface. (Dennis Shelden PhD thesis, MIT, 2002.)

---

CHAPTER 4: REFLECTIONS AND CONCLUSIONS

4.1 Reflections on the exploration results

As revealed earlier, the three late experiments described in Chapter 3.2 were meant to identify ways of dealing with the apparent problems that emerge when parametrics is employed as a design methodology. It is clear that none of the examples can serve as a definite solution and they merely address aspects of the earlier encountered limitations. The skeleton rigidity and excessive predictability of the outcomes were the major drawbacks of the early examples and of simple parametric systems in general. These shortcomings were later successfully challenged.

The proposed "reasoned ambiguity" concept pointed to the necessity of achieving in a parametric system partial unpredictability of the emerging variations. This allows parametric methodology to act as a co-designer based on the system the human designer develops and manages, granted that such "collaboration" must generate meaningful results. To accomplish this, the designer must still pre-rationalize to some extent the behavior or the range of variations that the parametric skeleton affords. So, there must be a balance between the reasoned mechanism and the unexpected results that it generates.

Obviously, the dimensions of "reasoned ambiguity" can be quite extended and depend on how the skeleton is conceived. In the example addressing this issue, besides ambiguity derived from manipulating the main skeleton parameters, such results were also generated by the secondary parameters that either rotated or moved the skeleton. This caused unexpected building footprints to emerge and thus expanded the variations mechanism (Fig. 147-150).

Figure 147-150: variations resulting from the rotation parameter
In terms of meeting the expectations from Chapter 2.2.2, this example was mostly satisfactory. The system allowed multiple layers of information to be added through element associativity (i.e. solar panel skeletons). Furthermore, I supplemented the model with a secondary constraint – an array of lines pointing to a single location (a flag in the adjacent park, Fig. 147-150). The idea was to maximize the view of the interior to that point through a series of louvers built on the envelope and responsive to the location of the flag – a hypothetical goal. So, all the elements were associated with the main skeleton (circles), which meant whenever the building changed its configuration or location, the louvers would respond as well by always keeping the maximized view. This proves that parametric systems afford various types of new constraints be added without much difficulty.

In the first example, the skeleton rigidity problem was compensated by the mechanism that resulted in reasonably ambiguous variations. But I have determined that another way to avoid the issue is by carefully conceiving a system that is not entirely constrained and allows enough maneuverability of its components. The second experiment clearly indicated that such systems are possible, although the shortcomings of the case are also evident. First, there was a limitation in the automation mechanism itself. The emerging extremum lines were always perpendicular to each other and parallel to the main axes – the road. This was due to their dependence on the footprints representing the architectural program components, which led to a relatively reduced range of substantially different variations in terms of topology. Furthermore, the model did not provide an efficient way of responding to the additional potential constraints such as the residential buildings not parallel to the road. The alternative is to simply expand the usage of
“intelligent” extremums by building desired profiles on small skeletons associated to points driven by the extremums – something that was just partially implemented (the horizontal orange surface guide, Fig.101). Such a set up would allow the new profiles to respond more efficiently to the extreme changes of the extremums’ dimensions. Furthermore, the profiles themselves would be manipulated parametrically for achieving a more versatile design language and consideration of additional constraints (Fig.151). Note that this example also provided the mechanism to explore reasonably ambiguous but meaningful envelope configurations (refer to Chapter 3.2.2).

The “reasoned ambiguity” example also pointed to the importance of systematized emergence or deletion of elements and the ability of parametric systems to cope. This capability may be especially valuable when a designer has some pre-conceived decisions regarding the use of particular structural systems or employing a design language that would require a rigorous use of rules defining, for instance, the building’s envelope.

So, this example showed one of the ways to address the issue – a quite limited technique in terms of versatility of cases it can be applied to. The system did not create new elements or eliminate the unnecessary ones. Its limitation was in the need to create an array of elements that would cover the range of possible variations, from which only the ones satisfying the imposed condition (i.e. roof footprint) would show up. In the end this was an automated Boolean-based system that was cutting elements through constraints.

The last example that used scripting was a more powerful method because it allowed complex rules to govern the design. But the limitation was very similar to the previously
encountered one. I needed to create all the elements beforehand that were either turned on or off by the rules depending on the specific conditions they met. Nevertheless, it was a great improvement over the first approach, which was merely simulating the rules. It introduced a highly systematic way of automating an aspect of the design. This shows great potential of combining two powerful methods – generative and parametric. The ultimate solution would be to employ full fetched programming that platforms like CATIA support. That means elements would be created from scratch when conditions specified by the rules are met.

Now, how can one summarize the benefits or difficulties a designer would extract from relying on parametrics? All the examples discussed in this thesis raised separate questions and pointed to some answers. Most experiments also obeyed a natural increase in complexity that a designer would gain after a more in-depth familiarization with the methodology.

The following charts were developed in an attempt to clearly summarize the strengths and weaknesses of each experiment/technique. A range of 1 to 10 serves as a comparison scale.
4.2 Conclusion and recommendations

This thesis determined that as currently practiced, the architectural design is a highly under-constrained and time consuming process, at least in its initial stages. It is hard if not impossible to change the way architects design by gradually introducing constraints that refine the solution space. It is not the design process that needs revision but rather the use of tools that influence the designer’s decisions. Parametrics is about a rigorous implementation of designer-established rules through the means of constraints and associations between parameters and components. It helps visualize the design spaces. But for this methodology to be truly valuable in generating ideas, the rigor of properly constraining a design task must always be accompanied by the flexibility of the parametric skeleton(s). We are most accustomed to the use of parallel ordering of skeletons, which as seen in the early explorations leads to similar topological solutions. This is not an exciting prospective because as designers we gain little from such a paradigm. For the excitement to emerge a designer must always be kept on guard, be surprised by the meaningful results of the system he created – a very much irregular process. "Irregularity means greater freedom without transgressing the law." And to achieve this, the use of partial ordering of skeletons seems most promising as established in the late experiments. We’ve seen that music is very much an art form based on such an arrangement and its wealth is derived from the elegant balance of the systematic use of rules and the flexibility of these that affords such a great variety of results.

Today parametric or associative design is synonymous with the use of computational platforms. Nevertheless, it is important to reconfirm that machines without the human mind are incapable of being creative and generating meaning, unless man’s intelligence can be mimicked. Parametric methodology is “a construct for thinking first and foremost. It does not have any inherent qualities that ensure a better design, on the contrary it tends to be rather restrictive in most software implementations. Aesthetics are hard to parameterize…”49 This is why the human mind is irreplaceable in the creative process, although relying on parametrics has clearly supported an innovative approach to designing. Nevertheless, a designer must be careful not to fall into the trap of thinking that “this kind of design tool will enable principal designers to quickly produce a variety of ideas, and make these accessible to other team members in a general, re-usable, executable, and extensible form.”50 Parametric design can indeed become a paradigm shift if it will be able to streamline the design process according to this ideal. Meantime, the most feasible improvement is to combine the generative and parametric methods to achieve greater meaning in generating solutions.

Appendix A

This appendix contains the simple Visual Basic script used in the final experiment. Please refer to Chapter 3.2.3 for a description of the code’s operation and the governing rules, as well as an explanation of what went wrong with the original code. Also, refer to the CD-ROM with the thesis CATIA files for clarity of script operation.

```
set Exterior\Rib.7\Activity=true
Exterior\Rib.1\Activity=true
Exterior\Rib.2\Activity=true
Exterior\Rib.3\Activity=true
Exterior\Rib.4\Activity=true
Exterior\Rib.5\Activity=true
Exterior\Rib.6\Activity=true
Exterior\Rib.7\Activity=true
Exterior\Rib.9\Activity=true
Exterior\Rib.10\Activity=true
Exterior\Rib.11\Activity=true
Exterior\Rib.12\Activity=true
Exterior\Rib.13\Activity=true
Exterior\Rib.14\Activity=true
Exterior\Rib.15\Activity=true
Middle_Columns\Rib.17\Activity=true
Middle_Columns\Rib.18\Activity=true
Middle_Columns\Rib.19\Activity=true
Middle_Columns\Rib.20\Activity=true
Middle_Columns\Rib.21\Activity=true
Middle_Columns\Rib.22\Activity=true
Middle_Columns\Rib.23\Activity=true
Middle_Columns\Rib.24\Activity=true
Middle_Columns\Rib.25\Activity=true
Middle_Columns\Rib.26\Activity=true
Middle_Columns\Rib.27\Activity=true
Middle_Columns\Rib.28\Activity=true
Middle_Columns\Rib.29\Activity=true
Middle_Columns\Rib.30\Activity=true
Middle_Columns\Rib.31\Activity=true
Middle_Columns\Rib.32\Activity=true
Middle_Columns\Rib.33\Activity=true
Middle_Columns\Rib.34\Activity=true
```
if length(Surfaces\Line_Y) < 36m and
length(Surfaces\Line_X) < 18m
{
    Exterior\Rib.5\Activity=false
    Exterior\Rib.10\Activity=false
    Middle_Columns\Rib.25\Activity=false
    Middle_Columns\Rib.29\Activity=false
    Middle_Columns\Rib.30\Activity=false
    Middle_Columns\Rib.31\Activity=false
    Middle_Columns\Rib.32\Activity=false
    Middle_Columns\Rib.33\Activity=false
    Middle_Columns\Rib.34\Activity=false
    Middle_Columns\Rib.35\Activity=false
    Middle_Columns\Rib.36\Activity=false
    Middle_Columns\Rib.37\Activity=false
    Middle_Columns\Rib.38\Activity=false
    Middle_Columns\Rib.39\Activity=false
    Middle_Columns\Rib.40\Activity=false
}

else if length(Surfaces\Line_Y) < 36m and
length(Surfaces\Line_X) >= 18m
{
    Exterior\Rib.5\Activity=false
    Exterior\Rib.10\Activity=false
    Middle_Columns\Rib.27\Activity=false
    Middle_Columns\Rib.25\Activity=false
    Middle_Columns\Rib.22\Activity=false
    Middle_Columns\Rib.19\Activity=false
    Middle_Columns\Rib.19\Activity=false
}

else if length(Surfaces\Line_Y) >= 36m and
length(Surfaces\Line_X) < 18m
{
    Middle_Columns\Rib.29\Activity=false
    Middle_Columns\Rib.30\Activity=false
    Middle_Columns\Rib.31\Activity=false
    Middle_Columns\Rib.32\Activity=false
    Middle_Columns\Rib.33\Activity=false
    Middle_Columns\Rib.34\Activity=false
    Middle_Columns\Rib.35\Activity=false
    Middle_Columns\Rib.36\Activity=false
    Middle_Columns\Rib.37\Activity=false
    Middle_Columns\Rib.38\Activity=false
    Middle_Columns\Rib.39\Activity=false
    Middle_Columns\Rib.40\Activity=false
else if length(Surfaces\Line_Y) >= 36m and
length(Surfaces\Line_X) >= 18m
{
    Middle_Columns\Rib.27\Activity=false
    Middle_Columns\Rib.25\Activity=false
    Middle_Columns\Rib.22\Activity=false
    Middle_Columns\Rib.19\Activity=false
}

if length(Surfaces\Line_Y) < 30m and
length(Surfaces\Line_X) < 18m
{
    Exterior\Rib.4\Activity=false
    Exterior\Rib.5\Activity=false
    Exterior\Rib.10\Activity=false
    Exterior\Rib.11\Activity=false
    Middle_Columns\Rib.23\Activity=false
    Middle_Columns\Rib.24\Activity=false
    Middle_Columns\Rib.25\Activity=false
    Middle_Columns\Rib.29\Activity=false
    Middle_Columns\Rib.30\Activity=false
    Middle_Columns\Rib.31\Activity=false
    Middle_Columns\Rib.32\Activity=false
    Middle_Columns\Rib.33\Activity=false
    Middle_Columns\Rib.34\Activity=false
    Middle_Columns\Rib.35\Activity=false
    Middle_Columns\Rib.36\Activity=false
    Middle_Columns\Rib.37\Activity=false
    Middle_Columns\Rib.38\Activity=false
    Middle_Columns\Rib.39\Activity=false
    Middle_Columns\Rib.40\Activity=false
}

else if length(Surfaces\Line_Y) < 30m and
length(Surfaces\Line_X) >= 18m
{
    Exterior\Rib.4\Activity=false
    Exterior\Rib.11\Activity=false
    Middle_Columns\Rib.23\Activity=false
    Middle_Columns\Rib.24\Activity=false
    Middle_Columns\Rib.25\Activity=false
    Middle_Columns\Rib.29\Activity=false
    Middle_Columns\Rib.30\Activity=false
    Middle_Columns\Rib.31\Activity=false
    Middle_Columns\Rib.32\Activity=false
    Middle_Columns\Rib.33\Activity=false
    Middle_Columns\Rib.34\Activity=false
    Middle_Columns\Rib.35\Activity=false
    Middle_Columns\Rib.36\Activity=false
    Middle_Columns\Rib.37\Activity=false
    Middle_Columns\Rib.38\Activity=false
    Middle_Columns\Rib.39\Activity=false
    Middle_Columns\Rib.40\Activity=false
}
if length(Surfaces\LineY) <= 36m and length(Surfaces\LineX) < 12m {
    Middle_Columns\Rib.17\Activity=false
    Middle_Columns\Rib.18\Activity=false
    Middle_Columns\Rib.19\Activity=false
    Middle_Columns\Rib.20\Activity=false
    Middle_Columns\Rib.21\Activity=false
    Middle_Columns\Rib.22\Activity=false
    Middle_Columns\Rib.23\Activity=false
    Middle_Columns\Rib.24\Activity=false
    Middle_Columns\Rib.25\Activity=false
    Middle_Columns\Rib.26\Activity=false
    Middle_Columns\Rib.27\Activity=false
    Middle_Columns\Rib.28\Activity=false
}

if length(Surfaces\LineY) < 24m and length(Surfaces\LineX) <= 12m {
    Exterior\Rib.3\Activity=false
    Exterior\Rib.12\Activity=false
    Middle_Columns\Rib.20\Activity=false
    Middle_Columns\Rib.21\Activity=false
    Middle_Columns\Rib.22\Activity=false
    Middle_Columns\Rib.23\Activity=false
    Middle_Columns\Rib.24\Activity=false
    Middle_Columns\Rib.25\Activity=false
    Middle_Columns\Rib.26\Activity=false
    Middle_Columns\Rib.27\Activity=false
    Middle_Columns\Rib.28\Activity=false
}

if length(Surfaces\LineY) < 18m and length(Surfaces\LineX) <= 12m {
    Exterior\Rib.2\Activity=false
    Exterior\Rib.13\Activity=false
}

else if length(Surfaces\LineY) < 18m and length(Surfaces\LineX) <= 18m {
    Exterior\Rib.2\Activity=false
    Exterior\Rib.3\Activity=false
    Exterior\Rib.12\Activity=false
    Exterior\Rib.13\Activity=false
    Middle_Columns\Rib.20\Activity=false

Middle_Columns\Rib.21\Activity=false
Middle_Columns\Rib.22\Activity=false
Middle_Columns\Rib.32\Activity=false
Middle_Columns\Rib.33\Activity=false
Middle_Columns\Rib.34\Activity=false

else if length(Surfaces\Line_Y) < 24m and
length(Surfaces\Line_X) <= 18m
{
   Exterior\Rib.3\Activity=false
   Exterior\Rib.12\Activity=false
   Middle_Columns\Rib.20\Activity=false
   Middle_Columns\Rib.21\Activity=false
   Middle_Columns\Rib.22\Activity=false
   Middle_Columns\Rib.32\Activity=false
   Middle_Columns\Rib.33\Activity=false
   Middle_Columns\Rib.34\Activity=false
}

if length(Surfaces\Line_Y) < 36m and
length(Surfaces\Line_X) <= 18m
{
   Middle_Columns\Rib.23\Activity=false
   Middle_Columns\Rib.24\Activity=false
   Middle_Columns\Rib.35\Activity=false
   Middle_Columns\Rib.36\Activity=false
   Middle_Columns\Rib.37\Activity=false
}

if length(Surfaces\Line_Y) < 12m and
length(Surfaces\Line_X) <= 18m
{
   Exterior\Rib.1\Activity=false
   Exterior\Rib.9\Activity=false
}

else if length(Surfaces\Line_Y) < 12m and
length(Surfaces\Line_X) < 12m
{
   Middle_Columns\Rib.17\Activity=false
   Middle_Columns\Rib.18\Activity=false
   Middle_Columns\Rib.19\Activity=false
   Middle_Columns\Rib.29\Activity=false
   Middle_Columns\Rib.30\Activity=false
   Middle_Columns\Rib.31\Activity=false
}

92
else if length(Surfaces\Line_Y) < 12m and length(Surfaces\Line_X) >= 18m
    { Exterior\Rib.1\Activity=false
      Exterior\Rib.2\Activity=false
      Exterior\Rib.3\Activity=false
      Exterior\Rib.9\Activity=false
      Exterior\Rib.12\Activity=false
      Exterior\Rib.13\Activity=false
      Middle_Columns\Rib.20\Activity=false
      Middle_Columns\Rib.21\Activity=false
      Middle_Columns\Rib.22\Activity=false
      Middle_Columns\Rib.32\Activity=false
      Middle_Columns\Rib.33\Activity=false
      Middle_Columns\Rib.34\Activity=false
    }

else if length(Surfaces\Line_Y) < 18m and length(Surfaces\Line_X) >= 18m
    { Exterior\Rib.3\Activity=false
      Exterior\Rib.12\Activity=false
      Middle_Columns\Rib.20\Activity=false
      Middle_Columns\Rib.21\Activity=false
      Middle_Columns\Rib.22\Activity=false
      Middle_Columns\Rib.32\Activity=false
      Middle_Columns\Rib.33\Activity=false
      Middle_Columns\Rib.34\Activity=false
    }
Appendix B

This appendix identifies and explains geometrical and dimensional types of constraints. The difference is in the fact that geometrical constraints are relationships that establish limitations between objects, where as dimensional are defining the objects. Following is a description of the most common types of constrains. CATIA was used to illustrate the concept through simple examples. Note that the types of constraints are dependent on the number of constrained objects.

*Dimensional constraints*

1. **Distance** – value defining the distance between two elements
2. **Length** – value defining the elements
3. **Angle** between two elements
4. **Radius / Diameter**
Geometric constraints

1. **Fixed** – anchor sign – fixes the end of the element (one objects constraint)
2. **Horizontal** – forces the element be always in a horizontal state (one objects constraint)
3. **Vertical** – forces the element be always vertical (one objects constraint)
4. **Coincidence** – forces the selected side of one element occupy the same relative position in space as the selected side of the other element
5. **Concentricity** – common center constraint
6. **Tangency** – forces common point of contact between two curved entities
7. **Parallelism** – forces two lines to be always parallel to each other
8. **Perpendicularity** – forces two lines to be always perpendicular to each other
Bibliography


*Notes*: unless specified, all images are the property of the author;
see attached CD-ROM with CATIA files of the described experiments.